AN INVESTIGATION AND IMPLEMENTATION OF NONPARAMETRIC STATISTICAL TESTS ON A DIGITAL COMPUTER

by 632

KEITH ERIC MOE

B. S. E., Arizona State University, 1969

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1970

Approved by:

K. E. Kemp
Major Professor

LD
2665
R4
1970
M64
C.2

ACKNOWLEDGEMENT

The author would like to express his appreciation to Dr. W. J. Conover, Associate Professor of Statistics and Computer Science, Kansas State University. It was Dr. Conover's course and text on nonparametric statistics that gave the author the idea for this report. Although unable to be a member of the author's committee, Dr. Conover was able to give personal assistance and guidance during the planning stage of this paper. The author would also like to acknowledge the assistance of Mr. Eldon Wancura, Associate Professor, Kansas State University Library. Through Mr. Wancura the author was able to Xerox the appendices of this report using the reducing Xerox machine of the Library.

TABLE OF CONTENTS

PAGE

LIST OF TABLES iv

LIST OF FIGURES v

CHAPTER

1. INTRODUCTION 1

2. DESIGN 5

3. IMPLEMENTATION 13

INITIAL ORGANIZATION 14

CODING OF THE PRIMARY SUBROUTINES 15

SUPPORT SUBROUTINES 22

TESTING SUBROUTINES 23

DISTRIBUTION FUNCTIONS 33

4. SUMMARY 36

BIBLIOGRAPHY 38

APPENDICES 40

I. FORTRAN PROGRAM LISTINGS

II. ASSEMBLY LANGUAGE PROGRAM LISTINGS

III. SAMPLE OUTPUT

IV. PRELIMINARY USER'S GUIDE

LIST OF TABLES

TABLE		PAGE
I.	NUMERICAL AND PROGRAM CODES FOR TESTS PERFORMED ..	7
II.	NUMERICAL AND PROGRAM CODES FOR DATA TYPES	8
III.	DATA TYPES VERSUS TESTS PERFORMED	9
IV.	PARAMETER CARDS	11

LIST OF FIGURES

FIGURE		PAGE
1.	Samples of parameter cards.....	12
2.	Subroutine linkage structure.....	16

Chapter 1

INTRODUCTION

In recent years, statistical analysis has come into widespread use, primarily because the digital computer has taken the burden of laborious computation out of man's hands. The areas in which statistical analysis has been applied have been broadened because the digital computer has reduced the time and labor needed for adequate statistical analysis of the great quantity of data being produced today. The computer has also taken a good deal of the boredom out of statistics, boredom with the tedious and error prone arithmetical computations involved.

Statisticians have had the nature of their work changed by the computer. With a computer available, the statistician can develop complex experimental designs without sacrificing anything merely to insure computational simplicity. These more complex designs yield more information for the experimenter with greater accuracy than has ever been possible before. But in order to perform complex statistical analysis, programs must be available that will process the data in the desired manner.

Many programs have been written to perform statistical computations in many different ways. Initially, programs were written to perform only a single function (often only for a particular experiment). This led to much duplication of effort on the part of many statisticians and programmers. The result was many programs to do the same thing, sometimes with identical

results, other times with startling differences. To attempt to eliminate much of this unnecessary duplication brought about by single usage programming, statistical "packages" were developed to handle many types of data and methods of analysis.

These packages cover a variety of statistical problems. Some may be general enough to handle a certain type of data from a certain statistical design; while others may handle almost any kind of data from almost any kind of design. All of these packages have their limitations, but if one package is unsuitable, the experimenter can usually find another one that is more suitable. The basic design of these packages is that if the experimental data are statistically analyzable, the program can handle the data obtained from the experiment only subject to whatever restrictions were initially placed on the data. This is a rather tall order, and how well these packages have met this goal varies from good to bad.

There are in existence now, packages to handle data of such forms as one or more random samples, bivariate or multivariate samples, and randomized complete or incomplete blocks. Methods of analysis are many and include Analysis of Variance (AOV), t-Tests, LSD and NMRT comparisons, and correlation and regression analysis. Several packages have these statistical tests combined in such a way that the experimenter can find out more about his data than he could ever possibly want to know. The methods used in these packages usually revolve around some form of an Analysis of Variance. Some other techniques employed usually contribute to the AOV by indicating how well a regression

model fits the data. If an experimenter wants to analyze his data using an Analysis of Variance technique, he has many programs available to him.

Analysis of Variance methods have several short-comings which seem to be overlooked, ignored, or assumed to be insignificant. The short-comings arise from the assumptions which must be made in order for the F-Test to be valid. The most restrictive of these assumptions is that the data are normally distributed (or that the distribution function is known and hence the data can be analyzed). Analysis of Variance is not the only statistical technique that requires this assumption. Many of the most commonly used statistical methods include this assumption as an integral part of the theory from which they were developed. These techniques are called parametric statistics, since certain "parameters", or characteristics, are assumed to be of a known (or estimable) form.

But what is the experimenter to do if he feels or knows that his data cannot be assumed to be from a normally distributed population. He has three courses of action to follow. One is to ignore the problem and hope his results are valid. Another is to attempt to transform his data into a suitable distribution, but this requires some knowledge about the distribution of the data. The experimenter's third alternative is to change the method of analysis to one in which the assumption of a known distribution is not necessary. For this third alternative the experimenter may use nonparametric statistical techniques.

Nonparametric statistics can be considered to be that area of statistical analysis in which no assumptions need to be made about the data being analyzed, other than the assumption that the data are representative of the population(s) about which inferences are to be made. (If they aren't, the analysis is biased and the conclusions reached do not accept or reject the null hypothesis.) Thus, application of nonparametric techniques can be made with no fear on the part of the experimenter about how "normal" his data are. However, nonparametric statistics have had several drawbacks. The worst of these drawbacks is the lack of widespread acceptance. People are somewhat leery of them since they aren't as well known or as widely accepted as parametric methods. Another drawback has been that there was no comprehensive nonparametric statistical analysis package for computers available.

Individual programs and subroutines for individual nonparametric methods have been written, but they suffer from the same problems that the individual parametric programs do. They are usually specialized and cover only a narrow area of nonparametric analysis. This report is a study of the first attempt (to the author's knowledge) to implement a comprehensive set of nonparametric statistical tests in a single digital computer package. This package, coupled with several other packages (of the parametric type), can give the statistician almost all the computational ability he is likely to need to handle his customers' (and possibly his own) work.

Chapter 2

DESIGN

It had been decided at the outset that the programming language to use for this statistical package was FORTRAN. The primary reasons for this decision was because FORTRAN is mathematically oriented and because most of the existing statistical packages had been written in FORTRAN, thereby providing some common base from which to work. While the use of FORTRAN had certain disadvantages, the advantages that it did offer were considered to far outweigh these disadvantages. Some of the actual problems encountered by using FORTRAN and their circumventions are discussed in Chapter Three of this report.

The intended design of the nonparametric statistical package was to encompass the most common (in the author's judgement) nonparametric tests and their corresponding data types without restricting either one. Separate sections of the programs were planned as separate subroutines communicating with each other via a common interface.

Before the overall structure of the program was developed several preliminary items had to be decided. One was which statistical tests were to be included. Fifteen tests

were selected from a nonparametric statistical textbook[1] and from discussion with the author of that text. These tests were selected as being those that were most commonly used. A list of these fifteen tests appears in TABLE I.

After the selection of the tests, the types of data which were required by the tests had to be selected. Each of the data types has some distinguishing characteristic that makes it different from the other types, and hence, each was considered unique, even though some types were special cases of more general types. This allowed each individual test to be categorized by the type of data that was needed for that test. TABLE II lists the data types used and TABLE III lists the data types versus the tests performed.

Once the types of data had been decided upon, the methods of input needed to be defined. It was decided that the methods and formats of input should be as flexible as possible. This was to allow the user to code the input data in the manner best suited to his needs and data, rather than forcing him to conform to a fixed type of data input. But all this flexibility would be lost if the user could not alter, transform, or edit the data during input processing or before testing. Provisions had to be included to allow the user to transform his data to suit his needs, such as replacing observations by their square

---

[1] W. J. Conover, Practical Nonparametric Statistics, (Rough Draft; Manhattan, Kansas: In Press, 1970).

TABLE I

NUMERICAL AND PROGRAM
CODES FOR TESTS PERFORMED

TEST NUMBER	TEST NAME	PROGRAM CODE
1	SIGN TEST	SIGN
2	Mc NEMAR TEST	MCNEMAR
3	CHI-SQUARE TEST - PROB. DIFF.	CHIPROB
4	CHI-SQUARE TEST - INDEPENDENCE	CHINDEP
5	MEDIAN TEST	MEDIAN
6	WILCOXON SIGNED RANKS TEST	SIGNRANK
7	MANN-WHITNEY TEST	M-W
8	MOOD'S TEST	MOOD
9	SPEARMAN'S RHO	RHO
10	TREND TEST USING SPEARMAN'S RHO	TREND
11	KENDALL'S TAU	TAU
12	KRUSKAL-WALLIS TEST	K-W
13	FRIEDMAN TEST	FRIEDMAN
14	CHI-SQUARE GOODNESS OF FIT TEST	GOODFIT
15	DURBIN TEST	DURBIN

TABLE II

NUMERICAL AND PROGRAM
CODES FOR DATA TYPES

TEST NUMBER	DATA TYPE NAME	PROGRAM CODE
1	ONE RANDOM SAMPLE	1RS
2	TWO RANDOM SAMPLES	2RS
3	SEVERAL OR "K" RANDOM SAMPLES	KRS
4	BIVARIATE SAMPLE	BIVAR
5	MULTIVARIATE SAMPLE	MULVAR
6	TWO MATCHED SAMPLES	2MS
7	RANDOMIZED COMPLETE BLOCK	RCB
8	BALANCED INCOMPLETE BLOCK	BIBD

TABLE III

DATA TYPES VERSUS TESTS PERFORMED

TESTS PERFORMED	DATA TYPES 1RS	2RS	KRS	BIVAR	MULVAR	2MS	RCB	BIBD
SIGN				X		X		
MCNEMAR				X		X		
CHIPROB		1	1					
CHINDEP	2			2				
MEDIAN		X	X				X	
SIGNRANK	3			X		X		
M-W		X						
MOOD		X						
RHO				X				
TREND	X							
TAU				X				
K-W		X	X					
FRIEDMAN					X		X	
GOODFIT	1,4							
DURBIN								X

Notes:

1. Data dichotomization by COLUMN must be specified for a contingency table.
2. Data dichotomization by ROW and COLUMN must be specified for a contingency table.
3. A MEDIAN value to be tested against must be specified.
4. A hypothesized distribution function with appropriate parameters must be specified.

See TABLES I & II for the TESTS PERFORMED and DATA TYPES coding.

roots or discarding observations falling outside a certain range, etc.

Because the user might want to test his data under several different conditions, the program had to be able to save the data for future processing. This meant that the program had to save the data in some form that the user could use later, either by this or another program.

In nonparametric tests there is often the need to specify data dichotomization for contingency tables or to specify a special value for a test such as a median value. These values must be specified by the user. The need existed for these values, as well as the other information discussed above, to be specified in a simple, easy to learn form. The solution to this was to have a set of "parameter cards" supplied by the user as part of the input of each set of test data to be tested.

Each parameter card consisted of a keyword, which identified the contents of the card as meaning a particular thing, followed by a comma, followed by one or more parameters separated by commas. A list of parameter cards and their general contents appear in TABLE IV. Examples of parameter cards appear in Fig. 1 and APPENDIX III. A more thorough explanation of parameter card specifications appears in APPENDIX IV.

TABLE IV

PARAMETER CARDS

PARAMETER KEYWORD	GENERAL USE	MEANING OF PARAMETERS
TITLE	Experiment title	Contents of CC 7-80 used for title on output
DATA	Describe input data	Type of data, no. of samples, sample sizes or Type of data, no. of blocks, treatments, reps or Type of data, no. of blocks, treatments, blocks per treatment, treatments per block
LABEL	Label input data	Labels of 8 or fewer characters
BLOCK TREAT REP	Label blocks, treatments, replications	Labels of 8 or fewer characters
CODING	Describe input data	Data items per read, how data are broken down
SOURCE	Data input and output units	Input unit, input form, output unit, when to write output
DISTRIB	Test distribution or Median value or Level value	Distribution name, parms Median indicator, value Level indicator, value
CLASSES	Contingency table size and range	ROW or COLUMN, no. of rows or columns, upper limit of each row, column
TESTS	Tests to be performed	Alpha level, tail, tests
TRANS	User transformation	When to transform data
END OF PARAMETERS	Always the LAST parameter card, signals the end of the parameters for this test set	

```
Card Columns
0    0    1    1    2    2    3    3    4    4    5    5
1    6    1    6    1    6    1    6    1    6    1    6

TITLE,   THIS IS A TITLE CARD

TITLE=THIS IS ANOTHER TITLE CARD USING THE EQUAL SIGN

DATA,1RS,,27
     1 random sample of 27 observations
DATA,RCB,4,3,8
     randomized complete block with 4 blocks, 3 treatments,
     and 8 replications
LABEL=SAMPLE1,SAMP2,SMPL3,FOUR
     labels for samples 1, 2, 3, & 4
LABEL(5)=SMPLX,SMPLY
     labels for samples 5 & 6
BLOCK=BLKLABEL,TREAT=TRTLBL,REP=REPS
     labels for blocks, treatments, replications
CODING,4,2,2
     4 items on each read, 2 for sample 1, 2 for sample 2
CODING,1,25,30
     1 item on each read, 25 items go to sample 1, then 30 items
     go to sample 2, then 25 items to sample 1, etc.
SOURCE,1,EXTERNAL,3,RAW
     external (formatted) data read on unit 1, written out on
     unit 3 as it comes in
SOURCE,7,INTERNAL,17,TRANSFORMED
     internal (unformatted) data read on unit 7, written on
     unit 17 after user transformation
DISTRIB,NORMAL,7.,10.
     normal distribution with mean 7 and variance 10
DISTRIB,MEDIAN,-4.5
     median of data will be tested against -4.5
CLASSES,ROWS,4,4.,7.,10.5
     4 rows in a contingency table with upper limits
     of 4, 7, 10½, and +infinity
TESTS,.05,UPPER,SIGN
     upper tail Sign Test performed with alpha=.05
TESTS,.10,ALL
     all appropriate tests for supplied data performed as
     2-tailed tests with alpha=.10
TRANS,NONE
     no user transformation will be performed
TRANS,RAW
     transformation will take place as the data is read in
END OF PARAMETERS
     no more parameter cards, FORMAT & data may follow
     this parameter cards
```

Fig. 1. Samples of parameter cards.

Chapter 3

IMPLEMENTATION

Implementation of this package was not without its problems. The selection of FORTRAN as the primary programming language left something to be desired in several areas, most notably in the area of parameter card decoding. It was felt that no language available could handle all areas that needed to be covered with sufficient ease to justify the use of that single language exclusively. The final decision was that a mixed language approach be used, with FORTRAN being the primary language and IBM System/360 Assembly Language being secondary. It was realized by the author that such an approach would make the package machine dependent.

The machine that the package implemented on was an IBM System/360 running under the Operating System (OS). The use of the IBM FORTRAN (H) Compiler[2] and the System/360 Assembly Language using the OS Macro facilities[3] was not considered, by the author, as a severe restriction. The program was written to be system independent, i.e., it could be taken to any

---

[2]IBM Corporation, IBM System/360 Operating System: FORTRAN IV (G and H) Programmer's Guide (IBM System Reference Library, Form C28-6817-0, White Plains: IBM Corporation, 1967).

[3]IBM Corporation, IBM System/360 Operating System: Supervisor and Data Management Macro Instructions (IBM System Reference Library, Form C28-6647-3, White Plains: IBM Corporation, 1968).

System/360 installation running under OS and having sufficient memory for the program and run without modification. Certain features that any particular installation might want to change, such as the FORTRAN Unit Numbers used, were placed in several small subroutines so that they could be easily changed. To run on the System/360 Disk Operating System (DOS), several minor changes would have to be made to several other subroutines because of the different conventions of OS and DOS. These changes were considered minor by the author. Installation of the package on a non-System/360 computer was not considered impossible, but would be rather difficult. However, the package shares this disadvantage with several other statistical packages with which the author is familiar.

INITIAL ORGANIZATION

The first step in the implementation was to define all the fields that had to hold the information from the parameter cards and input data. These fields were organized into several different common blocks according to their general purposes. This allowed information to be passed throughout the subroutines used with ease. An interesting conclusion was reached when these common blocks were being organized. It was decided that almost all of the arithmetic done in the program could be done in single precision. The reason behind this lies in the statistical tests being programmed. Most of the test statistics computed consist of counts or sums of ranks. The six digit

accuracy of single precision arithmetic was considered sufficient for this purpose. The ultimate result of this was a substantial saving in memory space and computation time.

The common block organization that was first formulated was not the final one implemented. Although every common block initially planned stayed intact, several additional common blocks had to be designed to handle the items which had been overlooked during the planning stages. No great difficulties were caused by having to add these additional blocks since they were added before programming had progressed very far.

After the data fields were organized, the overall structure of the program was developed. This was done by organizing the functions that the program had to perform into a flowchart and then assigning subroutines to the major flowblocks. The final subroutine organization is shown in Fig. 2. The initial organization was essentially the same as that shown except for several functions which were divided into subroutines. Also, several functions were consolidated into single subroutines rather than the multiple subroutines planned. After the organization chart was made up the actual coding began.

CODING OF THE PRIMARY SUBROUTINES

Coding was not started with the main program, as one might expect, but rather with the parameter analysis subroutines. The primary subroutine was RDPARM, a FORTRAN subroutine to read parameter cards and analyze them only within the context of

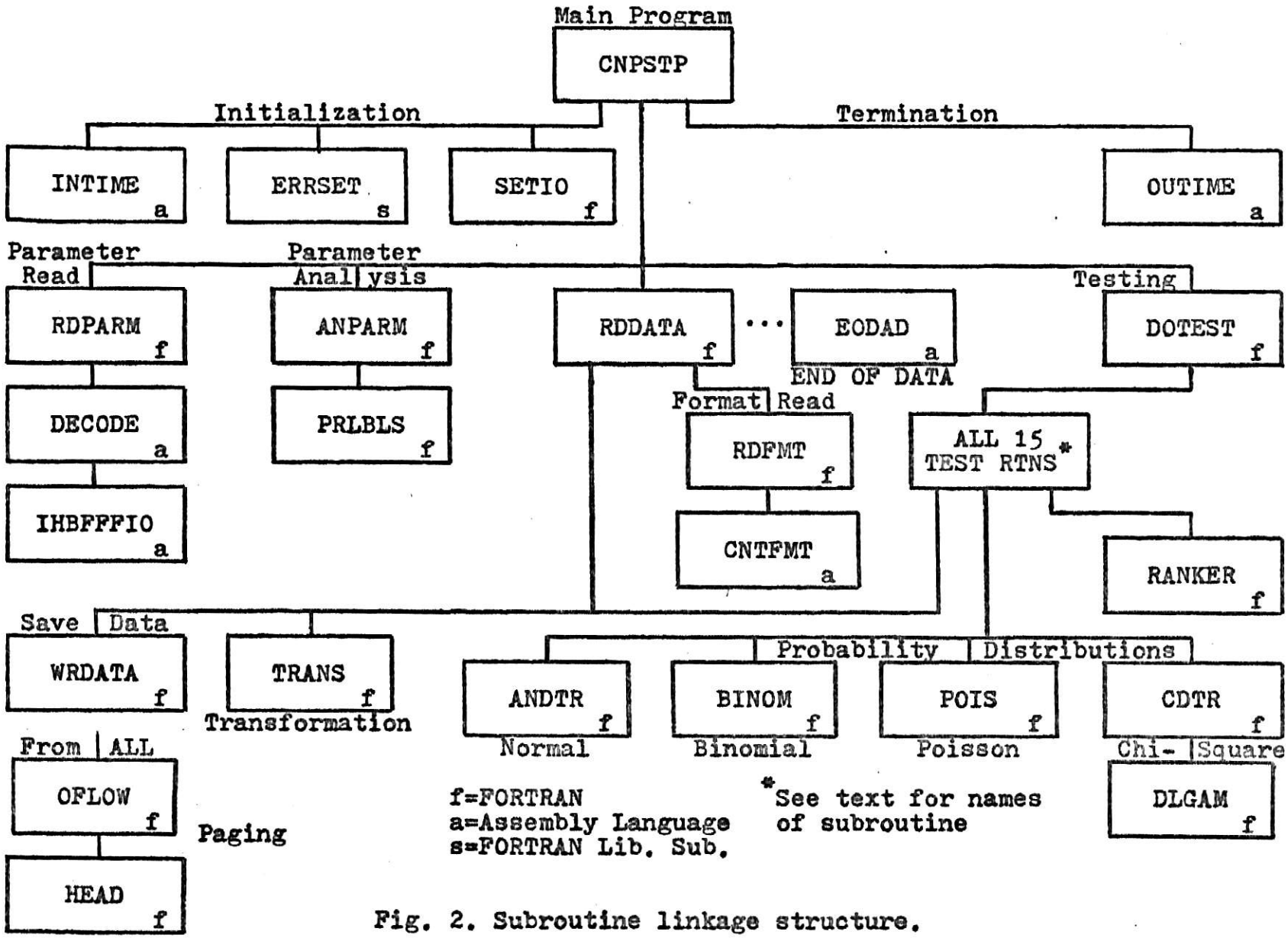

Fig. 2. Subroutine linkage structure.

themselves, i.e., conflict among parameters on separate cards was not checked. Because of the form selected for the parameter cards, the parameter cards were converted from their rather free form to a fixed form which was easily handled by FORTRAN. This was done by an Assembly Language subroutine, DECODE. This subroutine separated the parameters on a parameter card and placed each one in a separate field. DECODE also evaluated numerical parameters by converting every parameter from their external character form (EBCDIC) to internal real and integer numbers. To perform these conversions, DECODE took advantage of a pre-existing floating point data conversion routine written by the author. This routine, IHBFFFIO, is not discussed further in this report, since it had not been written specifically for this program and had no other use beyond that mentioned above.

The second phase of parameter analysis was carried out by subroutine ANPARM. This routine checked for conflicting or missing parameters, such as specifying a test that requires a contingency table, but not supplying one. When any errors were detected, a return code was placed into a special field and control was passed back to the main program. The presence of a non-zero return code indicated some error in processing. This method of signalling errors was used throughout the program and the occurance of a non-zero return code at any point in the processing resulted in termination of the processing for the current set of parameters and data. (Further discussion of this error handling is deferred until the writing of the main program is presented.)

After successful reading and analysis of the parameters, subroutine RDDATA was called to perform the actual reading of the user's data. If the data form was specified as being external (formatted data), RDDATA called RDFMT, which read the data FORMAT statement. Since the FORMAT was allowed to be longer than the room available on a single card, RDFMT read cards until all of the left and right parentheses were balanced. The actual counting of parentheses was done by an Assembly Language subroutine, CNTFMT, since a character by character scan of the FORMAT statement would have been difficult to program in FORTRAN. RDDATA then proceeded to read the user's data from the unit the user had specified.

Since the package was designed to handle multiple sets of data, a special end of data signal had to be devised. This consisted of a card with the words "END OF DATA" in columns 1 to 11 of the first card following the user's data. When this card was read under FORMAT control, a data conversion error occurred. In order to catch this error a special subroutine, EODAD, was written in Assembly Language to catch this condition. System/360 FORTRAN contained an Extended Error Message Facility[4] to allow program intervention upon occurrance of certain errors. This facility was used to examine the contents of any card that caused a data conversion error. By extracting the contents of

---

[4] FORTRAN IV (G and H) Programmer's Guide, pp. 103-116.

the card that caused a conversion error from the input buffer,[5] EODAD was able to determine the actual column in the card which caused the error. A flag was set in a special common block indicating that either a real data conversion had occurred, or that an "END OF DATA" card had been read. RDDATA examined this flag after every data read and acted accordingly, indicating the error to the user if his data card was in error and terminating the read if the "END OF DATA" card had been read. During the testing phase of the program, it was decided to modify EODAD to signal end of data if the TITLE parameter card of the next set of parameter cards was encountered while reading the input data. This was primarily to prevent the unnecessary bypassing of legitimate testing if the "END OF DATA" card was accidentally omitted following the data. If this were the case, the user was forgiven with a warning message and processing continued.

The user could have specified on the SOURCE parameter card that the data was not on the same input unit as the parameter cards. He could also have specified whether the data was formatted (external character form, EBCDIC) or unformatted (internal binary representation). This allowed the user to read data created in other programs or stored on the specified unit previously by this program. Also specified on the SOURCE card was the optional saving (writing out) of the data. This

---

[5]IBM Corporation, IBM System/360 Operating System: FORTRAN IV (H) Compiler. (IBM Program Logic, Form Y28-6642-3, White Plains: IBM Corporation, 1968), pp. 189-191.

let the user pass the input data to other tests later in the input of this program or to another program. The output of the data could occur at any one of four (or less, depending on the data type) different points during the processing of the data. The point of output was specified along with the output unit number. The data were written on this unit by subroutine WRDATA in internal format unless the unit number specified was the same unit that was used for the printed output. If the unit numbers were the same, the data were printed, producing a listing of the data.

Data transformation could occur on the input data if the user so specified on the TRANS parameter card. To perform the transformation, a call was made to the subroutine TRANS, which must have been written by the user, compiled, and link edited into the program. TRANS could add, change, or delete data to suit the user.

During input processing, each item[6] of data was treated separately. Each was assigned to a particular sample, variate, or block-treatment-replication combination and written out on intermediate storage for use in the actual processing. The actual testing was performed on these data items. Although in many cases it would have been possible to keep all of the data in core

---

[6] In this report, the term "item" means a single data value, regardless of the type of data it is from or its connection with any other item. The term "observation" is used to denote one or more "items" that are combined in some meaningful way, such as the two items in a bivariate observation, or all replications in a particular block-treatment combination.

and process it without the overhead involved in writing the data out and rereading it (often several times), it was felt that the program should handle the most general case of an undetermined amount of data without special routines to handle the case of small samples. It was also felt that the overhead of unnecessary I/O operations was negligible in most cases.

After correct processing of the input data each of the specified tests was performed. This was controlled by subroutine DOTEST which called a processing subroutine for each of the tests to be performed. At the completion of all testing (or termination of testing due to a processing error) control was passed (via the main program) to RDPARM, to read the next set of parameters.

The main program, given the acronym of CNPSTP for Complete NonParametric Statistical Test Package, was quite simple in design, but was not so simple in implementation. The main program was to pass control to RDPARM, ANPARM, RDDATA, and DOTEST, one after the other. After each subroutine call, however, the return code had to be interrogated for possible errors or for end of job. CNPSTP also had to perform initialization of key fields when first entered. This included a call to ERRSET, a FORTRAN supplied subroutine,[7] to enable data conversion error processing, a call to SETIO to establish the I/O unit numbers for processing, and a call to an Assembly Language subroutine,

---

[7] FORTRAN IV (G and H) Programmer's Guide, pp. 104-107.

INTIME, to allow access to the system timer facilities,[8] so that the total processing time could be recorded.

SUPPORT SUBROUTINES

Several general purpose subroutines were written to support particular functions that were needed by some or all portions of the program. Subroutine RANKER was used by some of the testing subroutines to sort the data in ascending sequence, so that the median could be determined or the data could be assigned ranks. RANKER read the data from one intermediate work unit and wrote sorted data onto another work unit. If the data contained more items than could be sorted in the work area in core, one or two intermediate work units were employed to hold sorted sections of the data for merging into the output.

Subroutines OFLOW and HEAD were used to control the printed output page headings. HEAD incremented the page number, printed a new page heading containing the page number, the date, and the user's title from the TITLE parameter card, and reset the line counter. It could be called any time that any subroutine in the program wanted to start a new page of printed output. OFLOW was usually called after a write command to the printed output unit to increment the current line count by the number of lines printed (specified as a parameter in the CALL statement) and to test this line count against a specified

---

[8] Supervisor and Data Management Macro Instructions, pp. 181-182, 201.

maximum number of lines per page (originally specified as 55 lines per page). If the number of lines per page had been reached or exceeded, OFLOW called HEAD to print a new page heading and reset the line counter.

Several subroutines were used to calculate probabilities from particular distribution functions. Some of these were written by the author, while others were extracted (with modifications) from the IBM Scientific Subroutine Package (SSP).[9] These routines provided the probability of a random variable being less than (in one case, greater than) or equal to a specified value, when the random variable was from a population with the particular distribution function called and the parameters specified. These routines are detailed individually after all of the individual tests.

TESTING SUBROUTINES

There were fifteen different statistical test subroutines. Some of these routines performed their tests on more than one type of data, but each was designed to perform a particular test, regardless of the data type. There was a great deal of duplication of code from one subroutine to another and when multiple tests were performed on one set of data several things (such as sorting of the data) might have been done more

---

[9] IBM Corporation, IBM System/360: Scientific Subroutine Package Version III, Programmer's Manual. (IBM Application Program, Form H20-0205-3, White Plains: IBM Corporation, 1969).

than once. No attempt was made to eliminate the duplication and redundancy that existed in this design. By making each test an independent subroutine, the modularity of the total package was preserved. Each of the test subroutines is discussed below. More detailed information on each test can be found in the appendices and the text from which they were taken.[10]

The Chi-Square Test for Independence (subroutine CHINDP) computes the probability of data classified under one criterion being independent of its classification under a second criterion. The subroutine built an "R x C" contingency table and classified each observation into both a row and a column. If a single random sample was supplied, each observation was used for both classifications. If a single random sample was supplied, but the user transformation subroutine supplied a second item for each observation, the first was used for the row classification and the second for the column classification. If the user had provided a bivariate sample, the subroutine classified each observation by using the first variate values as row values and the second variate values as column values. Items were categorized into rows (or columns) by placing an upper value limit on each row (column) and determining which upper limit an observed value was less than or equal to. These upper limits were provided by the user on the CLASSES parameter card. The function subprogram

---

[10]Conover, Practical Nonparametric Statistics.

CDTR[11] was used to determine the probability of the test statistic, calculated from the contingency table, being that large or larger under the null hypothesis of independence classification.

The Chi-Square Test for Probability Differences (subroutine CHIRXC) classified "R" random samples (which formed rows) into "C" classes (which formed columns) to make an "R x C" contingency table. The column classification was performed in the same manner as the classification in subroutine CHINDP. This contingency table was used to compute a test statistic and function CDTR was then used to find the probability of a value of the particular magnitude of this test statistic under the null hypothesis. The null hypothesis was that the probability of observations being placed in a particular column is the same for all samples.

The Friedman Test (subroutine FRDMAN) tested for treatment effects in randomized complete block experiments. The user provided data for all of the possible block-treatment combinations (multiple replications permitted). The observations were ranked within blocks, across treatments and replications. The ranks were summed within treatments, across blocks and replications. The resulting sums were used to calculate a test statistic with the chi-square distribution under the null hypothesis of no differences in treatments. Function CDTR was used to find the probability of a true null hypothesis.

---

[11] IBM SSP, pp. 81-83.

The Durbin Test (subroutine DURBIN) performed what could be considered a Friedman Test restricted to one replication, but extended to incomplete block data. The user specified the number of blocks, treatments, treatments observed in each block, and blocks in which treatments were observed. The observations were ranked within blocks, across treatments, and then summed within treatments, across blocks. The rank sums for the treatments were used to compute a test statistic having the chi-square distribution under the null hypothesis. The null hypothesis was the same as that of the Friedman Test, that all treatments have equal effects. As in the Friedman Test, function CDTR was used to determine the probability of a true null hypothesis.

The Chi-Square Goodness of Fit Test (subroutine GDFIT) was used to test the hypothesis that the data being tested came from a population with a certain distribution. The user provided a single random sample, contingency table limits for columns (see the discussion in the paragraph on the Chi-Square Test for Independence), and the name and parameters of the hypothesized distribution function. The routine used the sample to provide the first row of a "2 x C" contingency table and provided the second row by calculating the theoretical column breakdown from the sample size, the specified distribution, and the supplied parameters. The user could specify any one of four distributions: Normal, with user supplied mean and variance; Poisson, with user supplied mean; Uniform, with equal probability

assigned to each column; and Binomial, with specified parameters N and P. The functions ANDTR, POIS, and BINOM were used for the normal, Poisson, and binomial distribution calculations, respectively. No subroutine was needed for the uniform distribution. As with the other contingency table tests, function CDTR was used to determine the probability of a true null hypothesis.

The Mann-Whitney Test (subroutine MANWHT) was a test on two random samples to determine differences in their expected values (medians). The two samples were ranked in their entirety and the sum of the ranks of the first sample was computed. A test statistic was calculated and a large sample normal approximation was used to find the probability of a true null hypothesis, using the function ANDTR. Since a large sample approximation was used, a warning message was printed if either sample had fewer than 20 observations, but the probability was still calculated. The null hypothesis being tested could specify if expected value of the first sample should have been equal to, less than, or greater than the expected value of the second sample, versus a suitable alternative.

The Kruskal-Wallis Test (subroutine KRSKAL) is essentially a "K" sample Mann-Whitney Test. The subroutine tested a null hypothesis that the expected values (medians) of all "K" samples could be considered equal. The observations from all "K" samples were ranked without regard to the sample from which any observation came. The sums of the ranks for each sample were formed and a test statistic was calculated. The probability

of a true null hypothesis was determined from the chi-square distribution, using function CDTR. The chi-square distribution was actually a large sample approximation and could be rather poor if there were fewer than three samples or fewer than 15 observations (all samples combined) being tested. For the case of only two samples the Kruskal-Wallis Test gave the same probability as the Mann-Whitney Test of equal versus unequal expected values.

The Mc Nemar Test for Significance of Changes (subroutine MCNMAR) is used to test a bivariate sample or two matched samples (two random samples of the same size in which observations are paired - one from each sample - to form a bivariate sample) for changes between the two variates. The first variate represented a "before" situation and the second an "after" situation. In performing the test, both variates of each observation were classified as a zero or a one. The user specified a dichotomization level and every item less than this value was made a zero while every item greater than this value was made a one. Any equal items caused the whole observation (both variates) to be eliminated from the testing. If the user didn't provide a dichotomization value, the subroutine used one-half, thereby assuming the items were already zeros and ones. The probability of the null hypothesis of no significant change from before to after was determined from the binomial distribution (using function BINOM) if 20 or fewer observations changed and from the chi-square distribution (using function CDTR) otherwise.

The Median Test (subroutine MEDIAN) is a test of "K" random samples with a null hypothesis that all "K" samples came from populations with the same median, versus an alternative that at least one sample comes from a population with a median different from the medians of the other populations. The subroutine found the grand median and counted the number of observations in each sample greater than the grand median and the number less than or equal to the grand median. A test statistic, with the chi-square distribution under the null hypothesis, was calculated to find the probability of a true null hypothesis. Function CDTR was used for this purpose. An Extended Median Test was also included in this subroutine to perform a Median Test on randomized complete block experimental data. The tests performed on RCB data were: 1) A test for equal medians among blocks; 2) A test for equal medians among treatments; and 3) A test for equal medians among replications. The test statistic for each of these tests had a chi-square distribution under the null hypothesis and function CDTR was used to determine the probability of a true null hypothesis.

Mood's Test of Variances (subroutine MOOD) is used to contrast the variances of two random samples. The null hypothesis tested is that the variance (dispersion) of the first sample is equal to, greater than, or less than that of the second sample, versus a suitable alternative. When Mood's Test was performed by the subroutine, the two samples were ranked in totality and a test statistic was calculated from the ranks

assigned to the first sample. The probability of a true null hypothesis was determined from a large sample normal approximation. Function ANDTR was used for this purpose. If either sample was smaller than 20 observations a warning was printed, stating that this probability might be incorrect.

The parametric counterpart to Mood's Test is the F-Test. The advantages (and disadvantages) of Mood's Test (and others) is stated by Conover. "... the F test is known to be quite sensitive to departures from normality, ... This suggests using one of the above [Mood's and others] nonparametric tests when the populations may be nonnormal. In turn, the above nonparametric tests rapidly lose power in the presence of unlike location parameters, a factor by which the F test is unaffected."[12] The user must take this into account when using Mood's Test.

About the simplest nonparametric test included in this package is the Sign Test (subroutine SGNTST). The subroutine took a bivariate sample (or two matched samples) and counted the number of observations in which the first variate (sample) was smaller than its corresponding second variate (sample). The null hypothesis could state that this first variate was equal to, greater than, or less than the second variate, versus a suitable alternative. The probability of a true null hypothesis was determined by using the function BINOM which calculated the binomial distribution probability of the test statistic.

---

[12]Conover, pp. 370-371.

One of the measures of correlation incorporated into this package was Spearman's Rho (subroutine RHO). Spearman's Rho is a measure of correlation between the items in a bivariate sample, using ranks rather than the actual data values. The subroutine ranked the data by the first variate (the "X" items), then ranked the data on the second variate (the "Y" items). The differences in the "X" and "Y" ranks for all of the observations were used to calculate Rho, the measure of correlation. A test of positive, negative, or either positive or negative correlation was performed, depending on the user's request. Under the null hypothesis the probability of a Rho of a particular magnitude was determined by a large sample normal approximation (using function ANDTR). This was flagged as being questionable if less than 30 observations were present.

A second measure of correlation included in this package was Kendall's Tau (subroutine TAU). This test is a measure of rank correlation, like Spearman's Rho. The measure of correlation, Tau, was calculated by taking a bivariate sample and forming pairs of observations, each observation being an "X" and a "Y". If the X and Y of either of the two observations in a pair were both larger than the X and Y, respectively, of the other observation in the pair, the pair was counted as being concordant. If the X of one observation was larger than the X in its paired observation and the Y of that one observation was smaller than the Y in its paired observation, then the pair was counted as being discordant. Any pairs with ties were neither

concordant nor discordant and weren't counted. A test statistic and the value of Tau were calculated and the probability of no correlation versus some correlation, positive correlation, or negative correlation (according to the user's specifications) was determined from a large sample normal approximation, using function ANDTR. If the sample was less than 40 observations, a warning was printed, questioning the validity of the determined probability.

The Wilcoxon Signed Ranks Test (subroutine WILCOX) is used to test a bivariate sample or two matched samples for differences in the distributions of the two halves of the data. It can also test a single random sample against a single value to test the relation of the sample median to this specified value. The subroutine took the differences between variates, matched items, or observations and the specified value. These differences were ranked on their absolute values and the ranks of the positive differences were summed. This sum formed a test statistic which was used to compute the probability of a true null hypothesis, the null hypothesis stating that there were no differences in the distributions of the variates or samples, or that the median of the sample was the specified value. Since a large sample normal approximation was used to compute this probability a warning was printed if fewer than 20 observations were present.

The Test of Trend using Spearman's Rho (subroutine TREND) is used to test if any significant trend exists in a single

random sample. The direction of trend (upward, downward, or any direction) could be specified by the user. The subroutine took a single random sample and converted it into a bivariate sample by making the second item of each observation the observation number (1,2,3,...). Spearman's Rho was then calculated on this bivariate sample and used as a measure of trend. The probability of no trend and a Rho of the calculated magnitude was computed using the same large sample normal approximation that was used in the test of correlation using Spearman's Rho. Hence, a warning was printed if less than 30 observations were in the sample being tested.

No details on the actual computational formulas of test statistics and probabilities were given in the preceding paragraphs. To do so would be a rehash of a nonparametric statistics text. For details, the reader is referred to Conover and the appendix of the program listings.

DISTRIBUTION FUNCTIONS

An essential part of this package was the distribution function subprograms. These were used to determine the probability of test statistics and the hypothesized distributions for the Goodness of Fit Test. Some of the routines were written for this package while others were modified from other sources. Details on these routines follow.

Function ANDTR determined the probability of some random variable, distributed Normal(0,1), being less than or equal to

a specified parameter value, X. This function was a modified adaptation of the IBM SSP subroutine NDTR.[13] Besides changing the subroutine NDTR to the function ANDTR, a check had to be added to the subprogram to prevent exponent underflow[14] from occurring when trying to compute e^{-x}. If X is greater than 10 (or less than -10), the function would not compute the probability, but assumed that it was one (or zero).

Function BINOM determined the probability of some random variable distributed binomially with probability P and size N, being less than or equal to a specified parameter value Y. P and N were also specified as parameters. For N less than or equal to 20, the probability was calculated exactly, using a summation of the binomial expansion probabilities. If N was greater than 20, a normal approximation was used to determine the probability. There were two probabilities used in this function. One was a parameter and was the probability, P, of the binomial distribution. The other was calculated and represented the function value, the probability of a random variable being less than or equal to Y.

Function CDTR was a modification of the IBM SSP subroutine CDTR.[15] This function computed the probability of a random

---

[13] IBM SSP, p. 78.

[14] IBM Corporation, IBM System/360: Principles of Operation. (IBM System Reference Library, Form A22-6821-7, White Plains: IBM Corporation, 1968), pp. 50-51.

[15] IBM SSP, pp. 81-83.

variable, distributed with a chi-square distribution with G degrees of freedom, being greater than or equal to X. X and G were the calling arguments. This function called the function DLGAM, which was also a modification of an identically named subroutine from the SSP.[16] This function calculated the double precision log of gamma of the function argument. The function DLGAM was used only by function CDTR.

Function POIS calculated the probability of N or fewer events occurring from a population having a Poisson distribution with mean X. X and N were the calling arguments for the function. The function performed a summation from zero to N of that part of the probability function dependent only on N.[17] This summation was terminated if the amount contributed by any single term of the summation was less than 10^{-3} (.001). This sum was multiplied by e^{-X} to form the probability function value.

---

[16] IBM SSP, p. 362.

[17] Donald E. Knuth, The Art of Computer Programming, Vol. II, Seminumerical Algorithms (Reading: Addison-Wesley, 1969), p. 117.

Chapter 4

SUMMARY

This paper has presented a brief (at best) description of a nonparametric statistical test package for a digital computer, specifically the IBM System/360. This package was an attempt to fill a gap that was evident to the author in the area of statistical packages for digital computers. This package also attempted to give the user greater control over the test processing than previous packages have given.

A unique, perhaps, approach to presenting the statistical analysis to the user was used. Most packages observed by the author just printed statistical values, critical regions, probabilities, and other NUMBERS. This package attempted to present the testing in WORDS, by stating what the hypotheses being tested really were and by presenting the results also in words. Examples of this can be found in APPENDIX III. With this in mind, the user must select his analysis by thinking, "What is it that I am actually trying to show?" The author feels that this is one of the most important things the user must know.

Actual methods and techniques of test statistic computation and data handling were not presented. To do so in any detail would overburden this report with much trivia and repetition of sources that do a better job. The methods of computation were taken from Conover. The techniques used in the

actual programming can be extracted from APPENDICES I and II.

No explanation of the actual use of this package was made. It was felt by the author that it was better left to a User's Guide for the program, which would concern itself only with the use of the program. The original User's Guide written by the author is contained in APPENDIX IV of this report.

Nonparametric statistics are, in reality, very simple analysis techniques. Yet, it is felt by the author that many of the complex experimental situations that occur, require, in many cases, just this simple analysis to make the results valid. This nonparametric statistical package gives the complex experiment the simple analysis that is needed to give the experimenter faith in his results.

BIBLIOGRAPHY

Conover, W. J. Practical Nonparametric Statistics. Rough draft, Kansas State University, in press, 1970.

Feller, William. An Introduction to Probability Theory and Its Applications. Vol. I, 3rd ed. New York: Wiley, 1968.

International Business Machines Corporation. IBM System/360: FORTRAN IV Language. IBM Systems Reference Library, Form C28-6515-7. White Plains: International Business Machines Corporation, 1970.

_____. IBM System/360: FORTRAN IV Library Subprograms. IBM Systems Reference Library, Form C28-6596-4. White Plains: International Business Machines Corporation, 1968.

_____. IBM System/360 Operating System: FORTRAN IV (G and H) Programmer's Guide. IBM System Reference Library, Form C28-6817-0. White Plains: International Business Machines Corporation, 1967.

_____. IBM System/360 Operating System: FORTRAN IV (H) Compiler. IBM Program Logic, Form Y28-6642-3. White Plains: International Business Machines Corporation, 1969.

_____. IBM System/360 Operating System: Supervisor and Data Management Macro Instructions. IBM System Reference Library, Form C28-6647-3. White Plains: International Business Machines Corporation, 1968.

_____. IBM System/360: Principles of Operation. IBM System Reference Library, Form A22-6821-7. White Plains: International Business Machines Corporation, 1968.

_____. IBM System/360: Scientific Subroutine Package Version III, Programmer's Manual. IBM Application Program, Form H20-0205-3. White Plains: International Business Machines Corporation, 1969.

Knuth, Donald E. The Art of Computer Programming. Vol. II, Seminumerical Algorithms. Reading: Addison-Wesley, 1969.

APPENDIX I

FORTRAN PROGRAM LISTINGS

```
COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
      C                                                                      CNPSTP
      C     COMPLETE NON-PARAMETRIC STATISTICAL TESTS PACKAGE                CNPSTP
      C                                                                      CNPSTP
      C               AFFECTIONALLY KNOWN AS "SUPERSTAT"                     CNPSTP
      C                                                                      CNPSTP
      C               THIS PROGRAM WAS WRITTEN IN PARTIAL FULFILLMENT        CNPSTP
      C     OF THE DEGREE OF MASTER OF SCIENCE IN THE DEPARTMENT OF          CNPSTP
      C     STATISTICS AND COMPUTER SCIENCE, KANSAS STATE UNIVERSITY,        CNPSTP
      C     FALL, 1970, BY KEITH MOE.                                        CNPSTP
      C                                                                      CNPSTP
      C     THIS PROGRAM WILL PERFORM A NUMBER OF STATISTICAL TESTS ON       CNPSTP
      C     A VARIETY OF DATA, MUCH LIKE MANY OF THE STATISTICAL PACKAGES    CNPSTP
      C     THAT WERE IN EXISTANCE BEFORE THIS TIME.  THERE IS ONE,          CNPSTP
      C     VERY LARGE LIMITATION IN ALL OF THOSE PACKAGES.  THEY MAKE       CNPSTP
      C     THE ASSUMPTIONS NECESSARY TO PERFORM THE STATISTICAL TESTS       CNPSTP
      C     THAT THEY USE.  THE ASSUME THAT THE DISTRIBUTION OF THE DATA     CNPSTP
      C     TO BE TESTED IS KNOWN (USUALLY ASSUMED TO BE NORMALLY            CNPSTP
      C     DISTRIBUTED).                                                    CNPSTP
      C     THE STATISTICAL TESTS CHOOSEN FOR THIS PROGRAM ELIMINATE THAT    CNPSTP
      C     DISADVANTAGE, BECAUSE NONE OF THEM MAKE ANY ASSUMPTION ABOUT     CNPSTP
      C     THE DISTRIBUTION OF THE POPULATION BEING SAMPLED.  THESE         CNPSTP
      C     TESTS ARE USUALLY CALLED NON-PARAMETRIC TESTS.                   CNPSTP
      C     THE AUTHOR OF THIS PROGRAM FEEL THAT THE ASSUMPTION OF           CNPSTP
      C     NORMALITY IS, IN MANY  CASES, A VERY INVALID ASSUMPTION.         CNPSTP
      C     WHILE THESE TESTS ARE NOT AS POWERFUL AS PARAMETRIC TESTS ON     CNPSTP
      C     POPULATIONS THAT ARE NORMALLY DISTRIBUTED, WHEN THE DATA         CNPSTP
      C     IS NOT NORMAL, THESE TESTS ARE SOMETIMES INFINITELY MORE         CNPSTP
      C     POWERFUL THAN PARAMETRIC TECHNIQUES.                             CNPSTP
      C     THERE ARE MANY SITUATIONS, WHICH ARE OF INTEREST, IN WHICH       CNPSTP
      C     THE DATA IS NOT FROM A NORMAL POPULATION.  HENCE, THE USE        CNPSTP
      C     OF NON-PARAMETRIC TECHNIQUES IN THE ANALYSIS OF SUCH DATA        CNPSTP
      C     IS MANDATORY, IF THE USER DESIRES TO HAVE ANY CONFIDENCE         CNPSTP
      C     IN THE RESULTS OBTAINED.                                         CNPSTP
      C                                                                      CNPSTP
      C                                                                      CNPSTP
      C     COMMON BLOCKS                                                    CNPSTP
      C                                                                      CNPSTP
ISN 0002            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)           DECLARE
ISN 0003            INTEGER*4  ROWS,COLS                                   DECLARE
ISN 0004            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN               DECLARE
ISN 0005            INTEGER*4  DISTN,WHTRN                                 DECLARE
ISN 0006            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO            DECLARE
ISN 0007            REAL*8  BUFFER                                         DECLARE
```

```
ISN C006            INTEGER*4  EOD,CDCOL,BUFNO                                  DECLARE
ISN C009            COMMON /FORM/   FMT(100),IPR,IPS(40),OFORM,OOUT,RCBCD       DECLARE
ISN C010            REAL*8   FMT                                                DECLARE
ISN C011            INTEGER*4  OFORM,OOUT,RCBCD                                 DECLARE
ISN C012            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPOLS(40),  DECLARE
                   1     VALUE(40),IVALUE(40),RC                               DECLARE
ISN C013            REAL*8   PARM,SUBPRM                                        DECLARE
ISN C014            INTEGER*4  RC                                               DECLARE
ISN C015            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                 DECLARE
ISN C016            INTEGER*4  TAIL                                             DECLARE
ISN C017            LOGICAL*4  TESTS,DOALL                                      DECLARE
ISN C018            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),     DECLARE
                   1     DATE,TIME,PAGE,LINE                                   DECLARE
ISN C019            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME            DECLARE
ISN C020            INTEGER*4  PAGE                                             DECLARE
ISN C021            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
ISN C022            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE  DECLARE
ISN C023            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT         DECLARE
ISN C024            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT         DECLARE
ISN C025            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                    DECLARE
ISN C026            INTEGER*4  ITEMS,LOCAT,RDNO                                 DECLARE
ISN C027            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                DECLARE
ISN C028            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                      DECLARE
ISN C029            EQUIVALENCE (WORK(1),IWRK(1))                               DECLARE
ISN C030            REAL*8  BLANK /' '/                                         DECLARE
ISN C031            EQUIVALENCE (BLANK,IBLANK)                                  DECLARE
                C                                                               CNPSTP
                C     FORMAT STATEMENTS                                         CNPSTP
                C                                                               CNPSTP
ISN C032          101 FORMAT (1CA8)                                             CNPSTP
ISN C033          200 FORMAT ('0', T12, 'IKMS001  TOTAL EXECUTION TIME WAS',    CNPSTP
                   1     F7.2, ' SECONDS')                                      CNPSTP
ISN C034          219 FORMAT ('-', T12, 'IKMS191  INPUT BYPASSED UNTIL END OF DATA') CNPSTP
ISN 0035          299 FORMAT ('-', T12, 'IKMS991  PROGRAM ERROR.')              CNPSTP
                C                                                               CNPSTP
                C     MISCELLANEOUS DATA USED IN THIS PROGRAM                   CNPSTP
                C                                                               CNPSTP
ISN C036            REAL*8 ENDOF /'END OF D'/                                   CNPSTP
ISN C037            REAL*4  STITLE, TITL /'TITL'/                               CNPSTP
ISN C038            INTEGER*4  TSTART, TEND, RETCD                              CNPSTP
ISN 0039            DATA LIMIT   /0/                                            ********
                C     TIME LIMIT IS 0, WHICH IMPLIES 1 HOUR                     ********
                C                                                               CNPSTP
```

```
                C     DECLARE 'EODAD' EXTERNAL BECAUSE THIS ROUTINE IS USED IN          CNPSTP
                C     A SUBROUTINE CALL PARAMETER LIST, NAMELY, THAT TO ERRSET,         CNPSTP
                C     WHICH IS USED TO CATCH THE DATA CONVERSION ERROR WHICH THE        CNPSTP
                C     'END OF DATA' CARD CAUSES.                                        CNPSTP
                C                                                                       CNPSTP
ISN C040              EXTERNAL  EODAD                                                   CNPSTP
                C                                                                       CNPSTP
                C     PERFORM THE PROGRAM INITIALIZATION                                CNPSTP
                C                                                                       CNPSTP
                C     GET THE START TIME FOR PROGRAM TIMING                             CNPSTP
                C                                                                       CNPSTP
ISN C041              CALL INTIME (ISTART,LIMIT)                                        CNPSTP
                C                                                                       CNPSTP
                C     SET THE ERROR EXIT FOR THE DATA CONVERSION ERROR - IHC215I        CNPSTP
                C                                                                       CNPSTP
ISN C042              CALL ERRSET (215,  0, -1, 1, EODAD)                               CNPSTP
                C                                                                       CNPSTP
                C     SET THE I/O UNITS TO WHAT THIS INSTALLATION USES                  CNPSTP
                C                                                                       CNPSTP
ISN C043              CALL SETIO                                                        CNPSTP
                C                                                                       CNPSTP
                C     ZERO OUT THE PROGRAM RETURN CODE TO PASS BACK TO THE SYSTEM       CNPSTP
                C                                                                       CNPSTP
ISN C044              DO 21 I = 1, WKSZE                                                CNPSTP
ISN C045           21 IWRK(I) = 0                                                       CNPSTP
ISN C046              RETCD = 0                                                         CNPSTP
ISN C047              DATE = BLANK                                                      CNPSTP
ISN C048              PAGE = 0                                                          CNPSTP
ISN C049              ECD = 0                                                           CNPSTP
                C                                                                       CNPSTP
                C     READ A SET OF TEST PARAMETER                                      CNPSTP
                C                                                                       CNPSTP
ISN C050            1 CALL RDPARM                                                       CNPSTP
ISN C051              IF (RC) 2, 9, 5                                                   CNPSTP
                C                                                                       CNPSTP
                C     RC NEGATIVE, END OF JOB REQUESTED                                 CNPSTP
                C                                                                       CNPSTP
ISN C052            2 IF (RC .EQ. -1) GO TO 3                                           CNPSTP
                C     -2 SIGNALS A PREMATURE END OF FILE                                CNPSTP
ISN C054              IF (RETCD .LT. 4) RETCD = 4                                       CNPSTP
                C                                                                       CNPSTP
                C     END OF JOB                                                        CNPSTP
                C                                                                       CNPSTP
```

```
ISN C056          3 CALL OUTIME(TEND)                                          CNPSTP
ISN C057            TIME = (TEND - TSTART) / 100.                              CNPSTP
ISN C058            WRITE (LST,200) TIME                                       CNPSTP
                C     CHECK THE RETURN CODE AND STOP APPROPRIATELY              CNPSTP
ISN C059            IF (RETCD .EQ. 0) STOP                                     CNPSTP
ISN C061            RETCD = RETCD / 4                                          CNPSTP
ISN C062            GO TO (4, 8, 12, 16), RETCD                                CNPSTP
ISN C063          4 STOP 4                                                     CNPSTP
ISN C064          8 STOP 8                                                     CNPSTP
ISN C065         12 STOP 12                                                    CNPSTP
ISN C066         16 STOP 16                                                    CNPSTP
                C                                                             CNPSTP
                C     A NON-ZERO RETURN CODE WAS RETURNED                     CNPSTP
                C                                                             CNPSTP
ISN C067          5 IF (RETCD .LT. RC) RETCD = RC                             CNPSTP
ISN C069            CALL OFLOW (C)                                            CNPSTP
                C     DON'T BYPASS DATA IF REAL END OF FILE WAS ENCOUNTERED    CNPSTP
ISN C070            IF (EOD .EQ. 2) GO TO 3                                   CNPSTP
ISN C072            IF (EOD .GT. 0) GO TO 1                                   CNPSTP
ISN C074            WRITE (LST,219)                                           CNPSTP
                C                                                             CNPSTP
                C     READ DATA UNTIL THE END OF DATA CARD OR END OF FILE     CNPSTP
                C                                                             CNPSTP
ISN C075          6 READ (RDR,101,END=3) PARM                                 CNPSTP
ISN C076            STITLE = PARM(1)                                          CNPSTP
ISN C077            IF (STITLE .NE. TITL) GO TO 7                             CNPSTP
                C                                                             CNPSTP
                C     ACCEPT A TITLE CARD INSTEAD OF END OF DATA              CNPSTP
                C                                                             CNPSTP
ISN C079            EOD = 3                                                   CNPSTP
ISN C080            BUFNO = 1                                                 CNPSTP
ISN C081            DO 14 I = 1, 10                                           CNPSTP
ISN C082         14 BUFFER(I,1) = PARM(I)                                     CNPSTP
ISN C083            GO TO 1                                                   CNPSTP
ISN C084          7 IF (PARM(1) .NE. ENDOF) GO TO 6                           CNPSTP
                C     GO TO START OVER WITH NEXT TEST                         CNPSTP
ISN C086            GO TO 1                                                   CNPSTP
                C                                                             CNPSTP
                C     ON A ZERO RETURN CODE, PROGRAM CONTINUES WITH THIS TEST CNPSTP
                C                                                             CNPSTP
ISN C087          9 CALL ANPARM                                               CNPSTP
ISN C088            IF (RC) 13, 10, 5                                         CNPSTP
ISN C089         10 CALL RDDATA                                               CNPSTP
```

```
   ISN C090            IF (RC) 2, 11, 5                                          CNPSTP
                C                                                                CNPSTP
                C      CLEAR SORTED DATA SWITCH AND GO DO THE TESTS               CNPSTP
                C                                                                CNPSTP
   ISN C091        11  SUBS(1) = 0                                               CNPSTP
   ISN C092            SUBS(2) = 0                                               CNPSTP
   ISN C093            CALL DOTEST                                               CNPSTP
   ISN C094            IF (EOD .EQ. 2) GO TO 3                                   CNPSTP
   ISN C096            IF (RC) 2, 1, 5                                           CNPSTP
                C                                                                CNPSTP
                C      PROGRAM ERROR - NEGATIVE RETURN CODE FROM AN IMPOSSIBLE PLACE CNPSTP
                C                                                                CNPSTP
   ISN C097        13  WRITE (LST,299)                                           CNPSTP
   ISN C098            RETCD = 16                                                CNPSTP
   ISN C099            GO TO 3                                                   CNPSTP
                C                                                                CNPSTP
   ISN C100            END                                                       CNPSTP

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      99 ,PROGRAM SIZE =     1330

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                                   DATE  70.268/18.56.28

          COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN C0002              FUNCTION ANDTR (X)                                              ANDTR
                 C                                                                        ANDTR
                 C      THIS FUNCTION SUBPROGRAM COMPUTES THE PROBABILITY OF A RANDOM     ANDTR
                 C      VARIABLE, Y, DISTRIBUTED NORMALLY, BEING LESS THAN OR EQUAL      ANDTR
                 C      TO THE VALUE X.                                                  ANDTR
                 C                                                                       ANDTR
                 C      THE BASIS OF THIS ROUTINE IS IBM'S SSP, VERSION III.             ANDTR
                 C                                                                       ANDTR
   ISN C0003             REAL*8  XX                                                      ANDTR
   ISN C0004             REAL*4  VAL / 10. /                                             ANDTR
                 C                                                                       ANDTR
   ISN C0005             P = 1.                                                          ANDTR
   ISN C0006             AX = ABS(X)                                                     ANDTR
   ISN C0007             IF (AX .GE. VAL) GO TO 1                                        ANDTR
   ISN C0009             T=1.0/(1.0+0.2316419*AX)                                        ANDTR
   ISN C0010             XX = -X * X / 2.                                                ANDTR
   ISN C0011             D=0.3989423 * DEXP(XX)                                          ANDTR
                 C                                                                       ANDTR
   ISN C0012             P=1.0 -D*T*((((1.330274*T-1.821256)*T+1.781478)*T-0.3565638)*T+  ANDTR
                        1    0.3193815)                                                 ANDTR
                 C                                                                       ANDTR
   ISN C0013            1 IF (X .LT. 0.) P = 1. - P                                     ANDTR
   ISN C0015             ANDTR = P                                                       ANDTR
                 C                                                                       ANDTR
   ISN C0016             RETURN                                                          ANDTR
   ISN C0017             END                                                             ANDTR

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*      SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =       16 ,PROGRAM SIZE =      508

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002            SUBROUTINE ANPARM                                                ANPARM
                C                                                                     ANPARM
                C     THIS SUBROUTINE PERFORMS THE INITIAL ANALYSIS OF THE            ANPARM
                C     PARAMETERS READ BY THE SUBROUTINE 'RDPARM'.  IT CHECKS          ANPARM
                C     PRIMARILY FOR SPECIFICATION OF TESTS THAT ARE INCOMPATIBLE      ANPARM
                C     WITH THE TYPE OF DATA THAT IS TO BE TESTED.  IT ALSO PERFORMS   ANPARM
                C     MISCELLANEOUS CHECK FOR MISSING, AMBIGUOUS, OR CONFLICTING      ANPARM
                C     PARAMETER SPECIFICATIONS.                                       ANPARM
                C                                                                     ANPARM
                C                                                                     ANPARM
                C     COMMON BLOCKS                                                   ANPARM
                C                                                                     ANPARM
 ISN 0003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                     DECLARE
 ISN 0004            INTEGER*4  ROWS,COLS                                             DECLARE
 ISN 0005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                         DECLARE
 ISN 0006            INTEGER*4  DISTN,WHTRN                                           DECLARE
 ISN 0007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                      DECLARE
 ISN 0008            REAL*8  BUFFER                                                   DECLARE
 ISN 0009            INTEGER*4  EOD,CDCOL,BUFNO                                       DECLARE
 ISN 0010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD            DECLARE
 ISN 0011            REAL*8   FMT                                                     DECLARE
 ISN 0012            INTEGER*4  DFORM,DOUT,RCBCD                                      DECLARE
 ISN 0013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),      DECLARE
                    1    VALUE(40),IVALUE(40),RC                                      DECLARE
 ISN 0014            REAL*8   PARM,SUBPRM                                             DECLARE
 ISN 0015            INTEGER*4  RC                                                    DECLARE
 ISN 0016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                      DECLARE
 ISN 0017            INTEGER*4  TAIL                                                  DECLARE
 ISN 0018            LOGICAL*4  TESTS,DOALL                                           DECLARE
 ISN 0019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),          DECLARE
                    1    DATE,TIME,PAGE,LINE                                          DECLARE
 ISN 0020            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                 DECLARE
 ISN 0021            INTEGER*4  PAGE                                                  DECLARE
 ISN 0022            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
 ISN 0023            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE    DECLARE
 ISN 0024            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT              DECLARE
 ISN 0025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT              DECLARE
 ISN 0026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                         DECLARE
 ISN 0027            INTEGER*4  ITEMS,LOCAT,RDNO                                      DECLARE
 ISN 0028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                     DECLARE
 ISN 0029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                           DECLARE
```

```
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                                   DECLARE
ISN C031            REAL*8  BLANK /' '/                                             DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                      DECLARE
                C                                                                   ANPARM
                C     OTHER DATA USED IN THIS SUBROUTINE                            ANPARM
                C                                                                   ANPARM
ISN C033            LOGICAL*4  TRYTST                                               ANPARM
                C                                                                   ANPARM
                C     FORMAT STATEMENTS                                             ANPARM
                C                                                                   ANPARM
ISN C034          108 FORMAT ('-', T12, 'THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUANPARM
                   1PPLIED DATA')                                                   ANPARM
ISN C035          109 FORMAT ('-', T12, 'THE DATA SUPPLIED CONSISTS OF')            ANPARM
ISN C036          110 FORMAT (' ', T12, 'THE SAMPLE IS OF SIZE', I5)                ANPARM
ISN C037          111 FORMAT ('+', T42, '1 RANDOM SAMPLE')                          ANPARM
ISN C038          112 FORMAT ('+', T42, '2 RANDOM SAMPLES')                         ANPARM
ISN C039          113 FORMAT ('+', T42, I2, ' RANDOM SAMPLES')                      ANPARM
ISN C040          114 FORMAT ('+', T42, 'A BIVARIATE SAMPLE')                       ANPARM
ISN C041          115 FORMAT ('+', T42, 'A MULTIVARIATE SAMPLE OF MULTIPLICITY', I3) ANPARM
ISN C042          116 FORMAT ('+', T42, '2 MATCHED SAMPLES')                        ANPARM
ISN C043          117 FORMAT ('+', T42, 'A RANDOMIZED COMPLETE BLOCK, WITH',        ANPARM
                     1    I3, ' BLOCKS,', I3, ' TREATMENTS, AND',                  ANPARM
                     2    I3, ' REPLICATIONS')                                     ANPARM
ISN C044          118 FORMAT ('+', T42, 'AN INCOMPLETE BLOCK, WITH', I3,            ANPARM
                     1    ' BLOCKS AND', I3, ' TREATMENTS')                        ANPARM
ISN C045         1181 FORMAT (' ', T12, 'THERE ARE', I3, ' BLOCKS PER TREATMENT AND', ANPARM
                     1    I3, ' TREATMENTS PER BLOCK')                             ANPARM
ISN C046          119 FORMAT (' ', T12, 'THE SAMPLES ARE OF SIZES', T42, 7I10/      ANPARM
                     1    (T12, 10I10 /) )                                         ANPARM
ISN C047          121 FORMAT ('0', T12, 'THE SIGN TEST')                            ANPARM
ISN C048          122 FORMAT ('0', T12, 'THE MC NEMAR TEST')                        ANPARM
ISN C049          123 FORMAT ('0', T12, 'THE CHI SQUARE TEST OF DIFFERENCES - ',    ANPARM
                     1    I3, ' BY', I3)                                           ANPARM
ISN C050          124 FORMAT ('0', T12, 'THE CHI SQUARE TEST FOR INDEPENDENCE')     ANPARM
ISN C051          125 FORMAT ('0', T12, 'THE MEDIAN TEST')                          ANPARM
ISN C052          126 FORMAT ('0', T12, 'THE WILCOXON SIGNED RANK TEST')            ANPARM
ISN C053          127 FORMAT ('0', T12, 'THE MANN-WHITNEY TEST')                    ANPARM
ISN C054          128 FORMAT ('0', T12, 'MOOD''S TEST OF VARIANCE')                 ANPARM
ISN C055          129 FORMAT ('0', T12, 'CORRELATION USING SPEARMAN''S RHO')        ANPARM
ISN C056          130 FORMAT ('0', T12, 'TEST OF TREND USING SPEARMAN''S RHO')      ANPARM
ISN C057          131 FORMAT ('0', T12, 'CORRELATION USING KENDALL''S TAU')         ANPARM
ISN C058          132 FORMAT ('0', T12, 'THE KRUSKAL-WALLIS TEST')                  ANPARM
ISN C059          133 FORMAT ('0', T12, 'THE FRIEDMAN TEST')                        ANPARM
```

```
ISN C060        134 FORMAT ('0', T12, 'THE CHI-SQUARE GOODNESS OF FIT TEST')          ANPARM
ISN C061        135 FORMAT ('0', T12, 'THE DURBIN TEST')                              ANPARM
ISN C062        208 FORMAT ('-', T12, 'IKMS08I  INVALID NO. OF SAMPLES FOR SAMPLE TYPEANPARM
                   1     ')                                                           ANPARM
ISN C063        209 FORMAT ('-', T12, 'IKMS09I  INVALID TEST FOR TYPE OF DATA')       ANPARM
ISN C064        210 FORMAT ('-', T12, 'IKMS10I  MISSING OR INVALID RCB/BIBD PARAMETERSANPARM
                   1     ')                                                           ANPARM
ISN C065        211 FORMAT ('-', T12, 'IKMS11I  NO SAMPLE SIZES PROVIDED WHERE REQUIREANPARM
                  1D')                                                              ANPARM
ISN C066        212 FORMAT ('-', T12, 'IKMS12I  MISSING MEDIAN/LEVEL SPECIFICATION') ANPARM
ISN C067        213 FORMAT ('-', T12, 'IKMS13I  CLASS INFORMATION IMPROPERLY SPECIFIEDANPARM
                   1     ')                                                           ANPARM
ISN C068        214 FORMAT ('-', T12, 'IKMS14I  DISTRIBUTION PARAMETERS INVALID FOR SPANPARM
                  1ECIFIED TYPE OF DISTRIBUTION')                                   ANPARM
ISN C069        215 FORMAT ('-', T12, 'IKMS15I  EQUAL SAMPLE SIZES ASSUMED.')         ANPARM
ISN C070        216 FORMAT ('-', T12, 'IKMS16I  END OF DATA WILL DETERMINE SAMPLE SIZEANPARM
                  1S.')                                                             ANPARM
ISN C071        217 FORMAT ('-', T12, 'IKMS17I  TRANSFORMATION INVALID FOR THIS DATA')ANPARM
ISN C072        218 FORMAT ('-', T12, 'IKMS18I  NO TESTS SPECIFIED')                  ANPARM
ISN C073        299 FORMAT ('-', T12, 'IKMS99I  PROGRAM ERROR.')                      ANPARM
                C                                                                   ANPARM
                C     SELECT ROUTINE BY DATA TYPE                                   ANPARM
                C                                                                   ANPARM
ISN C074              WKSZE = WKSZE                                                 ANPARM
ISN C075              WRITE (LST,109)                                               ANPARM
ISN C076              GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 9), TYPE                    ANPARM
                C                                                                   ANPARM
                C     1 RANDOM SAMPLE                                               ANPARM
                C                                                                   ANPARM
ISN C077            1 WRITE (LST,111)                                               ANPARM
ISN C078              IF (SMPLS .EQ. 0) SMPLS = 1                                   ANPARM
ISN C080              IF (VARITE .EQ. 0) VARITE = 1                                 ANPARM
ISN C082              IF (SMPLS .NE. 1 .OR. VARITE .NE. 1) GO TO 1208               ANPARM
ISN C084              WRITE (LST,110) SSIZE(1)                                      ANPARM
ISN C085              CALL OFLOW(4)                                                 ANPARM
ISN C086              IF (SSIZE(1) .GT. 0) GO TO 11                                 ANPARM
ISN C088              WRITE (LST,216)                                               ANPARM
ISN C089              CALL OFLOW(3)                                                 ANPARM
                C                                                                   ANPARM
ISN C090           11 IF (.NOT. DOALL) GO TO 12                                     ANPARM
ISN C092              IF (ROWS .NE. 0 .AND. COLS .NE. 0) TESTS(4) = .TRUE.          ANPARM
ISN C094              TESTS(10) = .TRUE.                                            ANPARM
ISN C095              IF (DISTN .LT. 0) TESTS(6) = .TRUE.                           ANPARM
```

```
ISN C097            IF (DISTN .GT. 0) TESTS(14) = .TRUE.                        ANPARM
                C                                                               ANPARM
ISN C099           12 TRYTST = TESTS(2) .OR. TESTS(3) .OR. TESTS(5) .OR. TESTS(15)  ANPARM
ISN C100            DO 13 I = 7, 9                                              ANPARM
ISN C101           13 TRYTST = TRYTST .OR. TESTS(I) .OR. TESTS(I+4)             ANPARM
ISN C102            IF (TRYTST) GO TO 1209                                      ANPARM
ISN C104            GO TO 1000                                                  ANPARM
                C                                                               ANPARM
                C     2 RANDOM SAMPLES                                          ANPARM
                C                                                               ANPARM
ISN C105            2 WRITE (LST,112)                                           ANPARM
ISN C106            IF (SMPLS .EQ. 0) SMPLS = 2                                 ANPARM
ISN C108            IF (VARITE .EQ. 0) VARITE = 1                               ANPARM
ISN C110            IF (SMPLS .NE. 2 .OR. VARITE .NE. 1) GO TO 1208             ANPARM
ISN C112            IF (SSIZE(1) .EQ. 0 .AND. SSIZE(2) .NE. 0) GO TO 1211       ANPARM
ISN C114            IF (SSIZE(1) .NE. 0 .AND. SSIZE(2) .EQ. 0) GO TO 1211       ANPARM
ISN C116            WRITE (LST,119) SSIZE(1), SSIZE(2)                          ANPARM
ISN C117            CALL OFLOW(4)                                               ANPARM
ISN C118            IF (SSIZE(1) .NE. 0) GO TO 21                               ANPARM
ISN C120            WRITE (LST,215)                                             ANPARM
ISN C121            WRITE (LST,216)                                             ANPARM
ISN C122            CALL OFLOW(6)                                               ANPARM
                C                                                               ANPARM
ISN C123           21 IF (.NOT. DCALL) GO TO 22                                 ANPARM
ISN C125            TESTS(5) = .TRUE.                                           ANPARM
ISN C126            TESTS(7) = .TRUE.                                           ANPARM
ISN C127            TESTS(8) = .TRUE.                                           ANPARM
ISN C128            TESTS(12) = .TRUE.                                          ANPARM
ISN C129            IF (COLS .NE. 0) TESTS(3) = .TRUE.                          ANPARM
ISN C131           22 TRYTST = TESTS(1) .OR. TESTS(2) .OR. TESTS(4) .OR. TESTS(6)  ANPARM
ISN C132            DO 23 I = 9, 11                                             ANPARM
ISN C133           23 TRYTST = TRYTST .OR. TESTS(I) .OR. TESTS(I+4)             ANPARM
ISN C134            IF (TRYTST) GO TO 1209                                      ANPARM
ISN C136            GO TO 1000                                                  ANPARM
                C                                                               ANPARM
                C     K RANDOM SAMPLES                                          ANPARM
                C                                                               ANPARM
ISN C137            3 WRITE (LST,113) SMPLS                                     ANPARM
ISN C138            CALL OFLOW(3)                                               ANPARM
ISN C139            IF (SMPLS .LE. 2) GO TO 1208                                ANPARM
ISN C141            IF (VARITE .EQ. 0) VARITE = 1                               ANPARM
ISN C143            IF (VARITE .NE. 1) GO TO 1208                               ANPARM
ISN C145            WRITE (LST,119) (SSIZE(I), I = 1, SMPLS)                    ANPARM
```

```
ISN C146            K = 1                                                    ANPARM
ISN C147            IF (SMPLS .GT. 7) K = (SMPLS - 8) / 10 + 2               ANPARM
ISN C149            CALL OFLOW(K)                                            ANPARM
                C                                                            ANPARM
ISN C150            J = 0                                                    ANPARM
ISN C151            DO 31 I = 1, SMPLS                                       ANPARM
ISN C152         31 J = J + SSIZE(I)                                         ANPARM
ISN C153            IF (J .NE. 0) GO TO 32                                   ANPARM
ISN C155            WRITE (LST,215)                                          ANPARM
ISN C156            WRITE (LST,216)                                          ANPARM
ISN C157            CALL OFLOW(6)                                            ANPARM
ISN C158            GO TO 34                                                 ANPARM
                C                                                            ANPARM
ISN C159         32 DO 33 I = 1, SMPLS                                       ANPARM
ISN C160            IF (SSIZE(I) .EQ. 0) GO TO 1208                          ANPARM
ISN C162         33 CONTINUE                                                 ANPARM
                C                                                            ANPARM
ISN C163         34 IF (.NOT. DOALL) GO TO 35                                ANPARM
ISN C165            IF (COLS .NE. 0) TESTS(3) = .TRUE.                       ANPARM
ISN C167            TESTS(5) = .TRUE.                                        ANPARM
ISN C168            TESTS(12) = .TRUE.                                       ANPARM
                C                                                            ANPARM
ISN C169         35 TRYTST = TESTS(1) .OR. TESTS(2) .OR. TESTS(4) .OR. TESTS(13)  ANPARM
                  1    .OR. TESTS(14) .OR. TESTS(15)                         ANPARM
ISN C170            DO 36 I = 6, 11                                          ANPARM
ISN C171         36 TRYTST = TRYTST .OR. TESTS(I)                            ANPARM
ISN C172            IF (TRYTST) GO TO 1209                                   ANPARM
ISN C174            GO TO 1000                                               ANPARM
                C                                                            ANPARM
                C     BIVARIATE SAMPLE                                       ANPARM
                C                                                            ANPARM
ISN C175          4 WRITE (LST,114)                                          ANPARM
ISN C176            IF (SMPLS .EQ. 0) SMPLS = 1                              ANPARM
ISN C178            IF (VARITE .EQ. 0) VARITE = 2                            ANPARM
ISN C180            IF (SMPLS .NE. 1 .OR. VARITE .NE. 2) GO TO 1208          ANPARM
ISN C182            WRITE (LST,110) SSIZE(1)                                 ANPARM
ISN C183            CALL OFLOW(4)                                            ANPARM
ISN C184            IF (SSIZE(1) .GT. 0) GO TO 41                            ANPARM
ISN C186            WRITE (LST,216)                                          ANPARM
ISN C187            CALL OFLOW(3)                                            ANPARM
                C                                                            ANPARM
ISN C188         41 IF (.NOT. DOALL) GO TO 42                                ANPARM
ISN C190            TESTS(1) = .TRUE.                                        ANPARM
```

```
ISN C191            TESTS(2) = .TRUE.                                                    ANPARM
ISN C192            IF (ROWS .NE. 0 .AND. COLS .NE. 0) TESTS(4) = .TRUE.                 ANPARM
ISN C194            TESTS(6) = .TRUE.                                                    ANPARM
ISN C195            TESTS(9) = .TRUE.                                                    ANPARM
ISN C196            TESTS(11) = .TRUE.                                                   ANPARM
                C                                                                        ANPARM
ISN C197           42 TRYTST = TESTS(7) .OR. TESTS(8) .OR. TESTS(12) .OR. TESTS(3)       ANPARM
ISN C198              TRYTST =TRYTST .OR. TESTS(13) .OR. TESTS(14) .OR. TESTS(15)        ANPARM
ISN C199              TRYTST = TRYTST .OR. TESTS(5) .OR. TESTS(10)                       ANPARM
ISN C200              IF (TRYTST) GO TO 1209                                             ANPARM
ISN C202              GO TO 1000                                                         ANPARM
                C                                                                        ANPARM
                C     MULTIVARIATE SAMPLE                                                ANPARM
                C                                                                        ANPARM
ISN C203            5 WRITE (LST,115) VARITE                                             ANPARM
ISN C204              IF (SMPLS .EQ.0) SMPLS = 1                                         ANPARM.
ISN C206              IF (VARITE .LE. 2 .OR. SMPLS .NE. 1) GO TO 1208                    ANPARM
ISN C208              WRITE (LST,110) SSIZE(1)                                           ANPARM
ISN C209              CALL OFLOW(4)                                                      ANPARM
ISN C210              IF (SSIZE(1) .GT. 0) GO TO 51                                      ANPARM
ISN C212              WRITE (LST,216)                                                    ANPARM
ISN C213              CALL OFLOW(3)                                                      ANPARM
                C                                                                        ANPARM
ISN C214           51 IF (DOALL) TESTS(13) = .TRUE.                                      ANPARM
ISN C216              TRYTST = TESTS(14) .OR. TESTS(15)                                  ANPARM
ISN C217              DO 52 I = 1, 12                                                    ANPARM
ISN C218           52 TRYTST = TRYTST .OR. TESTS(I)                                      ANPARM
ISN C219              IF (TRYTST) GO TO 1209                                             ANPARM
ISN C221              GO TO 1000                                                         ANPARM
                C                                                                        ANPARM
                C     2 MATCHED SAMPLES                                                  ANPARM
                C                                                                        ANPARM
ISN C222            6 WRITE (LST,116)                                                    ANPARM
ISN C223              IF (SMPLS .EQ. 0) SMPLS = 2                                        ANPARM
ISN C225              IF (VARITE .EQ. 0) VARITE = 1                                      ANPARM
ISN C227              IF (SMPLS .NE. 2 .OR. VARITE .NE. 1) GO TO 1208                    ANPARM
ISN C229              WRITE (LST,110) SSIZE(1)                                           ANPARM
ISN C230              CALL OFLOW(4)                                                      ANPARM
ISN C231              SSIZE(2) = SSIZE(1)                                                ANPARM
ISN C232              IF (SSIZE(1) .GT. 0) GO TO 61                                      ANPARM
ISN C234              WRITE (LST,216)                                                    ANPARM
ISN C235              CALL OFLOW(3)                                                      ANPARM
                C                                                                        ANPARM
```

```
ISN C236          61 IF (.NOT. DOALL) GO TO 62                                    ANPARM
ISN C236             TESTS(1) = .TRUE.                                            ANPARM
ISN C239             TESTS(2) = .TRUE.                                            ANPARM
ISN C240             TESTS(6) = .TRUE.                                            ANPARM
ISN C241             IF (TESTS(3) .OR. TESTS(4) .OR. TESTS(5)) GO TO 1209         ANPARM
                C                                                                 ANPARM
ISN C243          62 DO 63 I = 7, 15                                              ANPARM
ISN C244             IF (TESTS(I)) GO TO 1209                                     ANPARM
ISN C246          63 CONTINUE                                                     ANPARM
ISN C247             GO TO 1000                                                   ANPARM
                C                                                                 ANPARM
                C     RANDOMIZED COMPLETE BLOCK                                   ANPARM
                C                                                                 ANPARM
ISN C248           7 SMPLS = BLK * TRT                                            ANPARM
ISN C249             VARITE = SMPLS                                               ANPARM
ISN C250             WRITE (LST,117) BLK, TRT, REP                                ANPARM
ISN C251             CALL OFLOW(3)                                                ANPARM
ISN C252             IF (.NOT. DOALL) GO TO 71                                    ANPARM
ISN C254             TESTS(5) = .TRUE.                                            ANPARM
ISN C255             TESTS(13) = .TRUE.                                           ANPARM
                C                                                                 ANPARM
ISN C256          71 TRYTST = TESTS(14) .OR. TESTS(15)                            ANPARM
ISN C257             DO 72 I = 1, 4                                               ANPARM
ISN C258          72 TRYTST = TRYTST .OR. TESTS(I) .OR. TESTS(I+5) .OR. TESTS(I+8) ANPARM
ISN C259             IF (TRYTST) GO TO 1209                                       ANPARM
ISN C261             IF (RCBCD .EQ. 0) RCBCD = 9                                  ANPARM
ISN C263             GO TO 1000                                                   ANPARM
                C                                                                 ANPARM
                C     INCOMPLETE BLOCK DESIGN (BALANCED OR OTHERWISE)             ANPARM
                C                                                                 ANPARM
ISN C264           8 SMPLS = TRT * BLKT                                           ANPARM
ISN C265             VARITE = BLK * TRTB                                          ANPARM
ISN C266             WRITE (LST,118) BLK, TRT                                     ANPARM
ISN C267             WRITE (LST,1181) BLKT, TRTB                                  ANPARM
ISN C268             CALL OFLOW(4)                                                ANPARM
ISN C269             IF (VARITE .NE. SMPLS) GO TO 1210                            ANPARM
ISN C271             IF (DOALL) TESTS(15) = .TRUE.                                ANPARM
                C                                                                 ANPARM
ISN C273             DO 81 I = 1, 14                                              ANPARM
ISN C274             IF (TESTS(I)) GO TO 1209                                     ANPARM
ISN C276          81 CONTINUE                                                     ANPARM
ISN C277             IF (RCBCD .EQ. 0) RCBCD = 9                                  ANPARM
ISN C279             GO TO 1000                                                   ANPARM
```

```
                C
                C       PROGRAM ERROR - BAD TEST SHOULD NEVER REACH THIS SUBROUTINE
                C
ISN C280            9 WRITE (LST,299)
ISN C281              CALL OFLOW(3)
ISN C282              RC = 16
ISN C283              RETURN
                C
                C       END OF DATA CHECKS - NOW CHECK TESTS
                C
ISN C284         1CCC IF (IPR .EQ. C) IPR = 1
ISN C286              WRITE (LST,1C8)
ISN C287              CALL OFLOW(3)
ISN C288              NTST = 0
ISN C289              I = 0
                C       LOOK UP NEXT TEST TO BE CHECKED/PRINTED
ISN C29C         1CC1 ITST = I + 1
ISN C291              DO 1002 I = ITST, 15
ISN 0292              IF (TESTS(I)) GO TO 1C03
ISN C294         1CC2 CONTINUE
ISN C295              GO TO 2100
                C
                C       FOUND A TEST - COUNT IT AND GO TO IT FOR LISTING
                C
ISN C296         1CC3 NTST = NTST + 1
ISN 0297              GO TO (2001, 2002, 2CC3, 2C04, 2005, 2006, 2007, 2008,
                     1     2009, 2010, 2011, 2012, 2013, 2014, 2015), I
                C
                C       SIGN TEST - NUMBER 1
                C
ISN C298         2CC1 WRITE (LST,121)
ISN C299              CALL OFLOW(2)
ISN C30C              GO TO 1CC1
                C
                C       MC NEMAR TEST - NUMBER 2
                C
ISN C301         2002 WRITE (LST,122)
ISN C3C2              CALL OFLOW(2)
ISN C303              GO TO 1001
                C
                C       CHI-SCUARE TEST OF DIFFERENCES - R BY C - TEST NUMBER 3
                C
ISN C3C4         2CC3 WRITE (LST,123) SMPLS, COLS
```

```
ISN C305            CALL OFLOW(2)                                          ANPARM
ISN C306            IF (COLS .LT. 2) GO TO 1213                            ANPARM
ISN C308            GO TO 1001                                             ANPARM
                C                                                          ANPARM
                C     CHI-SQUARE TEST FOR INDEPENDENCE - NUMBER 4          ANPARM
                C                                                          ANPARM
ISN C309         2004 WRITE (LST,124)                                      ANPARM
ISN C310            CALL OFLOW(2)                                          ANPARM
ISN C311            IF (ROWS .LT. 2 .OR. COLS .LT. 2) GO TO 1213           ANPARM
ISN C313            GO TO 1001                                             ANPARM
                C                                                          ANPARM
                C     MEDIAN TEST - NUMBER 5                               ANPARM
                C                                                          ANPARM
ISN C314         2005 WRITE (LST,125)                                      ANPARM
ISN C315            CALL OFLOW(2)                                          ANPARM
ISN C316            GO TO 1001                                             ANPARM
                C                                                          ANPARM
                C     WILCOXON SIGNED RANK TEST - NUMBER 6                 ANPARM
                C                                                          ANPARM
ISN C317         2006 WRITE (LST,126)                                      ANPARM
ISN C318            CALL OFLOW(2)                                          ANPARM
ISN C319            IF (TYPE .NE. 1) GO TO 1001                            ANPARM
ISN C321            IF (DISTN .NE. -1) GO TO 1212                          ANPARM
ISN C323            GO TO 1001                                             ANPARM
                C                                                          ANPARM
                C     MANN-WHITNEY TEST - NUMBER 7                         ANPARM
                C                                                          ANPARM
ISN C324         2007 WRITE (LST,127)                                      ANPARM
ISN C325            CALL OFLOW(2)                                          ANPARM
ISN C326            GO TO 1001                                             ANPARM
                C                                                          ANPARM
                C     MOOD'S TEST - NUMBER 8                               ANPARM
                C                                                          ANPARM
ISN C327         2008 WRITE (LST,128)                                      ANPARM
ISN C328            CALL OFLOW(2)                                          ANPARM
ISN C329            GO TO 1001                                             ANPARM
                C                                                          ANPARM
                C     CORRELATION - SPEARMAN'S RHO - NUMBER 9              ANPARM
                C                                                          ANPARM
ISN C330         2009 WRITE (LST,129)                                      ANPARM
ISN C331            CALL OFLOW(2)                                          ANPARM
ISN C332            GO TO 1001                                             ANPARM
                C                                                          ANPARM
```

```
                C     TREND WITH SPEARMAN'S RHO - NUMBER 10                    ANPARM
                C                                                              ANPARM
ISN 0333         2010 WRITE (LST,130)                                          ANPARM
ISN 0334              CALL OFLOW(2)                                            ANPARM
ISN 0335              GO TO 1001                                               ANPARM
                C                                                              ANPARM
                C     CORRELATION - KENDALL'S TAU - NUMBER 11                  ANPARM
                C                                                              ANPARM
ISN 0336         2011 WRITE (LST,131)                                          ANPARM
ISN 0337              CALL OFLOW(2)                                            ANPARM
ISN 0338              GO TO 1001                                               ANPARM
                C                                                              ANPARM
                C     KRUSKAL-WALLIS TEST - NUMBER 12                          ANPARM
                C                                                              ANPARM
ISN 0339         2012 WRITE (LST,132)                                          ANPARM
ISN 0340              CALL OFLOW(2)                                            ANPARM
ISN 0341              GO TO 1001                                               ANPARM
                C                                                              ANPARM
                C     FRIEDMAN TEST - NUMBER 13                                ANPARM
                C                                                              ANPARM
ISN 0342         2013 WRITE (LST,133)                                          ANPARM
ISN 0343              CALL OFLOW(2)                                            ANPARM
ISN 0344              GO TO 1001                                               ANPARM
                C                                                              ANPARM
                C     CHI-SQUARE GOODNESS OF FIT - NUMBER 14                   ANPARM
                C                                                              ANPARM
ISN 0345         2014 WRITE (LST,134)                                          ANPARM
ISN 0346              CALL OFLOW(2)                                            ANPARM
ISN 0347              IF (DISTN .LT. 1) GO TO 1214                             ANPARM
                C     ON THE GOODNESS OF FIT TEST, AT LEAST 3 DICHOTOMIZATION CLASSES ANPARM
                C     AND THE TYPE OF DISTRIBUTION TO BE 'FITTED' MUST BE SPECIFIED ANPARM
ISN 0349              IF (COLS .LT. 3) GO TO 1213                              ANPARM
ISN 0351              GO TO (2024, 2034, 2044, 2054), DISTN                    ANPARM
                C                                                              ANPARM
ISN 0352         2024 IF (DPARM2 .LE. 0.) GO TO 1214                           ANPARM
ISN 0354              GO TO 1001                                               ANPARM
                C                                                              ANPARM
ISN 0355         2034 IF (DPARM1 .LE. 0.) GO TO 1214                           ANPARM
ISN 0357              GO TO 1001                                               ANPARM
                C                                                              ANPARM
ISN 0358         2044 GO TO 1001                                               ANPARM
                C                                                              ANPARM
ISN 0359         2054 IF (DPARM1 .LE. 0.) GO TO 1214                           ANPARM
```

```
ISN C361            IF (DPARM2 .LE. 0. .OR. DPARM2 .GE. 1.) GO TO 1214                ANPARM
ISN C363            GO TO 1001                                                        ANPARM
                    C                                                                 ANPARM
                    C     DURBIN TEST - NUMBER 15                                     ANPARM
                    C                                                                 ANPARM
ISN C364             2015 WRITE (LST,135)                                             ANPARM
                    C     GO TO 1001 NOT NEEDED SINCE THIS IS THE LAST TEST           ANPARM
                    C                                                                 ANPARM
                    C     CHECK FOR AT LEAST 1 TEST                                   ANPARM
                    C                                                                 ANPARM
ISN C365             2100 IF (NTST .EQ. 0) GO TO 1218                                 ANPARM
                    C                                                                 ANPARM
                    C     CALL SUBROUTINE TO PRINT THE LABELS THAT THE USER SPECIFIED ANPARM
                    C                                                                 ANPARM
ISN C367            CALL PRLBLS                                                       ANPARM
                    C                                                                 ANPARM
                    C     CHECK TRANSFORMATION SPECIFICATION                          ANPARM
                    C                                                                 ANPARM
ISN C368            IF (WHTRN .NE. 1) GO TO 2102                                      ANPARM
ISN C370            IF (TYPE .EQ. 5 .OR. TYPE .GE. 7) GO TO 1217                      ANPARM
                    C                                                                 ANPARM
                    C     END OF PARAMETER ANALYSIS                                   ANPARM
                    C                                                                 ANPARM
ISN C372             2102 RC = 0                                                      ANPARM
ISN C373            RETURN                                                            ANPARM
                    C                                                                 ANPARM
                    C     ERRORS IN THE ANALYSIS OF THE PARAMETERS                    ANPARM
                    C     SEE THE CORRESPONDING FORMAT STATEMENTS FOR THE MEANING     ANPARM
                    C     OF EACH PARTICULAR ERROR ROUTINE.                           ANPARM
                    C                                                                 ANPARM
ISN C374             1208 WRITE (LST,208)                                             ANPARM
ISN C375            GO TO 1313                                                        ANPARM
                    C                                                                 ANPARM
ISN C376             1209 WRITE (LST,209)                                             ANPARM
ISN C377            GO TO 1313                                                        ANPARM
                    C                                                                 ANPARM
ISN C378             1210 WRITE (LST,210)                                             ANPARM
ISN C379            GO TO 1313                                                        ANPARM
                    C                                                                 ANPARM
ISN C380             1211 WRITE (LST,211)                                             ANPARM
ISN C381            GO TO 1313                                                        ANPARM
                    C                                                                 ANPARM
ISN C382             1212 WRITE (LST,212)                                             ANPARM
```

```
ISN C383            GO TO 1313                                              ANPARM
                C                                                           ANPARM
ISN C384         1213 WRITE (LST,213)                                       ANPARM
ISN C385            GO TO 1313                                              ANPARM
                C                                                           ANPARM
ISN C386         1214 WRITE (LST,214)                                       ANPARM
ISN C387            GO TO 1313                                              ANPARM
                C                                                           ANPARM
                C     IKMS15I AND IKMS16I ARE NOT CONSIDERED ERRORS         ANPARM
                C                                                           ANPARM
ISN C388         1217 WRITE (LST,217)                                       ANPARM
ISN C389            GO TO 1313                                              ANPARM
                C                                                           ANPARM
ISN C390         1218 WRITE (LST,218)                                       ANPARM
                C                                                           ANPARM
                C     SET ERROR RETURN CODE AND RETURN                      ANPARM
                C                                                           ANPARM
ISN C391         1313 CALL OFLOW(3)                                         ANPARM
ISN C392            RC = 8                                                  ANPARM
ISN C393            RETURN                                                  ANPARM
                C                                                           ANPARM
ISN C394            END                                                     ANPARM

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     393 ,PROGRAM SIZE =    .5922

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
  ISN 0002            FUNCTION BINOM (N,P,Y)                                        BINOM
                C                                                                   BINOM
                C     THIS FUNCTION SUBPROGRAM CALCULATES THE PROBABILITY OF SOME   BINOM
                C     RANDOM VARIABLE X BEING LESS THAN OR EQUAL TO Y, UNDER THE    BINOM
                C     BINOMIAL DISTRIBUTION FROM A POPULATION OF SIZE N AND         BINOM
                C     PROBABILITY DISTRIBUTION LEVEL P.                             BINOM
                C                                                                   BINOM
                C                                                                   BINOM
                C     NECESSARY TYPE AND PRECISION DECLARATIONS                     BINOM
                C                                                                   BINOM
  ISN 0003            INTEGER*4  Y                                                  BINOM
  ISN 0004            REAL*8  AP,AP1,PROB,PN,PN1,FN,FI,FNI                          BINOM
                C                                                                   BINOM
                C     CHECK THE SPECIFIED PARAMETERS                                BINOM
                C                                                                   BINOM
  ISN 0005            AP = P                                                        BINOM
  ISN 0006            AP1 = 1.D0 - AP                                               BINOM
  ISN 0007            PROB = 0.D0                                                   BINOM
  ISN 0008            IF (P .GE. 1. .OR. P .LE. 0.) GO TO 99                        BINOM
  ISN 0010            IF (Y .LT. 0) GO TO 99                                        BINOM
  ISN 0012            IF (N .LE. 0) GO TO 99                                        BINOM
                C                                                                   BINOM
  ISN 0014            PROB = 1.D0                                                   BINOM
  ISN 0015            IF (Y .GE. N) GO TO 99                                        BINOM
                C                                                                   BINOM
                C     CHECK DIVIDING LINE ON METHOD CALCULATION                     BINOM
                C                                                                   BINOM
  ISN 0017            IF (N .GT. 20) GO TO 51                                       BINOM
                C                                                                   BINOM
                C     CALCULATE THE PROBABILITY BY A SUMMATION                      BINOM
                C                                                                   BINOM
  ISN 0019         11 PROB = 0.D0                                                   BINOM
  ISN 0020            PN1 = 1.D0                                                    BINOM
  ISN 0021            DO 12 I = 1, N                                                BINOM
  ISN 0022         12 PN1 = PN1 * AP1                                               BINOM
                C                                                                   BINOM
  ISN 0023            PN = 1.D0                                                     BINOM
  ISN 0024            PROB = PN1                                                    BINOM
  ISN 0025            IF (Y .EQ. 0) GO TO 99                                        BINOM
                C                                                                   BINOM
  ISN 0027            DO 20 I = 1, Y                                                BINOM
```

```
ISN C028            PN = PN * AP                                          BINOM
ISN C029            PN1 = PN1 / AP1                                       BINOM
                C                                                         BINOM
ISN C030            FN = 1.DO                                             BINOM
ISN C031            FI = 1.DO                                             BINOM
ISN C032            FNI = 1.DO                                            BINOM
                C   N FACTORIAL                                           BINOM
ISN C033            DO 14 J = 1, N                                        BINOM
ISN C034         14 FN = FN * J                                           BINOM
                C   I FACTORIAL                                           BINOM
ISN C035            DO 15 J = 1, I                                        BINOM
ISN C036         15 FI = FI * J                                           BINOM
                C   (N-I) FACTORIAL                                       BINOM
ISN C037            M = N - I                                             BINOM
ISN C038            DO 16 J = 1, M                                        BINOM
ISN C039         16 FNI = FNI * J                                         BINOM
                C                                                         BINOM
                C   SUM UP THE PROBABILITY ASSOCIATED WITH THIS TERM       BINOM
                C                                                         BINOM
ISN C040            FN = FN / FI                                          BINOM
ISN C041         20 PROB = PROB + (FN / FNI) * PN * PN1                   BINOM
                C                                                         BINOM
ISN C042            GO TO 99                                              BINOM
                C                                                         BINOM
                C   CALCULATE USING THE NORMAL APPROXIMATION              BINOM
                C                                                         BINOM
ISN C043         51 FN = N                                                BINOM
ISN C044            FNI = DSQRT (FN * AP * AP1)                           BINOM
ISN C045            FX = (DFLOAT(Y) - FN * AP) / FNI                      BINOM
ISN C046            PROB = ANDTR(FX)                                      BINOM
                C                                                         BINOM
                C   SET THE FUNCTION EQUAL TO THE CALCULATED PROBSBILITY   BINOM
                C                                                         BINOM
ISN C047         99 BINOM = PROB                                          BINOM
                C                                                         BINOM
ISN C048            RETURN                                                BINOM
                C                                                         BINOM
ISN C049            END                                                   BINOM

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      48 ,PROGRAM SIZE =     1056
```

```
*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            FUNCTION CDTR(X,G)                                              CDTR
                C                                                                   CDTR
                C     THIS FUNCTION COMPUTES THE PROBABILITY THAT THE RANDOM        CDTR
                C     VARIABLE Y, DISTRIBUTED ACCORDING TO THE CHI-SQUARE           CDTR
                C     DISTRIBUTION WITH G DEGREES OF FREEDOM, IS LESS THAN OR       CDTR
                C     EQUAL TO X.  G IS A CONTINUOUS PARAMETER.                     CDTR
                C                                                                   CDTR
                C           ALPHAT = CDTR(X,G)                                      CDTR
                C                                                                   CDTR
                C     THIS FUNCTION IS AN ADAPTATION OF THE SUBROUTINE CDTR FROM    CDTR
                C     IBM'S SCIENTIFIC SUBROUTINE PACKAGE, VERSION III.  IT USES THECDTR
                C     FUNCTIONS ANDTR AND DLGAM, WHICH WERE ADAPTED FROM THE        CDTR
                C     SAME SOURCE.                                                  CDTR
                C                                                                   CDTR
                C     REFER TO THE IBM MANUAL H20-0205 FOR MORE DETAILS ON THESE    CDTR
                C     SUBROUTINES.                                                  CDTR
                C                                                                   CDTR
                C                                                                   CDTR
                C     COMMON BLOCKS                                                 CDTR
                C                                                                   CDTR
ISN C003            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),    CDTR
                   1    VALUE(40),IVALUE(40),RC                                     CDTR
ISN C004            REAL*8   PARM,SUBPRM                                          CDTR
ISN C005            INTEGER*4  RC                                                 CDTR
ISN C006            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT           CDTR
ISN C007            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT           CDTR
                C                                                                   CDTR
                C     ERROR FORMAT STATEMENTS                                       CDTR
                C                                                                   CDTR
ISN C008          297 FORMAT ('0', T12, 'IKMS97I  FAILURE TO CONVERGE IN CHI-SQ FUNCTIONCDTR
                   1    ')                                                          CDTR
ISN C009          298 FORMAT ('0', T12, 'IKMS98I  INVALID INPUT TO CHI-SQ FUNCTION')CDTR
                C                                                                   CDTR
                C     DECLARE NECESSARY DOUBLE PRECISION ITEMS                      CDTR
                C                                                                   CDTR
ISN C010            REAL*8  DLGAM                                                 CDTR
ISN C011            REAL*8  XX, DLXX, X2, DLX2, GG, G2, DLT3, THETA, THP1,        CDTR
                   1    GLG2, DD, T11, SER, CC, XI, FAC, TLOG, TERM, GTH, A2,      CDTR
                   2    A, B, C, DT2, DT3, THPI                                    CDTR
                C                                                                   CDTR
                C        TEST FOR VALID INPUT DATA                                  CDTR
```

```
                C                                                                CDTR
ISN 0012              IF(G-(.5-1.E-5)) 590,10,10                                 CDTR
ISN 0013           10 IF(G-2.E+5) 20,20,590                                      CDTR
ISN 0014           20 IF(X) 590,30,30                                            CDTR
                C                                                                CDTR
                C        TEST FOR X NEAR 0.0                                     CDTR
                C                                                                CDTR
ISN 0015           30 IF(X-1.E-8) 40,40,80                                       CDTR
ISN 0016           40 P=0.0                                                      CDTR
ISN 0017              GO TO 610                                                  CDTR
                C                                                                CDTR
                C        TEST FOR X GREATER THAN 1.E+6                           CDTR
                C                                                                CDTR
ISN 0018           80 IF (X .LE. 1.E6) GO TO 100                                 CDTR
ISN 0020              P=1.0                                                      CDTR
ISN 0021              GO TO 610                                                  CDTR
                C                                                                CDTR
                C        SET PROGRAM PARAMETERS                                  CDTR
                C                                                                CDTR
ISN 0022          100 XX=DBLE(X)                                                 CDTR
ISN 0023              DLXX=DLOG(XX)                                              CDTR
ISN 0024              X2=XX/2.D0                                                 CDTR
ISN 0025              DLX2=DLOG(X2)                                              CDTR
ISN 0026              GG=DBLE(G)                                                 CDTR
ISN 0027              G2=GG/2.D0                                                 CDTR
                C                                                                CDTR
                C        TEST FOR G GREATER THAN 1000.0                          CDTR
                C        TEST FOR X GREATER THAN 2000.0                          CDTR
                C                                                                CDTR
ISN 0028          150 IF(G-1000.) 160,160,180                                    CDTR
ISN 0029          160 IF(X-2000.) 190,190,170                                    CDTR
ISN 0030          170 P=1.0                                                      CDTR
ISN 0031              GO TO 610                                                  CDTR
ISN 0032          180 A=DLOG(XX/GG)/3.D0                                         CDTR
ISN 0033              A=DEXP(A)                                                  CDTR
ISN 0034              B=2.D0/(9.D0*GG)                                           CDTR
ISN 0035              C=(A-1.D0+B)/DSQRT(B)                                      CDTR
ISN 0036              SC=SNGL(C)                                                 CDTR
ISN 0037              P = ANDTR(SC)                                              CDTR
ISN 0038              GO TO 490                                                  CDTR
                C                                                                CDTR
                C        COMPUTE THETA                                           CDTR
                C                                                                CDTR
```

```
ISN C039        19C K= IDINT(G2)                                                CDTR
ISN C040            THETA=G2-DFLOAT(K)                                          CDTR
ISN C041            IF(THETA-1.D-8) 200,200,210                                 CDTR
ISN C042        2CC THETA=0.DO                                                  CDTR
ISN C043        21C THP1=THETA+1.DO                                             CDTR
                C                                                               CDTR
                C       SELECT METHOD OF COMPUTING T1                           CDTR
                C                                                               CDTR
ISN C044            IF(THETA)23C,230,220                                        CDTR
ISN C045        22C IF(XX-10.DO)260,260,320                                     CDTR
                COMPUTE T1 FOR THETA EQUALS 0.0                                 CDTR
ISN C046        23C IF(X2-1.68D02) 250,24C,24C                                  CDTR
ISN C047        24C T1=1.0                                                      CDTR
ISN C048            GO TO 400                                                   CDTR
ISN C049        25C T11=1.DO-DEXP(-X2)                                          CDTR
ISN C050            T1=SNGL(T11)                                                CDTR
ISN C051            GO TO 400                                                   CDTR
                C                                                               CDTR
                C       COMPUTE T1 FOR THETA GREATER THAN 0.0 AND               CDTR
                C       X LESS THAN OR EQUAL TO 10.0                            CDTR
                C                                                               CDTR
ISN C052        26C SER=X2*(1.DO/THP1 -X2/(THP1+1.DO))                          CDTR
ISN C053            J=+1                                                        CDTR
ISN C054            CC=DFLOAT(J)                                                CDTR
ISN C055            DO 270 IT1=3,30                                             CDTR
ISN C056            XI=DFLOAT(IT1)                                              CDTR
ISN C057            FAC = DLGAM(XI)                                             CDTR
ISN C058            TLOG= XI*DLX2-FAC-DLOG(XI+THETA)                            CDTR
ISN C059            TERM=DEXP(TLOG)                                             CDTR
ISN C060            TERM=DSIGN(TERM,CC)                                         CDTR
ISN C061            SER=SER+TERM                                                CDTR
ISN C062            CC=-CC                                                      CDTR
ISN C063            IF(DABS(TERM)-1.D-9) 280,27C,270                            CDTR
ISN C064        27C CONTINUE                                                    CDTR
ISN C065            GO TO 600                                                   CDTR
ISN C066        28C IF(SER) 600,600,290                                         CDTR
ISN C067        29C GTH = DLGAM(THP1)                                           CDTR
ISN C068            TLOG=THETA*DLX2+DLOG(SER)-GTH                               CDTR
ISN C069            IF(TLOG+1.68D02) 3CC,300,31C                                CDTR
ISN C070        3CC T1=0.0                                                      CDTR
ISN C071            GO TO 400                                                   CDTR
ISN C072        31C T11=DEXP(TLOG)                                              CDTR
ISN C073            T1=SNGL(T11)                                                CDTR
```

```
ISN C074            GO TO 400                                                    CDTR
                C                                                                CDTR
                C       COMPUTE T1 FOR THETA GREATER THAN 0.0 AND                CDTR
                C       X GREATER THAN 10.C AND LESS THAN 2000.0                 CDTR
                C                                                                CDTR
ISN C075          32C A2=0.DO                                                    CDTR
ISN C076            DO 340 I=1,25                                                CDTR
ISN C077            XI=DFLOAT(I)                                                 CDTR
ISN C078            GTH = DLGAM(THP1)                                            CDTR
ISN C079            T11=-(13.DO*XX)/XI +THP1*DLOG(13.DO*XX/XI) -GTH-DLOG(XI)     CDTR
ISN C080            IF(T11+1.68D02) 340,340,330                                  CDTR
ISN C081          33C T11=DEXP(T11)                                              CDTR
ISN C082            A2=A2+T11                                                    CDTR
ISN C083          34C CONTINUE                                                   CDTR
ISN C084            A=1.01282051+THETA/15C.DO-XX/312.DO                          CDTR
ISN C085            B=DABS(A)                                                    CDTR
ISN C086            C= -X2+THP1*DLX2+DLOG(B)-GTH-3.951243718581427               CDTR
ISN C087            IF(C+1.68D02) 370,370,350                                    CDTR
ISN C088          35C IF (A) 360,370,380                                         CDTR
ISN C089          36C C=-DEXP(C)                                                 CDTR
ISN C090            GO TO 390                                                    CDTR
ISN C091          37C C=0.DO                                                     CDTR
ISN C092            GO TO 390                                                    CDTR
ISN 0093          38C C=DEXP(C)                                                  CDTR
ISN C094          39C C=A2+C                                                     CDTR
ISN C095            T11=1.DO-C                                                   CDTR
ISN 0096            T1=SNGL(T11)                                                 CDTR
                C                                                                CDTR
                C       SELECT PROPER EXPRESSION FOR P                           CDTR
                C                                                                CDTR
ISN C097          4CC IF(G-2.) 420,410,410                                       CDTR
ISN C098          41C IF(G-4.) 450,460,460                                       CDTR
                C                                                                CDTR
                C       COMPUTE P FOR G GREATER THAN ZERO AND LESS THAN 2.0      CDTR
                C                                                                CDTR
ISN C099          42C GTH = DLGAM(THP1)                                          CDTR
ISN 0100            DT2=THETA*DLXX-X2-THP1*.6931471805599453 -GTH                CDTR
ISN C101            IF(DT2+1.68D02) 430,430,440                                  CDTR
ISN C102          43C P=T1                                                       CDTR
ISN C103            GO TO 490                                                    CDTR
ISN 01C4          44C DT2=DEXP(DT2)                                              CDTR
ISN C105            T2=SNGL(DT2)                                                 CDTR
ISN C106            P=T1+T2+T2                                                   CDTR
```

```
ISN C107            GO TO 490                                                   CDTR
                C                                                               CDTR
                C       COMPUTE P FOR G GREATER THAN OR EQUAL TO 2.0            CDTR
                C       AND LESS THAN 4.0                                       CDTR
                C                                                               CDTR
ISN C108          45C P=T1                                                      CDTR
ISN C109            GO TO 490                                                   CDTR
                C                                                               CDTR
                C       COMPUTE P FOR G GREATER THAN OR EQUAL TO 4.0            CDTR
                C       AND LESS THAN OR EQUAL TO 1000.0                        CDTR
                C                                                               CDTR
ISN C110          46C DT3=0.D0                                                  CDTR
ISN C111            DO 480 I3=2,K                                               CDTR
ISN C112            THPI=DFLOAT(I3)+THETA                                       CDTR
ISN C113            GTH = DLGAM(THPI)                                           CDTR
ISN C114            DLT3=THPI*DLX2-DLXX-X2-GTH                                  CDTR
ISN C115            IF(DLT3+1.68D02) 480,480,470                                CDTR
ISN C116          47C DT3=DT3+DEXP(DLT3)                                        CDTR
ISN C117          48C CONTINUE                                                  CDTR
ISN C118            T3=SNGL(DT3)                                                CDTR
ISN C119            P=T1-T3-T3                                                  CDTR
                C                                                               CDTR
                C       SET ERROR INDICATOR                                     CDTR
                C                                                               CDTR
ISN C120          49C IF(P) 500,520,520                                         CDTR
ISN C121          50C IF(ABS(P)-1.E-7) 510,510,600                              CDTR
                C                                                               CDTR
ISN C122          51C P=0.0                                                     CDTR
ISN C123            GO TO 610                                                   CDTR
                C                                                               CDTR
ISN C124          52C IF(1.-P) 530,550,550                                      CDTR
ISN C125          53C IF(ABS(1.-P)-1.E-7) 540,540,600                           CDTR
                C                                                               CDTR
ISN C126          540 P=1.0                                                     CDTR
ISN C127            GO TO 610                                                   CDTR
                C                                                               CDTR
ISN C128          55C IF(P-1.E-8) 560,560,570                                   CDTR
                C                                                               CDTR
ISN C129          56C P=0.0                                                     CDTR
ISN C130            GO TO 610                                                   CDTR
                C                                                               CDTR
ISN C131          57C IF((1.0-P)-1.E-8) 580,580,610                             CDTR
                C                                                               CDTR
```

```
ISN C132        58C P=1.0                                                       CDTR
ISN C133            GO TO 610                                                   CDTR
                C                                                               CDTR
ISN C134        59C IER=-1                                                      CDTR
ISN C135            GO TO 620                                                   CDTR
                C                                                               CDTR
ISN C136        60C IER=+1                                                      CDTR
ISN C137            GO TO 620                                                   CDTR
                C                                                               CDTR
ISN C138        61C IER=0                                                       CDTR
                C                                                               CDTR
ISN C139        62C CDTR=1.0-P                                                  CDTR
ISN 0140            IF (IER) 910, 990, 92C                                      CDTR
                C                                                               CDTR
                C               IER=-1 --- AN INPUT PARAMETER IS INVALID.  X IS LESS   CDTR
                C                          THAN C.C, OR G IS LESS THEN 0.5 OR GREATER  CDTR
                C                          THAN 2*10**5.                        CDTR
                C                                                               CDTR
ISN C141        91C WRITE (LST,298)                                             CDTR
ISN C142            GO TO 930                                                   CDTR
                C                                                               CDTR
                C               IER=+1 --- INVALID OUTPUT.  P IS LESS THAN ZERO OR     CDTR
                C                          GREATER THAN ONE, OR SERIES FOR TI (SEE     CDTR
                C                          MATHEMATICAL DESCRIPTION) HAS FAILED TO     CDTR
                C                          CONVERGE.                            CDTR
                C                                                               CDTR
ISN C143        920 WRITE (LST,297)                                             CDTR
                C                                                               CDTR
ISN C144        93C RC = 12                                                     CDTR
ISN C145            CDTR = 0.                                                   CDTR
ISN C146            CALL OFLOW(2)                                               CDTR
                C                                                               CDTR
ISN C147        99C RETURN                                                      CDTR
                C                                                               CDTR
ISN C148            END                                                         CDTR

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     147 ,PROGRAM SIZE =     2962

*STATISTICS*  NO  DIAGNOSTICS GENERATED
```

****** END OF COMPILATION ******

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            SUBROUTINE CHINDP                                                    CHINDP
                    C                                                                    CHINDP
                    C     THE CHI-SQUARE TEST FOR INDEPENDENCE                           CHINDP
                    C                                                                    CHINDP
                    C     THE CHI-SQUARE TEST FOR INDEPENDENCE IS USED TO TEST THAT WHEN  CHINDP
                    C     DATA IS CLASSIFIED BY 2 DIFFERENT CRITERIA, THE PROBABILITY    CHINDP
                    C     OF BEING CLASSIFIED INTO SOME "ROW" CATAGORY IS INDEPENDENT    CHINDP
                    C     OF THE PROBABILITY OF BEING CLASSIFIED INTO SOME "COLUMN"      CHINDP
                    C     CATAGORY.                                                      CHINDP
                    C                                                                    CHINDP
                    C     THE DATA IS DICHOTOMIZED INTO ROWS AND COLUMNS BY LEVELS       CHINDP
                    C     THAT THE USER MUST PROVIDE ON THE CLASSES,ROWS AND THE         CHINDP
                    C     CLASSES,COLUMNS CARDS.  THIS IS USED TO FORM AN "R" BY "C"     CHINDP
                    C     CONTINGENCY TABLE.  IF A SINGLE SAMPLE IS PROVIDED AS DATA,    CHINDP
                    C     THE SINGLE DATA VALUE FROM EACH OBSERVATION IS USED TO         CHINDP
                    C     CLASSIFY THE DATA BY BOTH ROW AND COLUMN.  HOWEVER, THE USER   CHINDP
                    C     CAN PROVIDE A SECOND VALUE FOR EACH OBSERVATION VIA THE        CHINDP
                    C     TRANS SUBROUTINE.  IN THIS CASE, THE "ORIGINAL" DATA ITEM      CHINDP
                    C     (POSSIBLY TRANSFORMED) IS USED TO CLASSIFY BY ROW AND THE      CHINDP
                    C     "NEW" DATA ITEM IS USED TO CLASSIFY BY COLUMNS.  IF THE        CHINDP
                    C     SAMPLE PROVIDED BY THE USER IS A BIVARIATE SAMPLE, THE FIRST   CHINDP
                    C     VARAITE IS USED FOR ROW CLASSIFICATION AND THE SECOND IS       CHINDP
                    C     USED FOR COLUMN CLASSIFICATION.  THESE VALUES CAN ALSO         CHINDP
                    C     BE ALTERED BY THE SUBROUTINE TRANS.                            CHINDP
                    C                                                                    CHINDP
                    C     THE CRITICAL REGION IS DETERMINED FROM THE CHI-SQUARE          CHINDP
                    C     DISTRIBUTION, AS THE NAME OF THE TEST IMPLIES.                 CHINDP
                    C                                                                    CHINDP
                    C     FOR FURTHER INFORMATION, SEE CONOVER, W.J., PRACTICAL          CHINDP
                    C     NON-PARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 247-261.         CHINDP
                    C                                                                    CHINDP
                    C                                                                    CHINDP
                    C     COMMON BLOCKS                                                  CHINDP
                    C                                                                    CHINDP
ISN C003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                         DECLARE
ISN C004            INTEGER*4  ROWS,COLS                                                 DECLARE
ISN C005            COMMON /DIST/  DISTN,CPARM1,CPARM2,WHTRN                             DECLARE
ISN C006            INTEGER*4  DISTN,WHTRN                                               DECLARE
ISN C007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                          DECLARE
ISN C008            REAL*8  BUFFER                                                       DECLARE
ISN C009            INTEGER*4  EOD,CDCOL,BUFNO                                           DECLARE
```

```
ISN C010            COMMON /FORM/    FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD                DECLARE
ISN C011            REAL*8   FMT                                                          DECLARE
ISN C012            INTEGER*4  DFORM,DOUT,RCBCD                                           DECLARE
ISN C013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),           DECLARE
                   1    VALUE(40),IVALUE(40),RC                                          DECLARE
ISN C014            REAL*8   PARM,SUBPRM                                                  DECLARE
ISN C015            INTEGER*4  RC                                                         DECLARE
ISN C016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                           DECLARE
ISN C017            INTEGER*4  TAIL                                                       DECLARE
ISN C018            LOGICAL*4  TESTS,DOALL                                                DECLARE
ISN C019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),               DECLARE
                   1    DATE,TIME,PAGE,LINE                                              DECLARE
ISN C020            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                      DECLARE
ISN C021            INTEGER*4  PAGE                                                       DECLARE
ISN C022            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)     DECLARE
ISN C023            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE         DECLARE
ISN C024            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                   DECLARE
ISN C025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                   DECLARE
ISN C026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                              DECLARE
ISN C027            INTEGER*4  ITEMS,LOCAT,RDNO                                           DECLARE
ISN C028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                          DECLARE
ISN C029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                                DECLARE
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                                         DECLARE
ISN C031            REAL*8  BLANK /' '/                                                   DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                            DECLARE
                C                                                                         CHINDP
                C     OTHER DATA NEEDED TO BE DECLARED                                    CHINDP
                C                                                                         CHINDP
ISN C033            REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                            CHINDP
ISN C034            REAL*4  LOVAL /-1.E75/, HIVAL /+1.E75/                                CHINDP
ISN C035            INTEGER*2  H0 /'H0'/, H1 /'H1'/                                       CHINDP
ISN C036            REAL*8  ROWLBL /'ROW'/, COLLBL /'COLUMN'/                             CHINDP
ISN C037            REAL*8  HYPLBL(2) /'   IS   ', ' IS NOT '/                            CHINDP
ISN C038            INTEGER*4  CURSMP,CURVAR,CURCNT                                       CHINDP
                C                                                                         CHINDP
                C     FORMAT STATEMENTS                                                   CHINDP
ISN C039          231 FORMAT ('-', T12, 'IKMS31I  UNEQUAL SAMPLE/VARIATE SIZES AFTER TRACHINDP
                   INSFORMATION')                                                         CHINDP
ISN C040          232 FORMAT ('-', T12, 'IKMS32I  TOO FEW ', A8, ' CLASSES SPECIFIED')  CHINDP
ISN C041          233 FORMAT ('-', T12, 'IKMS33I  TOO FEW ', A8, ' DICHOTOMIZATION LEVELCHINDP
                   IS SPECIFIED')                                                         CHINDP
ISN C042          3C1 FORMAT ('-', T50, 'THE CHI-SQUARE TEST FOR INDEPENDENCE')         CHINDP
ISN C043          3C2 FORMAT ('0', T12, A2, ': THE CLASSIFICATION OF AN OBSERVATION BY 'CHINDP
```

```
                1    , 2A8, 'INDEPENDENT OF THE CLASSIFICATION BY ', A8)                CHINDP
ISN C044       303 FORMAT ('0', T12, 'THE SAMPLE IS DICHOTOMIZED BY ', A8, ' INTO',     CHINDP
                1    I3, ' CLASSES AS FOLLOWS:' //                                      CHINDP
                2    (' ', T12, 4(G10.4, ' TO ', G10.4, 5X) ) )                         CHINDP
ISN C045       304 FORMAT ('0', T12, 'THE SAMPLE WAS DISTRIBUTED AS FOLLOWS:' /         CHINDP
                1    '0', 8X, 8X, A8, 2015)                                             CHINDP
ISN C046       305 FORMAT (' ', A8, ' TOTALS ', 8X, 2015)                               CHINDP
ISN C047       306 FORMAT (' ', I4, I9,  11X, 2015)                                     CHINDP
ISN C048       307 FORMAT ('0', T12, 'THE SAMPLE SIZE IS', I6)                          CHINDP
ISN C049       308 FORMAT ('0', T12, 'VERSUS')                                          CHINDP
ISN C050       309 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5, ' WITH',          CHINDP
                1    I4, ' DEGREES OF FREEDOM')                                         CHINDP
ISN C051       310 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')           CHINDP
ISN C052       311 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                               CHINDP
ISN C053       312 FORMAT ('-', T46, 'END OF THE CHI-SQUARE TEST FOR INDEPENDENCE')     CHINDP
               C                                                                        CHINDP
               C     WRITE TITLE, FIX LABELS, WRITE HYPOTHESES                          CHINDP
               C                                                                        CHINDP
ISN C054             WKSZE = WKSZE                                                      CHINDP
ISN C055             WRITE (LST,301)                                                    CHINDP
ISN C056             IF (LABEL(2) .EQ. BLANK) LABEL(2) = LABEL(1)                       CHINDP
ISN C058             IF (LABEL(1) .EQ. BLANK) LABEL(1) = ROWLBL                         CHINDP
ISN C060             IF (LABEL(2) .EQ. BLANK) LABEL(2) = COLLBL                         CHINDP
ISN C062             WRITE (LST,302) HO,LABEL(1),HYPLBL(1),LABEL(2)                     CHINDP
ISN C063             WRITE (LST,308)                                                    CHINDP
ISN C064             WRITE (LST,302) H1,LABEL(1),HYPLBL(2),LABEL(2)                     CHINDP
ISN C065             CALL OFLOW(9)                                                      CHINDP
               C                                                                        CHINDP
               C     CHECK ROW SPECIFICATIONS                                           CHINDP
               C                                                                        CHINDP
ISN C066             IWCH = 1                                                           CHINDP
ISN C067             IF (ROWS .LT. 2) GO TO 1232                                        CHINDP
ISN C069             DATA(1) = LOVAL                                                    CHINDP
ISN C070             DO 11 I = 1, ROWS                                                  CHINDP
ISN C071          11 DATA(I+1) = RLEV(I)                                                CHINDP
ISN C072             DATA(ROWS+1) = HIVAL                                               CHINDP
               C                                                                        CHINDP
ISN C073             DO 12 I = 1, ROWS                                                  CHINDP
ISN C074             IF (DATA(I) .GE. DATA(I+1)) GO TO 1233                             CHINDP
ISN C076          12 CONTINUE                                                           CHINDP
               C                                                                        CHINDP
               C     CHECK COLUMN SPECS                                                 CHINDP
               C                                                                        CHINDP
```

```
ISN C077            IWCH = 2                                                    CHINDP
ISN C078            IF (COLS .LT. 2) GO TO 1232                                 CHINDP
ISN C080            DATA(26) = LOVAL                                            CHINDP
ISN C081            DO 13 I = 1, CCLS                                           CHINDP
ISN C082         13 DATA(I+26) = CLEV(I)                                        CHINDP
ISN C083            DATA(COLS+26) = HIVAL                                       CHINDP
                 C                                                              CHINDP
ISN C084            DO 14 I = 1, COLS                                           CHINDP
ISN C085            IF (DATA(I+25) .GE. DATA(I+26)) GO TO 1233                  CHINDP
ISN C087         14 CONTINUE                                                    CHINDP
                 C                                                              CHINDP
                 C     PRINT THE DICHOTOMIZATION LEVELS                         CHINDP
                 C                                                              CHINDP
ISN C088            WRITE (LST,303) LABEL(1),ROWS, (DATA(I), DATA(I+1), I = 1, ROWS)  CHINDP
ISN C089            J = 2. + (ROWS + 3) / 4                                     CHINDP
ISN C090            CALL OFLOW(J)                                               CHINDP
ISN C091            WRITE (LST,303) LABEL(2),COLS, (DATA(I+25), DATA(I+26),     CHINDP
                   1     I = 1, CCLS)                                           CHINDP
ISN C092            J = 2 + (COLS + 3) / 4                                      CHINDP
ISN C093            CALL OFLOW(J)                                               CHINDP
                 C                                                              CHINDP
                 C     INITIALIZE COUNTERS                                      CHINDP
                 C                                                              CHINDP
ISN C094            NSZ = 0                                                     CHINDP
ISN C095            ISZ = ROWS * CCLS                                           CHINDP
ISN C096            NRC = ISZ                                                   CHINDP
ISN C097            NCC = ISZ + ROWS                                            CHINDP
ISN C098            II = NCC + CCLS                                             CHINDP
ISN C099            DO 15 I = 1, II                                             CHINDP
ISN C100         15 IWRK(I) = 0                                                 CHINDP
ISN C101            IF (TYPE .EQ. 4) GO TO 31                                   CHINDP
                 C                                                              CHINDP
                 C     1 RANDOM SAMPLE                                          CHINDP
                 C                                                              CHINDP
ISN C103            REWIND UT1                                                  CHINDP
                 C                                                              CHINDP
ISN 0104         21 READ (UT1,END=81) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)       CHINDP
ISN C105            ITEMS = 1                                                   CHINDP
ISN C106            LOCAT(4) = 0                                                CHINDP
                 C                                                              CHINDP
                 C     DO USER TRANSFORMATION AND SAVING OF DATA                CHINDP
                 C                                                              CHINDP
ISN C107            IF (WHTRN .EQ. 1) CALL TRANS                                CHINDP
```

```
ISN C109            IF (ITEMS .LT. 1) GO TO 21                                  CHINDP
ISN C111            IF (ITEMS .EQ. 1) X(2) = X(1)                               CHINDP
ISN C113            IF (OUT .EQ. 0 .OR. DOUT .NE. 2) GO TO 22                   CHINDP
ISN C115            ITEMS = 1                                                   CHINDP
ISN C116            CALL WRDATA                                                 CHINDP
                C                                                               CHINDP
                C     CLASSIFY INTO A ROW                                       CHINDP
                C                                                               CHINDP
ISN C117         22 DO 23 I = 1, ROWS                                           CHINDP
ISN C118            IF (X(1) .LE. DATA(I+1)) GO TO 24                           CHINDP
ISN C120         23 CONTINUE                                                    CHINDP
ISN C121            GO TO 21                                                    CHINDP
                C                                                               CHINDP
ISN C122         24 IROW = I                                                    CHINDP
                C                                                               CHINDP
                C     CLASSIFY INTO A COLUMN                                    CHINDP
                C                                                               CHINDP
ISN C123            DO 25 I = 1, COLS                                           CHINDP
ISN C124            IF (X(2) .LE. DATA(I+26)) GO TO 26                          CHINDP
ISN C126         25 CONTINUE                                                    CHINDP
ISN C127            GO TO 21                                                    CHINDP
                C                                                               CHINDP
ISN C128         26 ICOL = I                                                    CHINDP
                C                                                               CHINDP
                C     OBSERVATIONS FALLING EXACTLY ON A LEVEL BOUNDARY ARE PLACED CHINDP
                C     INTO THE LOWER CLASS                                      CHINDP
                C                                                               CHINDP
                C     COUNT THE OBSERVATION AS PER ROW AND COLUMN               CHINDP
                C                                                               CHINDP
ISN C129            NSZ = NSZ + 1                                               CHINDP
ISN C130            IWRK(NRC+IROW) = IWRK(NRC+IROW) + 1                         CHINDP
ISN C131            IWRK(NCC+ICOL) = IWRK(NCC+ICOL) + 1                         CHINDP
ISN C132            IB = (IROW - 1) * COLS + ICOL                               CHINDP
ISN C133            IWRK(IB) = IWRK(IB) + 1                                     CHINDP
ISN C134            GO TO 21                                                    CHINDP
                C                                                               CHINDP
                C     BIVARIATE SAMPLE - FIRST PASS                             CHINDP
                C                                                               CHINDP
ISN C135         31 REWIND UT1                                                  CHINDP
ISN C136            REWIND UT2                                                  CHINDP
                C                                                               CHINDP
ISN C137         32 READ (UT1,END=33) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)       CHINDP
ISN C138            IF (CURVAR .NE. 1) GO TO 32                                 CHINDP
```

```
                C                                                                CHINDP
ISN C140            WRITE (UT2) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)              CHINDP
ISN C141            GO TO 32                                                     CHINDP
                C                                                                CHINDP
                C     END FIRST PASS - START SECOND PASS                         CHINDP
                C                                                                CHINDP
ISN C142         33 ENDFILE UT2                                                  CHINDP
ISN C143            REWIND UT1                                                   CHINDP
ISN C144            REWIND UT2                                                   CHINDP
                C                                                                CHINDP
ISN C145         34 READ (UT1,END=81) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(2)        CHINDP
ISN C146            IF (CURVAR .EQ. 1) GO TO 34                                  CHINDP
                C                                                                CHINDP
ISN C148            READ (UT2,END=1231) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)      CHINDP
ISN C149            ITEMS = 2                                                    CHINDP
ISN C150            LOCAT(4) = 0                                                 CHINDP
                C                                                                CHINDP
                C     PERFORM TRANSFORMATION AND SAVING OF DATA                  CHINDP
                C                                                                CHINDP
ISN C151            IF (WHTRN .EQ. 1) CALL TRANS                                 CHINDP
ISN C153            IF (ITEMS .NE. 2) GO TO 34                                   CHINDP
ISN C155            IF (OUT .EQ. 0 .OR. DOUT .NE. 2) GO TO 36                    CHINDP
ISN C157            CALL WRDATA                                                  CHINDP
                C                                                                CHINDP
                C     CLASSIFY BY ROW                                            CHINDP
                C                                                                CHINDP
ISN C158         36 DO 37 I = 1, ROWS                                            CHINDP
ISN C159            IF (X(1) .LE. DATA(I+1)) GO TO 38                            CHINDP
ISN C161         37 CONTINUE                                                     CHINDP
ISN C162            GO TO 34                                                     CHINDP
                C                                                                CHINDP
ISN C163         38 IROW = I                                                     CHINDP
                C                                                                CHINDP
                C     CLASSIFY BY COLUMN                                         CHINDP
                C                                                                CHINDP
ISN C164            DO 39 I = 1, COLS                                            CHINDP
ISN C165            IF (X(2) .LE. DATA(I+26)) GO TO 40                           CHINDP
ISN C167         39 CONTINUE                                                     CHINDP
ISN C168            GO TO 34                                                     CHINDP
                C                                                                CHINDP
ISN C169         40 ICOL = I                                                     CHINDP
                C                                                                CHINDP
                C     COUNT BY ROW AND COLUMN                                    CHINDP
```

```
                C                                                                   CHINDP
ISN C170            NSZ = NSZ + 1                                                   CHINDP
ISN C171            IWRK(NRC+IROW) = IWRK(NRC+IROW) + 1                             CHINDP
ISN C172            IWRK(NCC+ICOL) = IWRK(NCC+ICOL) + 1                             CHINDP
ISN C173            IB = (IROW - 1) * COLS + ICOL                                   CHINDP
ISN C174            IWRK(IB) = IWRK(IB) + 1                                         CHINDP
ISN C175            GO TO 34                                                        CHINDP
                C                                                                   CHINDP
                C     PRINT THE CONTINGENCY TABLE                                   CHINDP
                C                                                                   CHINDP
ISN C176         81 IF (OUT .NE. 0 .AND. OUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT CHINDP
ISN C178            OUT = 0                                                         CHINDP
ISN C179            WRITE (LST,304) LABEL(2), (I, I = 1, COLS)                      CHINDP
ISN C180            WRITE (LST,305) LABEL(1), (IWRK(NCC+I), I = 1, COLS)            CHINDP
ISN C181            CALL OFLOW(3)                                                   CHINDP
ISN C182            IB = 0                                                          CHINDP
                C                                                                   CHINDP
ISN C183            DO 82 I = 1, ROWS                                               CHINDP
ISN C184            WRITE (LST,306) I, IWRK(NRC+I), (IWRK(IB+J), J = 1, COLS)       CHINDP
ISN C185            CALL OFLOW(1)                                                   CHINDP
ISN C186         82 IB = IB + COLS                                                  CHINDP
                C                                                                   CHINDP
ISN C187            WRITE (LST,307) NSZ                                             CHINDP
ISN C188            CALL OFLOW(2)                                                   CHINDP
                C                                                                   CHINDP
                C     CALCULATE THE TEST STATISTIC                                  CHINDP
                C                                                                   CHINDP
ISN C189            IB = 0                                                          CHINDP
ISN C190            T = 0.                                                          CHINDP
ISN C191            FNSZ = NSZ                                                      CHINDP
                C                                                                   CHINDP
ISN C192            DO 84 I = 1, ROWS                                               CHINDP
ISN C193            RI = IWRK(NRC+I)                                                CHINDP
                C                                                                   CHINDP
ISN C194            DO 83 J = 1, COLS                                               CHINDP
ISN C195            IB = IB + 1                                                     CHINDP
ISN C196            CJ = IWRK(NCC+J)                                                CHINDP
ISN C197            EIJ = RI * CJ / FNSZ                                            CHINDP
ISN C198            OIJ = IWRK(IB)                                                  CHINDP
ISN C199         83 IF (EIJ .NE. 0.) T = T + OIJ * OIJ / EIJ                        CHINDP
                C                                                                   CHINDP
ISN C201         84 CONTINUE                                                        CHINDP
                C                                                                   CHINDP
```

```
                C     DETERMINE THE RESULTS OF THIS TEST                        CHINDP
                C                                                               CHINDP
   ISN C202           T = T - FNSZ                                              CHINDP
                C                                                               CHINDP
   ISN C203           DF = (ROWS - 1) * (COLS - 1)                              CHINDP
   ISN C204           IDF = DF                                                  CHINDP
   ISN C205           WRITE (LST,309) T, IDF                                    CHINDP
   ISN C206           CALL OFLOW(3)                                             CHINDP
                C                                                               CHINDP
   ISN C207           ALFHAT = CDTR(T,DF)                                       CHINDP
   ISN C208           I = 1                                                     CHINDP
   ISN C209           IF (ALPHA .GT. ALFHAT) I = 2                              CHINDP
   ISN C211           WRITE (LST,310) ALPHA,ACCREJ(I)                           CHINDP
   ISN C212           WRITE (LST,311) ALFHAT                                    CHINDP
   ISN C213           CALL OFLOW(5)                                             CHINDP
                C                                                               CHINDP
                C     END OF THE TEST                                           CHINDP
                C                                                               CHINDP
   ISN C214        91 WRITE (LST,312)                                           CHINDP
   ISN C215           CALL OFLOW(3)                                             CHINDP
   ISN C216           RETURN                                                    CHINDP
                C                                                               CHINDP
                C     ERRORS THAT MIGHT OCCUR                                   CHINDP
                C                                                               CHINDP
   ISN C217      1231 WRITE (LST,231)                                           CHINDP
   ISN C218           GO TO 99                                                  CHINDP
                C                                                               CHINDP
   ISN C219      1232 WRITE (LST,232) LABEL(IWCH)                               CHINDP
   ISN C220           GO TO 99                                                  CHINDP
                C                                                               CHINDP
   ISN C221      1233 WRITE (LST,233) LABEL(IWCH)                               CHINDP
                C                                                               CHINDP
                C     SET RETURN CODE AND END THE TEST                          CHINDP
                C                                                               CHINDP
   ISN C222        99 CALL CFLOW(3)                                            CHINDP
   ISN C223           RC = 12                                                   CHINDP
   ISN C224           GO TO 91                                                  CHINDP
                C                                                               CHINDP
   ISN C225           END                                                       CHINDP

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
```

```
*STATISTICS*      SOURCE STATEMENTS =      224 ,PROGRAM SIZE =      4546
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            SUBROUTINE CHIRXC                                                    CHIRXC
                    C                                                                    CHIRXC
                    C     THE CHI-SQUARE TEST FOR PROBABILITY DIFFERENCES - R BY C       CHIRXC
                    C                                                                    CHIRXC
                    C     THE CHI-SQUARE TEST FOR DIFFERENCES CAN BE USED TO TEST        CHIRXC
                    C     "R" RANDOM SAMPLES (ROWS), NOT NECESSARILY THE SAME SIZES,     CHIRXC
                    C     DIVIDED INTO "C" CLASSES (COLUMNS), HAVING EQUAL PROBABILITY   CHIRXC
                    C     OF BEING IN EACH CLASS.                                        CHIRXC
                    C                                                                    CHIRXC
                    C     THE CLASS INTERVALS ARE SPECIFIED ON THE CLASSES,COLUMNS,C,...  CHIRXC
                    C     CARD AND MUST SPECIFY AT LEAST 2 COLUMNS.                      CHIRXC
                    C                                                                    CHIRXC
                    C     AS THE NAME IMPLIES, THE CHI-SQUARE DISTRIBUTION IS USED       CHIRXC
                    C     TO DETERMINE THE CRITICAL REGION.                              CHIRXC
                    C                                                                    CHIRXC
                    C     FOR DETAILS, SEE CONOVER, W.J., PRACTICAL NON-PARAMETRIC       CHIRXC
                    C     STATISTICS, ROUGH DRAFT, PAGES 240-247.                        CHIRXC
                    C                                                                    CHIRXC
                    C                                                                    CHIRXC
                    C     COMMON BLOCKS                                                  CHIRXC
                    C                                                                    CHIRXC
ISN C003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                         DECLARE
ISN C004            INTEGER*4  ROWS,COLS                                                 DECLARE
ISN C005            COMMON /DIST/  DISTN,CPARM1,CPARM2,WHTRN                             DECLARE
ISN C006            INTEGER*4  DISTN,WHTRN                                               DECLARE
ISN C007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                          DECLARE
ISN C008            REAL*8  BUFFER                                                       DECLARE
ISN C009            INTEGER*4  EOD,CDCOL,BUFNO                                           DECLARE
ISN C010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD                DECLARE
ISN C011            REAL*8   FMT                                                         DECLARE
ISN C012            INTEGER*4  DFORM,DOUT,RCBCD                                          DECLARE
ISN C013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),          DECLARE
                   1    VALUE(40),IVALUE(40),RC                                         DECLARE
ISN C014            REAL*8   PARM,SUBPRM                                                 DECLARE
ISN C015            INTEGER*4  RC                                                        DECLARE
ISN C016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                          DECLARE
ISN C017            INTEGER*4  TAIL                                                      DECLARE
ISN C018            LOGICAL*4  TESTS,DOALL                                               DECLARE
ISN C019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),              DECLARE
                   1    DATE,TIME,PAGE,LINE                                             DECLARE
ISN C020            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                     DECLARE
```

```
ISN C021            INTEGER*4  PAGE                                                    DECLARE
ISN C022            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
ISN C023            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE     DECLARE
ISN C024            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                          DECLARE
ISN C027            INTEGER*4  ITEMS,LOCAT,RDNO                                       DECLARE
ISN C028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                      DECLARE
ISN C029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                            DECLARE
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                                     DECLARE
ISN C031            REAL*8  BLANK /' '/                                               DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                        DECLARE
                C                                                                     CHIRXC
                C     OTHER DATA USED IN THIS PROGRAM                                 CHIRXC
                C                                                                     CHIRXC
ISN C033            REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                        CHIRXC
ISN C034            INTEGER*4  CURSMP,CURVAR,CURCNT                                   CHIRXC
ISN C035            REAL*4  LOVAL /-1.E75/, HIVAL /+1.E75/                            CHIRXC
                C                                                                     CHIRXC
                C     FORMAT STATEMENTS                                               CHIRXC
                C                                                                     CHIRXC
ISN C036          232 FORMAT ('-', T12, 'IKMS321  TOO FEW COLUMN CLASSES SPECIFIED')  CHIRXC
ISN C037          233 FORMAT ('-', T12, 'IKMS331  TOO FEW COLUMN   DICHOTOMIZATION LEVELCHIRXC
                     1S SPECIFIED')                                                   CHIRXC
ISN C038          234 FORMAT ('-', T12, 'IKMS341  INVALID SAMPLE SIZE(S)')            CHIRXC
ISN C039          301 FORMAT ('-', T36, 'THE CHI-SQUARE TEST FOR DIFFERENCES IN PROBABILCHIRXC
                     1ITIES -', I3, ' BY', I3)                                        CHIRXC
ISN 0040          302 FORMAT ('0', T12, 'HO: THE PROBABILITY OF APPEARRING IN ANY COLUMNCHIRXC
                     1 IS THE SAME FOR ALL SAMPLES'/ '0', T12, 'VERSUS'/ '0', T12,    CHIRXC
                     2    'H1: THE PROBABILITY OF APPEARRING IN ANY COLUMN IS DIFFERENT CHIRXC
                     3FOR AT LEAST 2 SAMPLES')                                        CHIRXC
ISN C041          304 FORMAT ('-', T12, 'THE SAMPLES ARE TO BE DICHOTOMIZED INTO', I3, CHIRXC
                     1    ' CLASSES AS FOLLOWS:' //                                  CHIRXC
                     2    (' ', T12, 4(G10.4, ' TO ', G10.4, 5X) ) )                 CHIRXC
ISN C042          305 FORMAT ('0', T12, 'THE SAMPLES WERE DISTRIBUTED AS FOLLOWS:' /  CHIRXC
                     1    '0', ' SAMPLE NUMBER  SIZE   CLASSES', 20I5)               CHIRXC
ISN C043          306 FORMAT (' ', A8, I4, I7, 11X, 20I5)                           CHIRXC
ISN C044          307 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5, ' WITH',    CHIRXC
                     1    I4, ' DEGREES OF FREEDOM')                                 CHIRXC
ISN C045          308 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')    CHIRXC
ISN C046          309 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                        CHIRXC
ISN C047          310 FORMAT ('-', T41, 'END OF THE CHI-SQUARE TEST FOR PROBABILITY DIFFCHIRXC
                     1ERENCES')                                                       CHIRXC
```

```
                C                                                                   CHIRXC
                C     START BY PRINTING THE TITLE AND CHECKING THE COLUMNS          CHIRXC
                C                                                                   CHIRXC
ISN 0048              WKSZE = WKSZE                                                 CHIRXC
ISN 0049              WRITE (LST,301) SMPLS, COLS                                   CHIRXC
ISN 0050              CALL OFLOW(3)                                                 CHIRXC
                C                                                                   CHIRXC
ISN 0051              IF (COLS .LT. 2) GO TO 1232                                   CHIRXC
                C                                                                   CHIRXC
                C     PRINT THE HYPOTHESES                                          CHIRXC
                C                                                                   CHIRXC
ISN 0053           11 WRITE (LST,302)                                               CHIRXC
ISN 0054              CALL OFLOW(6)                                                 CHIRXC
                C                                                                   CHIRXC
                C     BUILD THE COLUMN LIMITS AND CHECK FOR NOT ENOUGH OF THEM      CHIRXC
                C                                                                   CHIRXC
ISN 0055              DATA(1) = LOVAL                                               CHIRXC
ISN 0056              DO 12 I = 1, COLS                                             CHIRXC
ISN 0057           12 DATA(I+1) = CLEV(I)                                           CHIRXC
ISN 0058              DATA(COLS+1) = HIVAL                                          CHIRXC
ISN 0059              DO 13 I = 1, COLS                                             CHIRXC
ISN 0060              IF (DATA(I) .GE. DATA(I+1)) GO TO 1233                        CHIRXC
ISN 0062           13 CONTINUE                                                      CHIRXC
                C                                                                   CHIRXC
                C     PRINT OUT THE LEVEL LIMITS                                    CHIRXC
                C                                                                   CHIRXC
ISN 0063              WRITE (LST,304) COLS, (DATA(I), DATA(I+1), I = 1, COLS)       CHIRXC
ISN 0064              I = 3 + (COLS + 3) / 4                                        CHIRXC
ISN 0065              CALL OFLOW(I)                                                 CHIRXC
                C                                                                   CHIRXC
                C     ZERO OUT SAMPLE COUNTERS                                      CHIRXC
                C                                                                   CHIRXC
ISN 0066              ISZ = SMPLS * COLS                                            CHIRXC
ISN 0067              II = ISZ + SMPLS                                              CHIRXC
ISN 0068              DO 15 I = 1, II                                               CHIRXC
ISN 0069           15 IWRK(I) = 0                                                   CHIRXC
ISN 0070              NSZ = 0                                                       CHIRXC
ISN 0071              IBASE = 0                                                     CHIRXC
                C                                                                   CHIRXC
                C     PASS THE DATA ONCE FOR EACH SAMPLE - SLOW, BUT NECESSARY      CHIRXC
                C                                                                   CHIRXC
ISN 0072              DO 40 ISMP = 1, SMPLS                                         CHIRXC
ISN 0073              ISZN = ISZ + ISMP                                             CHIRXC
```

```
ISN C074            REWIND UT1                                                  CHIRXC
                C                                                               CHIRXC
                C   READ ONE OBSERVATION                                        CHIRXC
                C                                                               CHIRXC
ISN C075         21 READ (UT1,END=31) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)       CHIRXC
ISN C076            IF (CURSMP .NE. ISMP) GO TO 21                              CHIRXC
                C                                                               CHIRXC
                C   PERFORM POSSIBLE TRANSFORMATION AND SAVING OF DATA          CHIRXC
                C                                                               CHIRXC
ISN C078            ITEMS = 1                                                   CHIRXC
ISN C079            LOCAT(4) = 0                                                CHIRXC
ISN C080            IF (WHTRN .EQ. 1) CALL TRANS                                CHIRXC
ISN C082            IF (ITEMS .LT. 1) GO TO 21                                  CHIRXC
ISN C084            IF (OUT .EQ. 0 .OR. DOUT .NE. 2) GO TO 22                   CHIRXC
ISN C086            CALL WRDATA                                                 CHIRXC
                C                                                               CHIRXC
                C   FIND WHERE EACH ITEM BELONGS (COLUMNS)                      CHIRXC
                C                                                               CHIRXC
ISN C087         22 DO 26 I = 1, ITEMS                                          CHIRXC
ISN C088            DO 24 J = 1, COLS                                           CHIRXC
ISN C089            IF (X(I) .LE. DATA(J+1)) GO TO 25                           CHIRXC
                C   NOTE THE VALUES EQUAL TO THE BORDERLINE ARE PLACED          CHIRXC
                C   IN THE LOWER COLUMN                                        CHIRXC
ISN C091         24 CONTINUE                                                    CHIRXC
ISN C092            GO TO 26                                                    CHIRXC
                C                                                               CHIRXC
                C   COUNT IN TOTOAL, IN SAMPLE, AND IN COLUMN                  CHIRXC
                C                                                               CHIRXC
ISN C093         25 IWRK(ISZN) = IWRK(ISZN) + 1                                 CHIRXC
ISN C094            IB = IBASE + J                                              CHIRXC
ISN C095            IWRK(IB) = IWRK(IB) + 1                                     CHIRXC
ISN C096            NSZ = NSZ + 1                                               CHIRXC
                C                                                               CHIRXC
ISN C097         26 CONTINUE                                                    CHIRXC
ISN C098            GO TO 21                                                    CHIRXC
                C                                                               CHIRXC
                C   END OF THE READING OF 1 SAMPLE                             CHIRXC
                C                                                               CHIRXC
ISN 0099         31 IBASE = IBASE + COLS                                        CHIRXC
                C                                                               CHIRXC
ISN 0100         40 CONTINUE                                                    CHIRXC
                C                                                               CHIRXC
                C   END OF ALL THE DATA PASSES                                  CHIRXC
```

```
                C                                                                   CHIRXC
ISN C101              IF (OUT .NE. 0 .AND. COUT .EC. 2 .AND. OUT .NE. LST) ENDFILE OUT  CHIRXC
ISN C103              OUT = 0                                                         CHIRXC
                C                                                                   CHIRXC
                C     PRINT THE SAMPLE BREAKDOWNS                                   CHIRXC
                C                                                                   CHIRXC
ISN C104              WRITE (LST,305) (I, I = 1, CCLS)                               CHIRXC
ISN C105              CALL OFLOW(4)                                                  CHIRXC
ISN C106              IB = 0                                                         CHIRXC
ISN C107              DO 41 I = 1, SMPLS                                             CHIRXC
ISN C108              WRITE (LST,306) LABEL(I), I, IWRK(ISZ+I),                      CHIRXC
                     1    (IWRK(IB+J), J = 1, COLS)                                  CHIRXC
ISN C109              CALL OFLOW(1)                                                  CHIRXC
ISN 0110              SSIZE(I) = IWRK(ISZ+I)                                         CHIRXC
ISN C111              IF (SSIZE(I) .LT. COLS) GO TC 1234                             CHIRXC
                C     BECAUSE OF THE LARGE SAMPLE ASSUMPTICN, EACH SAMPLE MUST        CHIRXC
                C     HAVE ANY MANY OBSERVATIONS AS THERE ARE COLUMNS                 CHIRXC
ISN C113           41 IB = IB + COLS                                                 CHIRXC
                C                                                                   CHIRXC
                C     SUM UP COLUMN TOTALS                                           CHIRXC
                C                                                                   CHIRXC
ISN C114              DO 43 I = 1, CCLS                                              CHIRXC
ISN C115              K = ISZ + I                                                    CHIRXC
ISN 0116              IWRK(K) = 0                                                    CHIRXC
                C                                                                   CHIRXC
ISN C117              DO 42 J = 1, SMPLS                                             CHIRXC
ISN C118              IB = (J - 1) * COLS + I                                        CHIRXC
ISN C119           42 IWRK(K) = IWRK(K) + IWRK(IB)                                   CHIRXC
                C                                                                   CHIRXC
ISN C120           43 CONTINUE                                                       CHIRXC
                C                                                                   CHIRXC
                C     CALCULATE THE TEST STATISTIC                                   CHIRXC
                C                                                                   CHIRXC
ISN C121              T = 0.                                                         CHIRXC
ISN C122              FNSZ = NSZ                                                     CHIRXC
ISN C123              IB = 0                                                         CHIRXC
                C                                                                   CHIRXC
ISN C124              DO 52 I = 1, SMPLS                                             CHIRXC
ISN C125              FNI = SSIZE(I)                                                 CHIRXC
                C                                                                   CHIRXC
ISN C126              DO 51 J = 1, COLS                                              CHIRXC
ISN C127              CJ = IWRK(ISZ+J)                                               CHIRXC
ISN C128              EIJ = FNI * CJ / FNSZ                                          CHIRXC
```

```
ISN C129            IB = IB + 1                                                 CHIRXC
ISN C130            OIJ = IWRK(IB)                                              CHIRXC
ISN C131         51 IF (EIJ .NE. 0.) T = T + OIJ * OIJ / EIJ                    CHIRXC
                 C                                                              CHIRXC
ISN C133         52 CONTINUE                                                    CHIRXC
                 C                                                              CHIRXC
                 C     DETERMINE THE CRITICAL REGION FROM THE CHI-SQUARE DISTRIBUTION CHIRXC
                 C                                                              CHIRXC
ISN C134            T = T - FNSZ                                                CHIRXC
ISN C135            DF = (SMPLS - 1) * (CCLS - 1)                               CHIRXC
ISN C136            IDF = DF                                                    CHIRXC
ISN C137            WRITE (LST,307) T, IDF                                      CHIRXC
ISN C138            CALL OFLOW(3)                                               CHIRXC
                 C                                                              CHIRXC
ISN C139            ALFHAT = CDTR(T,DF)                                         CHIRXC
ISN C140            I = 1                                                       CHIRXC
ISN C141            IF (ALPHA .GT. ALFHAT) I = 2                                CHIRXC
                 C                                                              CHIRXC
                 C     PRINT THE RESULT REACHED                                 CHIRXC
                 C                                                              CHIRXC
ISN 0143            WRITE (LST,308) ALPHA,ACCREJ(I)                             CHIRXC
ISN C144            WRITE (LST,309) ALFHAT                                      CHIRXC
ISN C145            CALL OFLOW(5)                                               CHIRXC
                 C                                                              CHIRXC
                 C     END OF THE TEST - RETURN TO CALLER                       CHIRXC
                 C                                                              CHIRXC
ISN C146         91 WRITE (LST,310)                                             CHIRXC
ISN 0147            CALL OFLOW(3)                                               CHIRXC
ISN C148            RETURN                                                      CHIRXC
                 C                                                              CHIRXC
                 C     DATA OR SPECIFICATION ERRORS                             CHIRXC
                 C                                                              CHIRXC
ISN C149       1232 WRITE (LST,232)                                             CHIRXC
ISN C150            GO TO 99                                                    CHIRXC
                 C                                                              CHIRXC
ISN C151       1233 WRITE (LST,233)                                             CHIRXC
ISN C152            GO TO 99                                                    CHIRXC
                 C                                                              CHIRXC
ISN C153       1234 WRITE (LST,234)                                             CHIRXC
                 C                                                              CHIRXC
ISN C154         99 CALL OFLOW(3)                                               CHIRXC
ISN C155            RC = 12                                                     CHIRXC
ISN C156            GO TO 91                                                    CHIRXC
```

```
                                                                                                  PAGE 007
                C                                                                                 CHIRXC
   ISN C157            END                                                                        CHIRXC

*OPTICNS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTICNS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     156 ,PRCGRAM SIZE =     3370

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          FUNCTION DLGAM (XX)                                                DLGAM
 ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)                                          DLGAM
            C                                                                         DLGAM
            C     THIS FUNCTION SUBPROGRAM COMPUTES THE DOUBLE PRECISION              DLGAM
            C     NATURAL LOGARITHM OF THE GAMMA FUNCTION OF XX, WHICH MUST ALSO      DLGAM
            C     BE DOUBLE PRECISION.                                                DLGAM
            C                                                                         DLGAM
            C     THIS FUNCTION IS AN ADAPTATION OF THE SUBROUTINE DLGAM              DLGAM
            C     FROM IBM'S SSP, VERSION III.                                        DLGAM
            C                                                                         DLGAM
            C           THE EULER-MCLAURIN EXPANSION TO THE SEVENTH DERIVATIVE TERM   DLGAM
            C           IS USED, AS GIVEN BY M. ABRAMOWITZ AND I.A. STEGUN,           DLGAM
            C           'HANDBOOK OF MATHEMATICAL FUNCTIONS', U. S. DEPARTMENT OF     DLGAM
            C           COMMERCE, NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS    DLGAM
            C           SERIES. 1966, EQUATION 6.1.41.                                DLGAM
            C                                                                         DLGAM
            C                                                                         DLGAM
 ISN 0004          ZZ=XX                                                              DLGAM
 ISN 0005          IF (XX .GT. 1.D10) GO TO 8                                         DLGAM
            C                                                                         DLGAM
            C     XX LESS THAN OR EQUAL TO 10**10                                     DLGAM
            C                                                                         DLGAM
 ISN 0007          TERM=1.D0                                                          DLGAM
 ISN 0008        5 IF (ZZ .GT. 18.D0) GO TO 7                                         DLGAM
 ISN 0010          TERM = TERM * ZZ                                                   DLGAM
 ISN 0011          ZZ = ZZ + 1.D0                                                     DLGAM
 ISN 0012          GO TO 5                                                            DLGAM
 ISN 0013        7 RZ2=1.D0/ZZ**2                                                     DLGAM
 ISN 0014          DLNG =(ZZ-0.5D0)*DLOG(ZZ)-ZZ +0.9189385332046727 -DLOG(TERM)+       DLGAM
                  1(1.D0/ZZ)*(.8333333333333333D-1 -(RZ2*(.2777777777777777D-2 +(RZ2*DLGAM
                  2(.7936507936507936D-3 -(RZ2*(.5952380952380952D-3)))))))           DLGAM
 ISN 0015          GO TO 10                                                           DLGAM
            C                                                                         DLGAM
            C     XX IS GREATER THAN 10**10                                           DLGAM
            C                                                                         DLGAM
 ISN 0016        8 DLNG=ZZ*(DLOG(ZZ)-1.D0)                                            DLGAM
            C                                                                         DLGAM
 ISN 0017       10 DLGAM = DLNG                                                       DLGAM
 ISN 0018          RETURN                                                             DLGAM
            C                                                                         DLGAM
 ISN 0019          END                                                                DLGAM
```

```
*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45
*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
*STATISTICS*     SOURCE STATEMENTS =      18 ,PROGRAM SIZE =      590
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE DOTEST                                                  DOTEST
              C                                                                        DOTEST
              C    THIS SUBROUTINE CHECKS THE ARRAY OF TESTS SPECIFICATIONS AND        DOTEST
              C    CALLS THE APPROPRIATE SUBROUTINES TO PERFORM THE ACTUAL             DOTEST
              C    TESTING.  IF AT ANY TIME, A TEST RETURNS AN ERROR RETURN CODE,      DOTEST
              C    FURTHER TESTING IS SUSPENDED, AND A RETURN TO THE CALLER OF         DOTEST
              C    THIS PROGRAM IS EXECUTED.                                           DOTEST
              C                                                                        DOTEST
              C                                                                        DOTEST
              C    COMMON BLOCKS                                                       DOTEST
              C                                                                        DOTEST
 ISN 0003          COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                        DECLARE
 ISN 0004          INTEGER*4  ROWS,COLS                                                DECLARE
 ISN 0005          COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                            DECLARE
 ISN 0006          INTEGER*4  DISTN,WHTRN                                              DECLARE
 ISN 0007          COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                         DECLARE
 ISN 0008          REAL*8  BUFFER                                                      DECLARE
 ISN 0009          INTEGER*4  EOD,CDCOL,BUFNO                                          DECLARE
 ISN 0010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD               DECLARE
 ISN 0011          REAL*8   FMT                                                        DECLARE
 ISN 0012          INTEGER*4  DFORM,DOUT,RCBCD                                         DECLARE
 ISN 0013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),         DECLARE
                  1    VALUE(40),IVALUE(40),RC                                         DECLARE
 ISN 0014          REAL*8   PARM,SUBPRM                                                DECLARE
 ISN 0015          INTEGER*4  RC                                                       DECLARE
 ISN 0016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                         DECLARE
 ISN 0017          INTEGER*4  TAIL                                                     DECLARE
 ISN 0018          LOGICAL*4  TESTS,DOALL                                              DECLARE
 ISN 0019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),             DECLARE
                  1    DATE,TIME,PAGE,LINE                                             DECLARE
 ISN 0020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                    DECLARE
 ISN 0021          INTEGER*4  PAGE                                                     DECLARE
 ISN 0022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)   DECLARE
 ISN 0023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE       DECLARE
 ISN 0024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                 DECLARE
 ISN 0025          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                 DECLARE
 ISN 0026          COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                            DECLARE
 ISN 0027          INTEGER*4  ITEMS,LOCAT,RDNO                                         DECLARE
 ISN 0028          COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                        DECLARE
 ISN 0029          INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                              DECLARE
 ISN 0030          EQUIVALENCE (WORK(1),IWRK(1))                                       DECLARE
```

```
ISN C031            REAL*8  BLANK /' '/                                        DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                DOTEST
                C                                                             DOTEST
                C     FORMAT STATEMENTS                                       DOTEST
                C                                                             DOTEST
ISN C033          149 FORMAT ('-', T47, 'REQUESTED TESTING OF THIS DATA IS COMPLETE')  DOTEST
ISN C034          222 FORMAT ('-', T12, 'IKMS221  ERROR IN PERFORMING THIS TEST CAUSES ADOTEST
                   1NY REMAINING TESTS TO BE SKIPPED')                        DOTEST
                C                                                             DOTEST
                C     INITIALIZE THE RETURN CODE AND TEST POINTER             DOTEST
                C                                                             DOTEST
ISN C035            WKSZE = WKSZE                                             DOTEST
ISN C036            RC = 0                                                    DOTEST
ISN C037            I = 0                                                     DOTEST
                C                                                             DOTEST
                C     INCREMENT TEST POINTER AND LOOK FOR MORE TESTS TO PERFORM  DOTEST
                C                                                             DOTEST
ISN C038          1 ITST = I + 1                                              DOTEST
                C                                                             DOTEST
ISN C039            DO 2 I = ITST, 15                                         DOTEST
ISN C040            IF (TESTS(I)) GO TO 3                                     DOTEST
ISN C042          2 CONTINUE                                                  DOTEST
                C                                                             DOTEST
                C     NO MORE TESTS FOUND - GO TO RETURN FROM THIS ROUTINE    DOTEST
                C                                                             DOTEST
ISN C043            GO TO 4                                                   DOTEST
                C                                                             DOTEST
                C     PRINT THE PAGE HEADING AND COMPUTE WHICH SUBROUTINE TO CALL  DOTEST
                C                                                             DOTEST
ISN C044          3 IF (LINE .GT. 6) CALL HEAD                                DOTEST
ISN C046            GO TO (1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008,    DOTEST
                  1     1009, 1010, 1011, 1012, 1013, 1014, 1015), I          DOTEST
                C                                                             DOTEST
                C     CALL THE APPROPRIATE SUBROUTINE TO PERFORM THE TEST     DOTEST
                C                                                             DOTEST
ISN C047         1001 CALL SGNTST                                             DOTEST
ISN C048            GO TO 2000                                                DOTEST
                C                                                             DOTEST
ISN C049         1002 CALL MCNMAR                                             DOTEST
ISN C050            GO TO 2000                                                DOTEST
                C                                                             DOTEST
ISN C051         1003 CALL CHIRXC                                             DOTEST
ISN C052            GO TO 2000                                                DOTEST
```

```
                C                                                               DOTEST
ISN C053         1004 CALL CHINDP                                               DOTEST
ISN C054              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C055         1005 CALL MEDIAN                                               DOTEST
ISN C056              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C057         1006 CALL WILCOX                                               DOTEST
ISN C058              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C059         1007 CALL MANWHT                                               DOTEST
ISN C060              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C061         1008 CALL MOOD                                                 DOTEST
ISN C062              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C063         1009 CALL RHO                                                  DOTEST
ISN C064              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C065         1010 CALL TREND                                                DOTEST
ISN C066              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C067         1011 CALL TAU                                                  DOTEST
ISN C068              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C069         1012 CALL KRSKAL                                               DOTEST
ISN C070              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C071         1013 CALL FRDMAN                                               DOTEST
ISN C072              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C073         1014 CALL GDFIT                                                DOTEST
ISN C074              GO TO 2000                                                DOTEST
                C                                                               DOTEST
ISN C075         1015 CALL DURBIN                                               DOTEST
ISN C076              GO TO 2000                                                DOTEST
                C                                                               DOTEST
                C     TEST HAS BEEN PERFORMED - CHECK RETURN CODE               DOTEST
                C                                                               DOTEST
ISN C077         2000 IF (RC .EQ. 0) GO TO 2001                                 DOTEST
                C                                                               DOTEST
                C     WRITE ERROR MESSAGE BECAUSE OF TEST ERROR                 DOTEST
                C                                                               DOTEST
```

```
ISN C079            WRITE (LST,222)                                          DOTEST
ISN C080            GO TO 5                                                  DOTEST
                C                                                            DOTEST
                C     NORMAL RETURN FROM TEST - CHECK FOR LAST TEST          DOTEST
                C                                                            DOTEST
ISN C081         2001 IF (ITST .LT. 15) GO TO 1                              DOTEST
                C                                                            DOTEST
                C     WRITE OUT NORMAL END OF TESTS MESSAGE                  DOTEST
                C                                                            DOTEST
ISN C083            4 WRITE (LST,149)                                        DOTEST
                C                                                            DOTEST
                C     RETURN TO CALLING PROGRAM                              DOTEST
                C                                                            DOTEST
ISN C084            5 RETURN                                                 DOTEST
                C                                                            DOTEST
ISN C085            END                                                      DOTEST

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      84 ,PROGRAM SIZE =      940

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE DURBIN                                                DURBIN
              C                                                                      DURBIN
              C    THE DURBIN TEST                                                   DURBIN
              C                                                                      DURBIN
              C    THE DURBIN TEST IS A TEST FOR TREATMENT DIFFERENCES IN A          DURBIN
              C    BALANCED INCOMPLETE BLOCK DESIGN (BIBD).  IT IS ACTUALLY A        DURBIN
              C    GENERALIZATION OF THE FRIEDMAN TEST TO BIBD DATA.  RANKS          DURBIN
              C    ARE USED TO COMPUTE THE TEST STATISTIC WHICH HAS A CHI-SQUARE     DURBIN
              C    DISTRIBUTION UNDER THE NULL HYPOTHESIS OF NO DIFFERENCES          DURBIN
              C    IN TREATMENTS.                                                    DURBIN
              C                                                                      DURBIN
              C    FOR DETAILS ON THE DURBIN TEST, SEE CONOVER, W.J., PRACTICAL      DURBIN
              C    NONPARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 438-445.             DURBIN
              C                                                                      DURBIN
              C                                                                      DURBIN
              C    COMMON BLOCKS                                                     DURBIN
              C                                                                      DURBIN
 ISN 0003          COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                      DECLARE
 ISN 0004          INTEGER*4  ROWS,COLS                                              DECLARE
 ISN 0005          COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                          DECLARE
 ISN 0006          INTEGER*4  DISTN,WHTRN                                            DECLARE
 ISN 0007          COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                       DECLARE
 ISN 0008          REAL*8  BUFFER                                                    DECLARE
 ISN 0009          INTEGER*4  EOD,CDCOL,BUFNO                                        DECLARE
 ISN 0010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD             DECLARE
 ISN 0011          REAL*8   FMT                                                      DECLARE
 ISN 0012          INTEGER*4  DFORM,DOUT,RCBCD                                       DECLARE
 ISN 0013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),       DECLARE
                  1    VALUE(40),IVALUE(40),RC                                       DECLARE
 ISN 0014          REAL*8   PARM,SUBPRM                                              DECLARE
 ISN 0015          INTEGER*4  RC                                                     DECLARE
 ISN 0016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                       DECLARE
 ISN 0017          INTEGER*4  TAIL                                                   DECLARE
 ISN 0018          LOGICAL*4  TESTS,DOALL                                            DECLARE
 ISN 0019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),           DECLARE
                  1    DATE,TIME,PAGE,LINE                                           DECLARE
 ISN 0020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                  DECLARE
 ISN 0021          INTEGER*4  PAGE                                                   DECLARE
 ISN 0022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
 ISN 0023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE     DECLARE
 ISN 0024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
```

```
ISN C025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                           DECLARE
ISN C027            INTEGER*4  ITEMS,LOCAT,RDNO                                        DECLARE
ISN C028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                       DECLARE
ISN C029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                             DECLARE
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                                      DECLARE
ISN C031            REAL*8  BLANK /' '/                                                DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                         DECLARE
                C                                                                      DURBIN
                C     OTHER DATA USED                                                  DURBIN
                C                                                                      DURBIN
ISN C033            REAL*8 ACCREJ(2) /'ACCEPTED', 'REJECTED'/                          DURBIN
ISN C034            REAL*4  RECORD(6)                                                  DURBIN
ISN C035            INTEGER*4  CURSMP,CURVAR,CURCNT,CURBLK,CURTRT,CURREP               DURBIN
ISN C036            EQUIVALENCE (RECORD(1),CURSMP,CURBLK), (RECORD(2),CURVAR,CURTRT),  DURBIN
                   1    (RECORD(3),CURCNT,CURREP), (RECORD(4),XRANK), (RECORD(6),OBS)  DURBIN
ISN C037            REAL*8  BLKLBL /'BLOCKS'/, TRTLBL /'TRTMNTS'/                      DURBIN
                C                                                                      DURBIN
                C     FORMAT STATEMENTS                                                DURBIN
                C                                                                      DURBIN
ISN C038          234 FORMAT ('-', T12, 'IKMS34I  INVALID SAMPLE SIZE(S)')             DURBIN
ISN C039          236 FORMAT ('-', T12, 'IKMS36I  TOO MANY TIED VALUES FOR THIS PROGRAM DURBIN
                     1TO HANDLE')                                                      DURBIN
ISN C040          301 FORMAT ('-', T61, 'THE DURBIN TEST')                             DURBIN
ISN C041          302 FORMAT ('0', T12, 'H0: THE', I3, ' TREATMENTS, CALLED ', A8,     DURBIN
                     1    ', ARE EQUALLY EFFECTIVE ACROSS BLOCKS, CALLED ', A8/       DURBIN
                     2    '0', T12, 'VERSUS'/ '0', T12, 'H1: AT LEAST 1 TREATMENT YIELDSDURBIN
                     3 LARGER OBSERVATIONS THAN THE OTHER TREATMENTS')                 DURBIN
ISN C042          303 FORMAT ('-', T12, 'THE OBSERVATIONS ARE TAKEN ACROSS', I3,       DURBIN
                     1    ' BLOCKS'/ ' ', T12, 'WITH', I3, ' TREATMENTS IN A BLOCK'/  DURBIN
                     2    ' ', T12, 'AND ', I3, ' BLOCKS IN A TREATMENT')             DURBIN
ISN C043          304 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5, ' WITH',      DURBIN
                     1    I3, ' DEGREES OF FREEDOM')                                  DURBIN
ISN C044          305 FORMAT ('-', T12, 'THERE WAS A TOTAL OF', I5, ' TIES IN',        DURBIN
                     1    I3, ' PLACES')                                              DURBIN
ISN C045          306 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', H0 IS ', A8, '.')       DURBIN
ISN C046          307 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                           DURBIN
ISN C047          308 FORMAT ('-', T57, 'END OF THE DURBIN TEST')                      DURBIN
                C                                                                      DURBIN
                C     START WITH THE TITLE AND HYPOTHESES                              DURBIN
                C                                                                      DURBIN
ISN C048            WRITE (LST,301)                                                    DURBIN
ISN C049            IF (BLOCK .EQ. BLANK) BLOCK = BLKLBL                               DURBIN
```

```
ISN C051            IF (TREAT .EQ. BLANK) TREAT = TRTLBL                    DURBIN
ISN C053            WRITE (LST,302) TRT,TREAT,BLOCK                         DURBIN
ISN C054            WRITE (LST,303) BLK,TRTB,BLKT                           DURBIN
ISN C055            CALL OFLOW(14)                                          DURBIN
                C                                                           DURBIN
                C     SORT THE DATA DOWN FOR RANKING                        DURBIN
                C                                                           DURBIN
ISN C056            SUBS(1) = 0                                             DURBIN
ISN C057            SUBS(2) = 0                                             DURBIN
ISN C058            CALL RANKER (UT4,UT1,UT2,UT3,J)                         DURBIN
ISN C059            IF (RC .NE. 0) GO TO 92                                 DURBIN
                C                                                           DURBIN
                C     RANK THE DATA WITHIN BLOCKS                           DURBIN
                C                                                           DURBIN
ISN C061            TIES = 0                                                DURBIN
ISN C062            PLCS = 0                                                DURBIN
ISN C063            REWIND UT2                                              DURBIN
                C                                                           DURBIN
ISN C064            DO 50 IBLK = 1, BLK                                     DURBIN
ISN C065            REWIND UT4                                              DURBIN
ISN C066            RANK = 0.                                               DURBIN
ISN C067         22 READ (UT4,END=1234) RECORD                              DURBIN
ISN C068            IF (CURBLK .NE. IBLK) GO TO 22                          DURBIN
                C                                                           DURBIN
                C     PROCESS THE OBSERVATION AND READ THE NEXT ONE         DURBIN
                C                                                           DURBIN
ISN C070         23 DO 24 I = 1, 6                                          DURBIN
ISN C071         24 DATA(I) = RECORD(I)                                     DURBIN
ISN C072            RANK = RANK + 1.                                        DURBIN
                C                                                           DURBIN
ISN C073         25 READ (UT4,END=26) RECORD                                DURBIN
ISN C074            IF (CURBLK .NE. IBLK) GO TO 25                          DURBIN
ISN C076            IF (DATA(6) .EQ. OBS) GO TO 31                          DURBIN
                C                                                           DURBIN
                C     RANK THE OBSERVATION SINCE IT IS NOT TIED             DURBIN
                C                                                           DURBIN
ISN C078            DATA(4) = RANK                                          DURBIN
ISN C079            WRITE (UT2) (DATA(I), I = 1, 6)                         DURBIN
ISN C080            GO TO 23                                                DURBIN
                C                                                           DURBIN
                C     END OF FILE - RANK LAST ITEM                          DURBIN
                C                                                           DURBIN
ISN C081         26 DATA(4) = RANK                                          DURBIN
```

```
ISN C082            WRITE (UT2) (DATA(I), I = 1, 6)                    DURBIN
ISN C083            GO TO 50                                           DURBIN
                C                                                      DURBIN
                C     PROCESS A TIED OBSERVATION AND ANY SUCCEEDING TIES DURBIN
                C                                                      DURBIN
ISN C084         31 PLCS = PLCS + 1                                    DURBIN
ISN C085            LIM = (WKSZE / 6) * 6                              DURBIN
ISN C086            IPTR = 0                                           DURBIN
ISN C087            DO 32 I = 1, 6                                     DURBIN
ISN C088         32 WORK(I) = DATA(I)                                  DURBIN
                C                                                      DURBIN
                C     SAVE THE TIED OBSERVATIONS AND READ THE NEXT ONE DURBIN
                C                                                      DURBIN
ISN C089         33 IPTR = IPTR + 6                                    DURBIN
ISN C090            IF (IPTR .EQ. LIM) GO TO 1236                      DURBIN
ISN C092            DO 34 I = 1, 6                                     DURBIN
ISN C093         34 WORK(IPTR+I) = RECORD(I)                           DURBIN
                C                                                      DURBIN
ISN C094         35 READ (UT4,END=41) RECORD                           DURBIN
ISN C095            IF (CURBLK .NE. IBLK) GO TO 35                     DURBIN
ISN C097            IF (DATA(6) .EQ. OBS) GO TO 33                     DURBIN
                C                                                      DURBIN
                C     AVERAGE RANKS FOR THIS SET OF TIES               DURBIN
                C                                                      DURBIN
ISN C099            J = (IPTR + 6) / 6                                 DURBIN
ISN C100            TIES = TIES + J                                    DURBIN
ISN C101            Z = 0.                                             DURBIN
ISN C102            DO 36 I = 1, J                                     DURBIN
ISN C103            Z = Z + RANK                                       DURBIN
ISN C104         36 RANK = RANK + 1.                                   DURBIN
ISN C105            Z = Z / J                                          DURBIN
ISN C106            JPTR = 0                                           DURBIN
                C                                                      DURBIN
ISN C107         37 WORK(JPTR+4) = Z                                   DURBIN
ISN C108            WRITE (UT2) (WORK(JPTR+I), I = 1, 6)               DURBIN
ISN 0109            JPTR = JPTR + 6                                    DURBIN
ISN 0110            IF (JPTR .LE. IPTR) GO TO 37                       DURBIN
ISN C112            RANK = RANK - 1.                                   DURBIN
ISN C113            GO TO 26                                           DURBIN
                C                                                      DURBIN
                C     END OF FILE DURING A TIE                         DURBIN
                C                                                      DURBIN
ISN 0114         41 J = (IPTR + 6) / 6                                 DURBIN
```

```
ISN C115            TIES = TIES + J                                                 DURBIN
ISN C116            Z = 0.                                                          DURBIN
                C                                                                   DURBIN
ISN C117            DO 42 I = 1, J                                                  DURBIN
ISN C118            Z = Z + RANK                                                    DURBIN
ISN C119         42 RANK = RANK + 1.                                                DURBIN
ISN C120            Z = Z / J                                                       DURBIN
ISN C121            JPTR = 0                                                        DURBIN
                C                                                                   DURBIN
ISN C122         43 WORK(JPTR+4) = Z                                                DURBIN
ISN C123            WRITE (UT2) (WORK(JPTR+I), I = 1, 6)                            DURBIN
ISN C124            JPTR = JPTR + 6                                                 DURBIN
ISN C125            IF (JPTR .LE. IPTR) GO TO 43                                    DURBIN
                C                                                                   DURBIN
                C     END OF BLOCK RANKING LOOP                                     DURBIN
                C                                                                   DURBIN
ISN C127         50 CONTINUE                                                        DURBIN
ISN C128            ENDFILE UT2                                                     DURBIN
ISN C129            REWIND UT2                                                      DURBIN
                C                                                                   DURBIN
                C     CALCULATE THE SUM OF THE RANKS WITHIN TREATMENTS, ACROSS BLOCKS DURBIN
                C                                                                   DURBIN
ISN C130            DO 51 I = 1, TRT                                                DURBIN
ISN C131         51 DATA(I) = 0.                                                    DURBIN
                C                                                                   DURBIN
ISN C132         52 READ (UT2,END=56) RECORD                                        DURBIN
ISN C133            DATA(CURTRT) = DATA(CURTRT) + XRANK                             DURBIN
ISN C134            GO TO 52                                                        DURBIN
                C                                                                   DURBIN
                C     CALCULATE THE TEST STATISTIC                                  DURBIN
                C                                                                   DURBIN
ISN C135         56 T = 0.                                                          DURBIN
ISN C136            DO 57 I = 1, TRT                                                DURBIN
ISN C137         57 T = T + DATA(I) ** 2                                            DURBIN
                C                                                                   DURBIN
ISN C138            TM1 = TRT - 1                                                   DURBIN
ISN C139            XKM1 = TRTB - 1                                                 DURBIN
ISN C140            XKP1 = TRTB + 1                                                 DURBIN
ISN C141            R = BLKT                                                        DURBIN
ISN C142            XT = TRT                                                        DURBIN
ISN C143            T = 12. * TM1 * T / (R * XT * XKM1 * XKP1)                      DURBIN
ISN 0144            T = T - 3. * R * TM1 * XKP1 / XKM1                              DURBIN
                C                                                                   DURBIN
```

```
ISN C145            IDF = TRT - 1                                          DURBIN
ISN C146            DF = IDF                                               DURBIN
ISN C147            IF (TIES .EQ. 0) GO TO 62                              DURBIN
ISN C149            WRITE (LST,305) TIES,PLCS                              DURBIN
ISN 0150            CALL OFLOW(3)                                          DURBIN
                C                                                          DURBIN
ISN C151         62 WRITE (LST,304) T,IDF                                  DURBIN
                C                                                          DURBIN
                C     DETERMINE CRITICAL REGION                            DURBIN
                C                                                          DURBIN
ISN C152            ALPHAT = CDTR(T,DF)                                    DURBIN
                C                                                          DURBIN
ISN C153         91 I = 1                                                  DURBIN
ISN C154            IF (ALPHA .GT. ALPHAT) I = 2                           DURBIN
ISN C156            WRITE (LST,306) ALPHA,ACCREJ(I)                        DURBIN
ISN C157            WRITE (LST,307) ALPHAT                                 DURBIN
ISN C158            CALL OFLOW(7)                                          DURBIN
                C                                                          DURBIN
                C     END OF THE TEST                                      DURBIN
                C                                                          DURBIN
ISN C159         92 WRITE (LST,308)                                        DURBIN
ISN C160            CALL OFLOW(3)                                          DURBIN
ISN C161            RETURN                                                 DURBIN
                C                                                          DURBIN
                C     ERRORS THAT CAN OCCUR                                DURBIN
                C                                                          DURBIN
ISN C162       1234 WRITE (LST,234)                                        DURBIN
ISN C163            GO TO 99                                               DURBIN
                C                                                          DURBIN
ISN C164       1236 WRITE (LST,236)                                        DURBIN
                C                                                          DURBIN
ISN C165         99 CALLOFLOW(3)                                           DURBIN
ISN C166            RC = 12                                                DURBIN
ISN C167            GO TO 92                                               DURBIN
                C                                                          DURBIN
ISN 0168            END                                                    DURBIN

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     167 ,PROGRAM SIZE =     3362

*STATISTICS*  NO  DIAGNOSTICS GENERATED
```

****** END OF COMPILATION ******

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002            SUBROUTINE FRDMAN                                                  FRDMAN
              C                                                                         FRDMAN
              C     THE FRIEDMAN TEST                                                   FRDMAN
              C                                                                         FRDMAN
              C     THE FRIEDMAN TEST IS USED ON DATA OF THE RANDOMIZED COMPLETE        FRDMAN
              C     BLOCK FORM OR ON A MULTIVARIATE SAMPLE IN WHICH EACH OBSERVATION    FRDMAN
              C     CONSISTS OF A BLOCK AND EACH VARIATE REPRESENTS A TREATMENT.        FRDMAN
              C     EACH BLOCK-TREATMENT COMBINATION CAN HAVE 1 OR MORE                 FRDMAN
              C     OBSERVATIONS.  IN THE CASE OF MULTIVARIATE DATA,                    FRDMAN
              C     ONLY 1 OBSERVATION PER COMBINATION IS ALLOWED. THE TEST             FRDMAN
              C     TESTS THE HYPOTHESIS OF EQUAL TREATMENT EFFECTS.                    FRDMAN
              C                                                                         FRDMAN
              C     FOR DETAILS ON THE FRIEDMAN TEST, SEE CONOVER, W.J., PRACTICAL      FRDMAN
              C     NONPARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 418-434.               FRDMAN
              C                                                                         FRDMAN
              C                                                                         FRDMAN
              C     COMMON BLOCKS                                                       FRDMAN
              C                                                                         FRDMAN
 ISN 0003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                        DECLARE
 ISN 0004            INTEGER*4  ROWS,COLS                                                DECLARE
 ISN 0005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                            DECLARE
 ISN 0006            INTEGER*4  DISTN,WHTRN                                              DECLARE
 ISN 0007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                         DECLARE
 ISN 0008            REAL*8  BUFFER                                                      DECLARE
 ISN 0009            INTEGER*4  EOD,CDCOL,BUFNO                                          DECLARE
 ISN 0010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD               DECLARE
 ISN 0011            REAL*8   FMT                                                        DECLARE
 ISN 0012            INTEGER*4  DFORM,DOUT,RCBCD                                         DECLARE
 ISN 0013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),         DECLARE
                    1    VALUE(40),IVALUE(40),RC                                         DECLARE
 ISN 0014            REAL*8   PARM,SUBPRM                                                DECLARE
 ISN 0015            INTEGER*4  RC                                                       DECLARE
 ISN 0016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                         DECLARE
 ISN 0017            INTEGER*4  TAIL                                                     DECLARE
 ISN 0018            LOGICAL*4  TESTS,DOALL                                              DECLARE
 ISN 0019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),             DECLARE
                    1    DATE,TIME,PAGE,LINE                                             DECLARE
 ISN 0020            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                    DECLARE
 ISN 0021            INTEGER*4  PAGE                                                     DECLARE
 ISN 0022            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)   DECLARE
 ISN 0023            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE       DECLARE
```

```
ISN C024            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                           DECLARE
ISN C027            INTEGER*4  ITEMS,LOCAT,RDNO                                        DECLARE
ISN C028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                       DECLARE
ISN C029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                             DECLARE
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                                      DECLARE
ISN C031            REAL*8  BLANK /' '/                                                DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                         DECLARE
                    C                                                                  FRDMAN
                    C     OTHER DATA NEEDED                                            FRDMAN
                    C                                                                  FRDMAN
ISN C033            REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                         FRDMAN
ISN C034            REAL*8  BLKLBL /'BLOCKS'/, TRTLBL /'TRTMNTS'/                      FRDMAN
ISN C035            REAL*4  RECORD(6)                                                  FRDMAN
ISN C036            INTEGER*4  TIES,PLCS                                               FRDMAN
ISN C037            INTEGER*4  CURSMP,CURVAR,CURCNT,CURBLK,CURTRT,CURREP               FRDMAN
ISN C038            EQUIVALENCE (RECORD(1),CURSMP,CURBLK), (RECORD(2),CURVAR,CURTRT),  FRDMAN
                   1    (RECORD(3),CURCNT,CURREP), (RECORD(4),XRANK),                  FRDMAN
                   2    (RECORD(6),OBS)                                               FRDMAN
                    C                                                                  FRDMAN
                    C     FORMAT STATEMENTS                                            FRDMAN
                    C                                                                  FRDMAN
ISN C039          231 FORMAT ('-', T12, 'IKMS31I  UNEQUAL SAMPLE/VARIATE SIZES AFTER TRAFRDMAN
                   1NSFORMATION')                                                      FRDMAN
ISN C040          234 FORMAT ('-', T12, 'IKMS34I  INVALID SAMPLE SIZE(S)')            FRDMAN
ISN C041          236 FORMAT ('-', T12, 'IKMS36I  TOO MANY TIED VALUES FOR THIS PROGRAM FRDMAN
                   1TO HANDLE')                                                        FRDMAN
ISN C042          301 FORMAT ('-', T60, 'THE FRIEDMAN TEST')                           FRDMAN
ISN C043          302 FORMAT ('0', T12, 'H0: THE', I3, ' TREATMENTS, CALLED ', A8,     FRDMAN
                   1    ', ARE EQUALLY EFFECTIVE ACROSS BLOCKS, CALLED ', A8/         FRDMAN
                   2    '0', T12, 'VERSUS'/ '0', T12,                                 FRDMAN
                   3    'H1: AT LEAST 1 TREATMENTS YIELDS LARGER OBSERVATIONS THAN THEFRDMAN
                   4 OTHER TREATMENTS')                                                FRDMAN
ISN C044          303 FORMAT ('-', T12, 'THE OBSERVATIONS ARE TAKEN ACROSS', I3,       FRDMAN
                   1    ' BLOCKS')                                                    FRDMAN
ISN C045          304 FORMAT ('0', T12, 'EACH BLOCK-TREATMENT COMBINATION IS OBSERVED', FRDMAN
                   1    I3, ' TIMES')                                                 FRDMAN
ISN C046          305 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5, ' WITH', I3,  FRDMAN
                   1    ' DEGREES OF FREEDOM')                                        FRDMAN
ISN C047          306 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', H0 IS ', A8, '.')       FRDMAN
ISN C048          307 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                           FRDMAN
ISN C049          308 FORMAT ('-', T56, 'END OF THE FRIEDMAN TEST')                    FRDMAN
```

```
ISN C05C        309 FORMAT ('-', T12, 'THERE WAS A TOTAL OF', I5, ' TIES IN',     FRDMAN
                   1    I3, ' PLACES')                                           FRDMAN
                C                                                                FRDMAN
                C     START WITH THE TITLE AND HYPOTHESES                        FRDMAN
                C                                                                FRDMAN
ISN C051              WRITE (LST,301)                                            FRDMAN
ISN C052              IF (BLOCK .EQ. BLANK) BLOCK = BLKLBL                       FRDMAN
ISN C054              IF (TREAT .EQ. BLANK) TREAT = TRTLBL                       FRDMAN
                C                                                                FRDMAN
ISN C056              IF (TYPE .EQ. 7) GO TO 11                                  FRDMAN
ISN C058              BLK = SSIZE(1)                                             FRDMAN
ISN C059              TRT = VARITE                                               FRDMAN
ISN C06C              REP = 1                                                    FRDMAN
                C                                                                FRDMAN
ISN C061           11 WRITE (LST,302) TRT,TREAT,BLOCK                            FRDMAN
ISN CC62              WRITE (LST,303) BLK                                        FRDMAN
ISN CC63              WRITE (LST,304) REP                                        FRDMAN
ISN C064              CALL OFLOW(14)                                             FRDMAN
                C                                                                FRDMAN
                C     CHECK FOR MULTIVARIATE DATA                                FRDMAN
                C                                                                FRDMAN
ISN C065              IF (TYPE .EQ. 7) GO TO 21                                  FRDMAN
ISN CC67              REWIND UT1                                                 FRDMAN
ISN CC68              REWIND UT2                                                 FRDMAN
ISN C069              DO 12 I = 1, VARITE                                        FRDMAN
ISN CC70           12 IWRK(I) = 0                                                FRDMAN
                C                                                                FRDMAN
ISN CC71           13 READ (UT1,END=14) RECORD                                   FRDMAN
ISN CC72              CURREP = 1                                                 FRDMAN
ISN CC73              IWRK(CURVAR) = IWRK(CURVAR) + 1                            FRDMAN
ISN CC74              CURBLK = IWRK(CURVAR)                                      FRDMAN
ISN CC75              WRITE (UT2) RECORD                                         FRDMAN
ISN CC76              GO TO 13                                                   FRDMAN
                C                                                                FRDMAN
ISN CC77           14 ENDFILE UT2                                                FRDMAN
ISN CC78              DO 15 I = 1, VARITE                                        FRDMAN
ISN C079              IF (IWRK(I) .NE. BLK) GO TO 1234                           FRDMAN
ISN C081           15 CONTINUE                                                   FRDMAN
                C                                                                FRDMAN
ISN C082              CALL RANKER (UT4,UT2,LT4,UT3,J)                            FRDMAN
ISN CC83              GO TO 1021                                                 FRDMAN
                C                                                                FRDMAN
                C     SORT RCB DATA                                              FRDMAN
```

```
                C                                                                FRDMAN
ISN C084           21 CALL RANKER (UT4,UT1,UT2,UT3,J)                            FRDMAN
ISN C085         1C21 IF (RC .NE. 0) GO TO 92                                    FRDMAN
ISN C087              SUBS(1) = C                                                FRDMAN
ISN C088              SUBS(2) = 0                                                FRDMAN
                C                                                                FRDMAN
                C     NOW ALL THE DATA IS SORTED AND LABELED BY BLOCKS, TREATMENTS, FRDMAN
                C     AND REPLICATIONS.  ASSIGN RANKS WITHIN BLOCKS.             FRDMAN
                C                                                                FRDMAN
ISN C089              TIES = 0                                                   FRDMAN
ISN C090              PLCS = 0                                                   FRDMAN
ISN C091              REWIND UT2                                                 FRDMAN
ISN C092              DO 50 IBLK = 1, BLK                                        FRDMAN
ISN C093              REWIND UT4                                                 FRDMAN
ISN C094              RANK = 0.                                                  FRDMAN
                C                                                                FRDMAN
ISN C095           22 READ (UT4,END=1231) RECORD                                 FRDMAN
ISN C096              IF (CURBLK .NE. IBLK) GO TO 22                             FRDMAN
                C                                                                FRDMAN
                C     PROCESS AN OBSERVATION FOR THIS BLOCK                      FRDMAN
                C                                                                FRDMAN
ISN C098           23 DO 24 I = 1, 6                                             FRDMAN
ISN C099           24 DATA(I) = RECORD(I)                                        FRDMAN
ISN C100              RANK = RANK + 1.                                           FRDMAN
                C                                                                FRDMAN
                C     READ THE NEXT OBSERVATION AND CHECK FOR A TIE              FRDMAN
                C                                                                FRDMAN
ISN C101           25 READ (UT4,END=26) RECORD                                   FRDMAN
ISN C102              IF (CURBLK .NE. IBLK) GO TO 25                             FRDMAN
ISN C104              IF (DATA(6) .EQ. OBS) GO TO 31                             FRDMAN
                C                                                                FRDMAN
                C     RANK THIS UNTIED OBSERVATION                               FRDMAN
                C                                                                FRDMAN
ISN C106              DATA(4) = RANK                                             FRDMAN
ISN C107              WRITE (UT2) (DATA(I), I = 1, 6)                            FRDMAN
ISN C108              GO TO 23                                                   FRDMAN
                C                                                                FRDMAN
                C     END OF FILE - RANK LAST ITEM                               FRDMAN
                C                                                                FRDMAN
ISN C109           26 DATA(4) = RANK                                             FRDMAN
ISN C110              WRITE (UT2) (DATA(I), I = 1, 6)                            FRDMAN
ISN C111              GO TO 50                                                   FRDMAN
                C                                                                FRDMAN
```

```
                C     HANDLE TIED OBSERVATIONS                                  FRDMAN
                C                                                               FRDMAN
ISN C112           31 PLCS = PLCS + 1                                           FRDMAN
ISN C113              LIM = (WKSZE / 6) * 6                                     FRDMAN
ISN C114              IPTR = 0                                                  FRDMAN
ISN C115              DO 32 I = 1, 6                                            FRDMAN
ISN C116           32 WORK(I) = DATA(I)                                         FRDMAN
                C                                                               FRDMAN
                C     SAVE THE TIED OBSERVATION WITH THE PREVIOUS ONES          FRDMAN
                C                                                               FRDMAN
ISN C117           33 IPTR = IPTR + 6                                           FRDMAN
ISN C118              IF (IPTR .EQ. LIM) GO TO 1236                             FRDMAN
ISN C120              DO 34 I = 1, 6                                            FRDMAN
ISN C121           34 WORK(IPTR+I) = RECORD(I)                                  FRDMAN
                C                                                               FRDMAN
                C     READ THE NEXT OBSERVATION AND CHECK FOR FURTHER TIES      FRDMAN
                C                                                               FRDMAN
ISN C122           35 READ (UT4,END=41) RECORD                                  FRDMAN
ISN C123              IF (CURBLK .NE. IBLK) GO TO 35                            FRDMAN
ISN C125              IF (DATA(6) .EQ. OBS) GO TO 33                            FRDMAN
                C                                                               FRDMAN
                C     END OF THIS SET OF TIES                                   FRDMAN
                C                                                               FRDMAN
ISN C127              J = (IPTR + 6) / 6                                        FRDMAN
ISN C128              TIES = TIES + J                                           FRDMAN
ISN C129              Z = 0.                                                    FRDMAN
                C                                                               FRDMAN
ISN C130              DO 36 I = 1, J                                            FRDMAN
ISN C131              Z = Z + RANK                                              FRDMAN
ISN C132           36 RANK = RANK + 1.                                          FRDMAN
ISN C133              Z = Z / J                                                 FRDMAN
ISN C134              JPTR = 0                                                  FRDMAN
                C                                                               FRDMAN
ISN C135           37 WORK(JPTR+4) = Z                                          FRDMAN
ISN C136              WRITE (UT2) (WORK(JPTR+I), I = 1, 6)                      FRDMAN
ISN C137              JPTR = JPTR + 6                                           FRDMAN
ISN C138              IF (JPTR .LE. IPTR) GO TO 37                              FRDMAN
ISN C140              RANK = RANK - 1.                                          FRDMAN
ISN C141              GO TO 26                                                  FRDMAN
                C                                                               FRDMAN
                C     EOF ON A TIE                                              FRDMAN
                C                                                               FRDMAN
ISN 0142           41 J = (IPTR + 6) / 6                                        FRDMAN
```

```
ISN C143            TIES = TIES + J                                          FRDMAN
ISN C144            Z = C.                                                   FRDMAN
ISN C145            DO 42 I = 1, J                                           FRDMAN
ISN C146            Z = Z + RANK                                             FRDMAN
ISN C147         42 RANK = RANK + 1.                                         FRDMAN
ISN C148            Z = Z / J                                                FRDMAN
ISN C149            JPTR = 0                                                 FRDMAN
                 C                                                           FRDMAN
ISN C150         43 WORK(JPTR+4) = Z                                         FRDMAN
ISN C151            WRITE (UT2) (WORK(JPTR+I), I = 1, 6)                     FRDMAN
ISN C152            JPTR = JPTR + 6                                          FRDMAN
ISN C153            IF (JPTR .LE. IPTR) GO TO 43                             FRDMAN
                 C                                                           FRDMAN
                 C     END OF RANKING THIS BLOCK                             FRDMAN
                 C                                                           FRDMAN
ISN C155         5C CONTINUE                                                 FRDMAN
                 C                                                           FRDMAN
ISN C156            ENDFILE UT2                                              FRDMAN
ISN C157            REWIND UT2                                               FRDMAN
                 C                                                           FRDMAN
                 C     SUM UP THE RANKS ACROSS BLOCKS, WITHIN TREATMENTS     FRDMAN
                 C                                                           FRDMAN
ISN 0158            CF = BLK * REP * (TRT * REP + 1)                         FRDMAN
ISN C159            CF = CF / 2.                                             FRDMAN
ISN C160            DO 51 I = 1, TRT                                         FRDMAN
ISN C161         51 DATA(I) = 0.                                             FRDMAN
                 C                                                           FRDMAN
ISN C162         52 READ (UT2,END=56) RECORD                                 FRDMAN
ISN 0163            DATA(CURTRT) = DATA(CURTRT) + XRANK                      FRDMAN
ISN C164            GO TO 52                                                 FRDMAN
                 C                                                           FRDMAN
                 C     END OF RANK SUMMING - CALCULATE TEST STATISTIC        FRDMAN
                 C                                                           FRDMAN
ISN C165         56 T = 0.                                                   FRDMAN
ISN C166            DO 57 I = 1, TRT                                         FRDMAN
ISN C167         57 T = T + (DATA(I) - CF) ** 2                              FRDMAN
                 C                                                           FRDMAN
ISN C168            D = BLK * TRT * REP * REP * (TRT * REP + 1)              FRDMAN
ISN C169            T = 12. * T / D                                          FRDMAN
ISN C170            IDF = TRT - 1                                            FRDMAN
ISN C171            DF = IDF                                                 FRDMAN
                 C                                                           FRDMAN
ISN C172            IF (TIES .EQ. 0) GO TC 61                                FRDMAN
```

```
ISN C174            WRITE (LST,309) TIES,PLCS                                  FRDMAN
ISN C175            CALL OFLOW(3)                                              FRDMAN
                C                                                              FRDMAN
ISN C176          61 WRITE (LST,305) T,IDF                                     FRDMAN
                C                                                              FRDMAN
                C     DETERMINE CRITICAL REGION AND CONCLUSION TO REACH        FRDMAN
                C                                                              FRDMAN
ISN C177            ALFHAT = CDTR(T,DF)                                        FRDMAN
                C                                                              FRDMAN
ISN C178          91 I = 1                                                     FRDMAN
ISN C179            IF (ALPHA .GT. ALFHAT) I = 2                               FRDMAN
ISN C181            WRITE (LST,306) ALPHA,ACCREJ(I)                            FRDMAN
ISN C182            WRITE (LST,307) ALFHAT                                     FRDMAN
ISN C183            CALL CFLOW(7)                                              FRDMAN
                C                                                              FRDMAN
                C     END OF THE TEST                                          FRDMAN
                C                                                              FRDMAN
ISN C184          92 WRITE (LST,308)                                           FRDMAN
ISN C185            CALL OFLOW(3)                                              FRDMAN
ISN C186            RETURN                                                     FRDMAN
                C                                                              FRDMAN
                C     ERRORS THAT MIGHT OCCUR                                  FRDMAN
                C                                                              FRDMAN
ISN C187         1231 WRITE (LST,231)                                          FRDMAN
ISN C188            GO TO 99                                                   FRDMAN
                C                                                              FRDMAN
ISN C189         1234 WRITE (LST,234)                                          FRDMAN
ISN C190            GO TO 99                                                   FRDMAN
                C                                                              FRDMAN
ISN C191         1236 WRITE (LST,236)                                          FRDMAN
                C                                                              FRDMAN
ISN C192          99 CALL OFLOW(3)                                             FRDMAN
ISN C193            RC = 12                                                    FRDMAN
ISN C194            GO TO 92                                                   FRDMAN
                C                                                              FRDMAN
ISN C195            END                                                        FRDMAN

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     194 ,PROGRAM SIZE =     3626

*STATISTICS*  NO  DIAGNOSTICS GENERATED
```

****** END OF COMPILATION ******

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            SUBROUTINE GDFIT                                                    GDFIT
                    C                                                                   GDFIT
                    C     THE CHI-SQUARE GOODNESS OF FIT TEST                           GDFIT
                    C                                                                   GDFIT
                    C     THE CHI-SQUARE GOODNESS OF FIT TEST IS A TEST DESIGNED        GDFIT
                    C     TO TEST HOW WELL A RANDOM SAMPLE FITS A PARTICULAR DISTRIBUTION GDFIT
                    C     FUNCTION.  THE HYPOTHESIS BEING TESTED IS THAT THE SAMPLE     GDFIT
                    C     COMES FROM A POPULATION WITH THE SPECIFIED DISTRIBUTION       GDFIT
                    C     FUNCTION.  THE DISTRIBUTION FUNCTIONS THAT ARE CURRENTLY      GDFIT
                    C     IMPLIMENTED ARE:  NORMAL, WITH SPECIFIED MEAN AND VARIANCE;   GDFIT
                    C     BINOMIAL, WITH SPECIFIED SIZE AND PROBABILITY; UNIFORM; AND   GDFIT
                    C     POISSON, WITH SPECIFIED MEAN.  TO PERFORM THE TEST, THE SAMPLE GDFIT
                    C     MUST BE DICHOTOMIZED INTO "C" CLASSES, WHICH THE USER MUST    GDFIT
                    C     PROVIDE.  POOR OR IMPROPER SELECTION OF THE CLASSES WILL RESULT GDFIT
                    C     IN A POOR TEST.                                               GDFIT
                    C                                                                   GDFIT
                    C     FOR DETAILS, SEE CONOVER, W.J., PRACTICAL NONPARAMETRIC       GDFIT
                    C     STATISTICS, ROUGH DRAFT, PAGES 293-306.                       GDFIT
                    C                                                                   GDFIT
                    C                                                                   GDFIT
                    C     COMMON BLOCKS                                                 GDFIT
                    C                                                                   GDFIT
ISN C003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                        DECLARE
ISN C004            INTEGER*4  ROWS,COLS                                                DECLARE
ISN C005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                            DECLARE
ISN C006            INTEGER*4  DISTN,WHTRN                                              DECLARE
ISN C007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                         DECLARE
ISN C008            REAL*8  BUFFER                                                      DECLARE
ISN C009            INTEGER*4  EOD,CDCOL,BUFNO                                          DECLARE
ISN C010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD               DECLARE
ISN C011            REAL*8   FMT                                                        DECLARE
ISN C012            INTEGER*4  DFORM,DOUT,RCBCD                                         DECLARE
ISN C013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),         DECLARE
                   1    VALUE(40),IVALUE(40),RC                                         DECLARE
ISN C014            REAL*8   PARM,SUBPRM                                                DECLARE
ISN C015            INTEGER*4  RC                                                       DECLARE
ISN C016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                         DECLARE
ISN C017            INTEGER*4  TAIL                                                     DECLARE
ISN C018            LOGICAL*4  TESTS,DOALL                                              DECLARE
ISN C019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),             DECLARE
                   1    DATE,TIME,PAGE,LINE                                             DECLARE
```

```
ISN C020            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                    DECLARE
ISN C021            INTEGER*4  PAGE                                                     DECLARE
ISN C022            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)  DECLARE
ISN C023            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE      DECLARE
ISN C024            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                           DECLARE
ISN C027            INTEGER*4  ITEMS,LOCAT,RDNO                                        DECLARE
ISN C028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                       DECLARE
ISN C029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                             DECLARE
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                                      DECLARE
ISN C031            REAL*8  BLANK /' '/                                                DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                         DECLARE
                 C                                                                     GDFIT
                 C     OTHER DATA ITEMS                                                GDFIT
                 C                                                                     GDFIT
ISN C033            REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                         GDFIT
ISN C034            REAL*4  LOVAL /-1.E75/, HIVAL /+1.E75/                             GDFIT
ISN C035            REAL*4  RECORD(6)                                                  GDFIT
ISN C036            INTEGER*4  CURSMP,CURVAR,CURCNT                                    GDFIT
ISN C037            EQUIVALENCE (RECORD(1),CURSMP), (RECORD(2),CURVAR),                GDFIT
                   1    (RECORD(3),CURCNT), (RECORD(6),OBS)                           GDFIT
ISN C038            REAL*8  P1, P2                                                     GDFIT
                 C                                                                     GDFIT
                 C     FORMAT STATEMENTS                                               GDFIT
ISN C039          232 FORMAT ('-', T12, 'IKMS32I  TOO FEW COLUMN CLASSES SPECIFIED')   GDFIT
ISN C040          233 FORMAT ('-', T12, 'IKMS33I  TOO FEW DICHOTOMIZATION LEVELS SPECIFIGDFIT
                     1ED')                                                             GDFIT
ISN C041          234 FORMAT ('-', T12, 'IKMS34I  INVALID SAMPLE SIZE(S)')             GDFIT
ISN C042          291 FORMAT ('-', T12, 'IKMS92I  ***** EXPECTED VALUES MAY BE TOO SMALLGDFIT
                     1 FOR VALID TEST. *****')                                         GDFIT
ISN C043          3C1 FORMAT ('-', T51, 'THE CHI-SQUARE GOODNESS OF FIT TEST')         GDFIT
ISN C044          3C2 FORMAT ('0', T12, 'HO: THE SAMPLE COMES FROM A NORMAL POPULATION WGDFIT
                     1ITH MEAN', G12.4, ' AND VARIANCE', G12.4)                        GDFIT
ISN C045          3C3 FORMAT ('0', T12, 'HO: THE SAMPLE COMES FROM A POPULATION WITH A PGDFIT
                     1OISSON DISTRIBUTION OF MEAN', G12.4)                             GDFIT
ISN C046          3C4 FORMAT ('0', T12, 'HO: THE SAMPLE COMES FROM A UNIFORMLY DISTRIBUTGDFIT
                     1ED POPULATION')                                                  GDFIT
ISN C047          3C5 FORMAT ('0', T12, 'HO: THE SAMPLE COMES FROM A POPULATION WITH A BGDFIT
                     1INOMIAL DISTRIBUTION OF PROBABILITY', F6.3, ' AND SIZE', I5)     GDFIT
ISN C048          3C6 FORMAT ('0', T12, 'VERSUS'/ '0', T12, 'H1: THE SAMPLE DOES NOT COMGDFIT
                     1E FROM A POPULATION AS DESCRIBED IN HO')                         GDFIT
ISN C049          3C7 FORMAT ('0', T12, 'THE SAMPLE WAS BROKEN DOWN AS FOLLOWS:'/      GDFIT
```

```
                  1    '0', T12, 'CLASS', T25, 'CLASS INTERVAL', T52, 'OBSERVED',     GDFIT
                  2    T67, 'EXPECTED')                                               GDFIT
ISN C050        308 FORMAT ('0', T12, I3, T22, G10.4, ' TO ', G10.4, T52, I5,         GDFIT
                  1    T67, F8.3)                                                     GDFIT
ISN C051        309 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5, ' WITH',       GDFIT
                  1    I3, ' DEGREES OF FREEDOM')                                     GDFIT
ISN C052        310 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')        GDFIT
ISN C053        311 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                            GDFIT
ISN C054        312 FORMAT ('-', T47, 'END OF THE CHI-SQUARE GOODNESS OF FIT TEST')   GDFIT
ISN C055        313 FORMAT ('0', T12, 'TOTAL', T52, I5)                               GDFIT
                C                                                                     GDFIT
                C     START WITH THE TITLE AND HYPOTHESES                             GDFIT
                C                                                                     GDFIT
ISN C056            WKSZE = WKSZE                                                     GDFIT
ISN C057            WRITE (LST,301)                                                   GDFIT
ISN C058            CALL OFLOW(3)                                                     GDFIT
ISN C059            GO TO (11, 12, 13, 14), DISTN                                     GDFIT
                C                                                                     GDFIT
ISN C060         11 WRITE (LST,302) DPARM1,DPARM2                                     GDFIT
ISN C061            GO TO 15                                                          GDFIT
                C                                                                     GDFIT
ISN C062         12 WRITE (LST,303) DPARM1                                            GDFIT
ISN C063            GO TO 15                                                          GDFIT
                C                                                                     GDFIT
ISN C064         13 WRITE (LST,304)                                                   GDFIT
ISN C065            GO TO 15                                                          GDFIT
                C                                                                     GDFIT
ISN C066         14 I = DPARM1                                                        GDFIT
ISN C067            WRITE (LST,305) DPARM2,I                                          GDFIT
                C                                                                     GDFIT
ISN C068         15 WRITE (LST,306)                                                   GDFIT
ISN C069            CALL OFLOW(6)                                                     GDFIT
                C                                                                     GDFIT
                C     CHECK COLUMN SPECIFICATIONS FOR SUITABILITY                     GDFIT
                C                                                                     GDFIT
ISN C070            IF (COLS .LT. 3) GO TO 1232                                       GDFIT
ISN C072            DATA(1) = LOVAL                                                   GDFIT
ISN C073            DO 16 I = 1, COLS                                                 GDFIT
ISN C074         16 DATA(I+1) = CLEV(I)                                               GDFIT
ISN C075            DATA(COLS+1) = HIVAL                                              GDFIT
ISN C076            DO 17 I = 1, COLS                                                 GDFIT
ISN C077            IF (DATA(I) .GE. DATA(I+1)) GO TO 1233                            GDFIT
ISN C079         17 CONTINUE                                                          GDFIT
```

```
                C                                                                       GDFIT
                C     READ THE SAMPLE ONE ITEM AT A TIME                                GDFIT
                C                                                                       GDFIT
ISN C080              DO 18 I = 1, CCLS                                                 GDFIT
ISN C081           18 IWRK(I) = C                                                       GDFIT
ISN C082              ISZ = 0                                                           GDFIT
ISN C083              REWIND UT1                                                        GDFIT
                C                                                                       GDFIT
ISN C084           21 READ (UT1,END=31) RECORD                                          GDFIT
ISN C085              X(1) = OBS                                                        GDFIT
ISN C086              ITEMS = 1                                                         GDFIT
ISN C087              LOCAT(4) = 0                                                      GDFIT
ISN C088              IF (WHTRN .EQ. 1) CALL TRANS                                      GDFIT
ISN C090              IF (ITEMS .LT. 1) GO TO 21                                        GDFIT
ISN C092              IF (OUT .NE. 0 .AND. COUT .EQ. 2) CALL WRDATA                     GDFIT
                C                                                                       GDFIT
                C     FIND HOLE FOR EACH OBSERVATION                                    GDFIT
                C                                                                       GDFIT
ISN C094              DO 26 I = 1, ITEMS                                                GDFIT
                C                                                                       GDFIT
ISN C095              DO 24 J = 1, CCLS                                                 GDFIT
ISN C096              IF (X(I) .LE. DATA(J+1)) GO TO 25                                 GDFIT
ISN C098           24 CONTINUE                                                          GDFIT
ISN C099              GO TO 26                                                          GDFIT
ISN C100           25 IWRK(J) = IWRK(J) + 1                                             GDFIT
ISN C101              ISZ = ISZ + 1                                                     GDFIT
                C                                                                       GDFIT
ISN C102           26 CONTINUE                                                          GDFIT
ISN C103              GO TO 21                                                          GDFIT
                C                                                                       GDFIT
                C     END OF DATA PASS                                                  GDFIT
                C                                                                       GDFIT
ISN C104           31 IF (OUT .NE. 0 .AND. COUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT  GDFIT
ISN C106              OUT = 0                                                           GDFIT
                C                                                                       GDFIT
                C     DETERMINE THE EXPECTED VALUES                                     GDFIT
                C                                                                       GDFIT
ISN C107              IF (ISZ .LT. CCLS) GO TO 1234                                     GDFIT
ISN C109              DO 32 I = 1, CCLS                                                 GDFIT
ISN C110           32 WORK(CCLS+1) = 0.                                                 GDFIT
ISN C111              P1 = DPARM1                                                       GDFIT
ISN C112              P2 = DPARM2                                                       GDFIT
                C                                                                       GDFIT
```

```
ISN C113            GO TO (41, 42, 43, 44), DISTN                          GDFIT
                C                                                          GDFIT
                C       NORMAL DISTRIBUTION                                GDFIT
                C                                                          GDFIT
ISN C114           41 P1 = DSQRT(P2)                                       GDFIT
ISN C115              L = COLS - 1                                         GDFIT
ISN C116              DO 1041 I = 1, L                                     GDFIT
ISN C117              E = (DATA(I+1) - DPARM1) / P1                        GDFIT
ISN C118              P = ANDTR(E)                                         GDFIT
ISN C119         1041 WORK(COLS+I) = P                                     GDFIT
ISN C120              GO TO 51                                             GDFIT
                C                                                          GDFIT
                C       POISSON DISTRIBUTION                               GDFIT
                C                                                          GDFIT
ISN C121           42 L = COLS - 1                                         GDFIT
ISN C122              DO 1042 I = 1, L                                     GDFIT
ISN C123              N = DATA(I+1)                                        GDFIT
ISN C124              P = POIS(DPARM1,N)                                   GDFIT
ISN C125         1042 WORK(COLS+I) = P                                     GDFIT
ISN C126              GO TO 51                                             GDFIT
                C                                                          GDFIT
                C       UNIFORM DISTRIBUTION                               GDFIT
                C                                                          GDFIT
ISN C127           43 P = 0.                                               GDFIT
ISN C128              Q = 1. / COLS                                        GDFIT
ISN C129              DO 1043 I = 1, COLS                                  GDFIT
ISN C130              P = P + Q                                            GDFIT
ISN C131         1043 WORK(COLS+I) = P                                     GDFIT
ISN C132              GO TO 51                                             GDFIT
                C                                                          GDFIT
                C       BINOMIAL DISTRIBUTION                              GDFIT
                C                                                          GDFIT
ISN C133           44 L = COLS - 1                                         GDFIT
ISN C134              N = DPARM1                                           GDFIT
ISN C135              DO 1044 I = 1, L                                     GDFIT
ISN C136              K = DATA(I+1)                                        GDFIT
ISN C137              P = BINOM (N,DPARM2,K)                               GDFIT
ISN C138         1044 WORK(COLS+I) = P                                     GDFIT
ISN C139              GO TO 51                                             GDFIT
                C                                                          GDFIT
ISN C140           51 WORK(COLS+COLS) = 1.                                 GDFIT
                C                                                          GDFIT
                C       DETERMINE EXPECTED VALUES FROM PROBABILITIES       GDFIT
```

```
                C                                                                GDFIT
ISN 0141              DO 52 I = 1, COLS                                          GDFIT
ISN 0142           52 WORK(COLS+I) = WORK(COLS+I) * ISZ                          GDFIT
                C                                                                GDFIT
ISN 0143              SUB = 0.                                                   GDFIT
ISN 0144              DO 53 I = 1, COLS                                          GDFIT
ISN 0145              WORK(COLS+I) = WORK(COLS+I) - SUB                          GDFIT
ISN 0146           53 SUB = SUB + WORK(COLS+I)                                   GDFIT
                C                                                                GDFIT
                C     WRITE OUT THE FREQUENCY TABLE                              GDFIT
                C                                                                GDFIT
ISN 0147              WRITE (LST,307)                                            GDFIT
ISN 0148              CALL OFLOW(4)                                              GDFIT
                C                                                                GDFIT
ISN 0149              DO 54 I = 1, COLS                                          GDFIT
ISN 0150              WRITE (LST,308) I,DATA(I),DATA(I+1),IWRK(I),WORK(COLS+I)   GDFIT
ISN 0151           54 CALL OFLOW(2)                                              GDFIT
                C                                                                GDFIT
ISN 0152              WRITE (LST,313) ISZ                                        GDFIT
ISN 0153              CALL OFLOW(2)                                              GDFIT
                C                                                                GDFIT
                C     CALCULATE TEST STATISTIC                                   GDFIT
                C                                                                GDFIT
ISN 0154              ILT1 = 0                                                   GDFIT
ISN 0155              T = 0.                                                     GDFIT
ISN 0156              DO 55 I = 1, COLS                                          GDFIT
ISN 0157              OJ2 = IWRK(I) ** 2                                         GDFIT
ISN 0158              IF (WORK(COLS+I) .LT. 4.) ILT1 = 1                         GDFIT
ISN 0160           55 IF (WORK(COLS+I) .GT. 0.) T = T + OJ2 / WORK(COLS+I)       GDFIT
ISN 0162              T = T - ISZ                                                GDFIT
ISN 0163              IDF = COLS - 1                                             GDFIT
ISN 0164              DF = IDF                                                   GDFIT
                C                                                                GDFIT
ISN 0165              WRITE (LST,309) T,IDF                                      GDFIT
ISN 0166              CALL OFLOW(2)                                              GDFIT
                C                                                                GDFIT
                C     DETERMINE THE CRITICAL REGION                              GDFIT
                C                                                                GDFIT
ISN 0167              ALFHAT = CDTR(T,DF)                                        GDFIT
                C                                                                GDFIT
ISN 0168           91 I = 1                                                      GDFIT
ISN 0169              IF (ALPHA .GT. ALFHAT) I = 2                               GDFIT
ISN 0171              WRITE (LST,310) ALPHA,ACCREJ(I)                            GDFIT
```

```
ISN C172            WRITE (LST,311) ALFHAT                                  GDFIT
ISN C173            CALL OFLOW(5)                                           GDFIT
ISN C174            IF (ILT1 .EQ. 0) GO TO 92                               GDFIT
ISN C176            WRITE (LST,291)                                         GDFIT
ISN C177            CALL OFLOW(3)                                           GDFIT
                C                                                           GDFIT
                C     END OF THE GOODNESS OF FIT TEST                       GDFIT
                C                                                           GDFIT
ISN C178         92 WRITE (LST,312)                                         GDFIT
ISN C179            CALL OFLOW(3)                                           GDFIT
ISN C180            RETURN                                                  GDFIT
                C                                                           GDFIT
                C     DATA OR COLUMN SPECIFICATION ERRORS                   GDFIT
                C                                                           GDFIT
ISN C181       1232 WRITE (LST,232)                                         GDFIT
ISN C182            GO TO 99                                                GDFIT
                C                                                           GDFIT
ISN C183       1233  WRITE (LST,233)                                        GDFIT
ISN C184            GO TO 99                                                GDFIT
                C                                                           GDFIT
ISN C185       1234 WRITE (LST,234)                                         GDFIT
                C                                                           GDFIT
ISN C186         99 CALL OFLOW(3)                                           GDFIT
ISN C187            RC = 12                                                 GDFIT
ISN C188            GO TO 92                                                GDFIT
                C                                                           GDFIT
ISN C189            END                                                     GDFIT

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     188 ,PROGRAM SIZE =     4026

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002            SUBROUTINE HEAD                                                  HEAD
                C                                                                      HEAD
                C     THIS SUBROUTINE PRINTS THE STANDARD PAGE HEADING FOR THE PROGRAM  HEAD
                C     AND RESETS THE PAGE LINE COUNT TO 5 (THE NUMBER OF LINES          HEAD
                C     PRINTED IN THE HEADING).  THE PAGE NUMBER, WHICH IS PRINTED       HEAD
                C     AS PART OF THR HEADING IS INCREMENTED BEFORE PRINTING, SO THAT    HEAD
                C     THE OUTPUT IS PROPERLY NUMBERED.  THE TITLE SUPPLIED ON THE       HEAD
                C     TITLE PARAMETER CARD AND THE CURRENT DATE ARE ALSO PRINTED        HEAD
                C     AS PART OF THE HEADING.  THIS INFORMATION IS EXTRACTED FROM       HEAD
                C     THE COMMON BLOCK /PRETTY/.                                        HEAD
                C                                                                      HEAD
                C     THERE IS A PROVISION FOR PRINTING, AS PART OF THE HEADING,        HEAD
                C     2 LINES OF 16 CHARACTERS EACH, POSSIBLY TO IDENTIFY THE           HEAD
                C     VERSION OR EDITION OF THE PROGRAM.  THESE ARE DEFINED WITHIN      HEAD
                C     THIS SUBROUTINE ONLY, AND MUST BE INITIALLIZED BY THE USER        HEAD
                C     IN DATA STATEMENTS TO PROVIDE THE DESIRED INFORMATION.            HEAD
                C                                                                      HEAD
                C     DEFINE THE COMMON BLOCKS OF INFORMATION                           HEAD
                C                                                                      HEAD
 ISN 0003            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),           HEAD
                    1     DATE,TIME,PAGE,LINE                                          HEAD
 ISN 0004            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                  HEAD
 ISN 0005            INTEGER*4  PAGE                                                   HEAD
                C                                                                      HEAD
 ISN 0006            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               HEAD
 ISN 0007            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               HEAD
                C                                                                      HEAD
                C     DEFINE THE HEADING FORMAT                                         HEAD
                C                                                                      HEAD
 ISN 0008        102 FORMAT ('1', T17, 2A8, T57, 'KANSAS STATE UNIVERSITY',            HEAD
                    1     T126, 'PAGE', I4/ ' ', T17, 2A8, T49,                        HEAD
                    2     'DEPT. OF STATISTICS & COMPUTER SCIENCE'/ ' ',               HEAD
                    3     T28, 10A8, T126, A8 // ' ')                                  HEAD
                C                                                                      HEAD
                C     DEFINE THE SPECIAL HEADING AREAS                                  HEAD
                C                                                                      HEAD
 ISN 0009            REAL*8   SPEC1(2),SPEC2(2)                                        HEAD
                C                                                                      HEAD
                C     INITIALIZE THE SPECIAL HEADING AS APPROPRIATE                     HEAD
                C                                                                      HEAD
 ISN 0010            DATA  SPEC1 /'KEITH ERIC MOE'/                                    HEAD
```

```
   ISN C011            DATA  SPEC2 /'MASTERS REPORT'/                                  HEAD
              C                                                                        HEAD
   ISN C012            PAGE = PAGE + 1                                                HEAD
   ISN C013            LINE = 5                                                       HEAD
   ISN C014            WRITE (LST,102) SPEC1,PAGE,SPEC2,TITLE,DATE                    HEAD
              C                                                                        HEAD
   ISN C015            RETURN                                                         HEAD
              C                                                                        HEAD
   ISN C016            END                                                            HEAD

*OPTICNS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      15 ,PROGRAM SIZE =      436

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            SUBROUTINE KRSKAL                                                    KRSKAL
                    C                                                                    KRSKAL
                    C     THE KRUSKAL-WALLIS TEST                                        KRSKAL
                    C                                                                    KRSKAL
                    C     THE KRUSKAL-WALLIS TEST IS ESSENTIALLY A K-SAMPLE MANN-WHITNEY KRSKAL
                    C     TEST.  IT IS USED ON 2 OR MORE RANDOM SAMPLES (NOT NECESSARILY KRSKAL
                    C     THE SAME SIZE) TO TEST FOR DIFFERENCES IN THEIR DISTRIBUTION   KRSKAL
                    C     FUNCTIONS.  FOR 2 SAMPLES, IT GIVES THE SAME RESULTS           KRSKAL
                    C     AS THE MANN-WHITNEY TEST.                                      KRSKAL
                    C                                                                    KRSKAL
                    C     FOR DETAILS ON THE KRUSKAL-WALLIS TEST, SEE CONOVER, W.J.,     KRSKAL
                    C     PRACTICAL NONPARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 404-415. KRSKAL
                    C                                                                    KRSKAL
                    C                                                                    KRSKAL
                    C     COMMON BLOCKS                                                  KRSKAL
                    C                                                                    KRSKAL
ISN C003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                         DECLARE
ISN C004            INTEGER*4  ROWS,COLS                                                 DECLARE
ISN C005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                             DECLARE
ISN C006            INTEGER*4  DISTN,WHTRN                                               DECLARE
ISN C007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                          DECLARE
ISN C008            REAL*8  BUFFER                                                       DECLARE
ISN C009            INTEGER*4  EOD,CDCOL,BUFNO                                           DECLARE
ISN C010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD                DECLARE
ISN C011            REAL*8   FMT                                                         DECLARE
ISN C012            INTEGER*4  DFORM,DOUT,RCBCD                                          DECLARE
ISN C013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),          DECLARE
                   1    VALUE(40),IVALUE(40),RC                                         DECLARE
ISN C014            REAL*8   PARM,SUBPRM                                                 DECLARE
ISN C015            INTEGER*4  RC                                                        DECLARE
ISN C016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                          DECLARE
ISN C017            INTEGER*4  TAIL                                                      DECLARE
ISN C018            LOGICAL*4  TESTS,DOALL                                               DECLARE
ISN C019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),              DECLARE
                   1    DATE,TIME,PAGE,LINE                                             DECLARE
ISN C020            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                     DECLARE
ISN C021            INTEGER*4  PAGE                                                      DECLARE
ISN C022            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)    DECLARE
ISN C023            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE        DECLARE
ISN C024            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                  DECLARE
ISN C025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                  DECLARE
```

```
ISN C026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                    DECLARE
ISN C027            INTEGER*4  ITEMS,LOCAT,RDNO                                DECLARE
ISN C028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                DECLARE
ISN C029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                     DECLARE
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                              DECLARE
ISN C031            REAL*8  BLANK /' '/                                        DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                 DECLARE
                C                                                              KRSKAL
                C     OTHER DATA USED IN THIS PROGRAM                          KRSKAL
                C                                                              KRSKAL
ISN C033            REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                 KRSKAL
ISN C034            INTEGER*4  CURSMP,CURVAR,CURCNT,TIES,PLCS                  KRSKAL
ISN C035            REAL*4  RECORD(6)                                          KRSKAL
ISN C036            EQUIVALENCE (RECORD(1),CURSMP), (RECORD(2),CURVAR),        KRSKAL
                   1    (RECORD(3),CURCNT), (RECORD(6),OBS)                    KRSKAL
                C                                                              KRSKAL
                C     FORMAT STATEMENTS                                        KRSKAL
                C                                                              KRSKAL
ISN C037          234 FORMAT ('-', T12, 'IKMS34I  INVALID SAMPLE SIZE(S)')     KRSKAL
ISN C038          290 FORMAT ('-', T12, 'IKMS90I  ***** BECAUSE OF SAMPLE SIZES, ALPHA HKRSKAL
                  1AT MAY NOT BE CORRECT.  VERIFY WITH TABLES. *****')         KRSKAL
ISN C039          301 FORMAT ('-', T60, 'THE KRUSKAL-WALLIS TEST')             KRSKAL
ISN C040          302 FORMAT ('0', T12, 'HO: ALL', I3, ' SAMPLES HAVE THE SAME DISTRIBUTKRSKAL
                  1ION'/ '0', T12, 'VERSUS'/ '0', T12,                         KRSKAL
                  2    'H1: AT LEAST 1 SAMPLE YIELDS LARGER OBSERVATIONS THAN THE OTHKRSKAL
                  3ERS')                                                       KRSKAL
ISN C041          303 FORMAT ('-', T12, 'THERE ARE', I3, ' SAMPLES WITH A TOTAL OF',  KRSKAL
                  1    I6, ' OBSERVATIONS')                                    KRSKAL
ISN C042          304 FORMAT ('0', T12, 'SAMPLE NUMBER', T32, 'SIZE', T42, 'RANK SUM'/  KRSKAL
                  1    (' ', T12, A8, I4, T30, I5, T42, G10.4) )               KRSKAL
ISN C043          305 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5, ' WITH', I3,  KRSKAL
                  1    ' DEGREES OF FREEDOM')                                  KRSKAL
ISN C044          306 FORMAT ('-', T12, 'THERE WERE', I5, ' TIED OBSERVATIONS IN',  KRSKAL
                  1    I3, ' PLACES')                                          KRSKAL
ISN C045          307 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')  KRSKAL
ISN C046          308 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                   KRSKAL
ISN C047          309 FORMAT ('-', T53, 'END OF THE KRUSKAL-WALLIS TEST')      KRSKAL
                C                                                              KRSKAL
                C     START WITH TITLE AND HYPOTHESES                          KRSKAL
                C                                                              KRSKAL
ISN C048            WKSZE = WKSZE                                              KRSKAL
ISN C049            WRITE (LST,301)                                            KRSKAL
ISN C050            WRITE (LST,302) SMPLS                                      KRSKAL
```

```
ISN C051            CALL OFLOW(9)                                                  KRSKAL
                C                                                                  KRSKAL
                C   CHECK FOR PREVIOUSLY SORTED                                    KRSKAL
                C                                                                  KRSKAL
ISN C052            IF (SUBS(1) .NE. 0) GO TO 31                                   KRSKAL
ISN C054            IF (WHTRN .EQ. 1 .OR. (OUT .NE. 0 .AND. DOUT .EQ. 2) ) GO TO 21 KRSKAL
                C                                                                  KRSKAL
                C   SORT THE DATA - NO TRANSFORMATION OR DATA OUTPUT NEEDED        KRSKAL
                C                                                                  KRSKAL
ISN C056            CALL RANKER (UT4,UT1,LT2,UT3,J)                                KRSKAL
ISN C057            IF (RC .NE. 0) GO TO 92                                        KRSKAL
ISN C059            SUBS(1) = -1                                                   KRSKAL
ISN C060            SUBS(2) = UT4                                                  KRSKAL
ISN C061            GO TO 31                                                       KRSKAL
                C                                                                  KRSKAL
                C   DATA MUST BE TRANSFORMED AND/OR WRITTEN OUT                    KRSKAL
                C                                                                  KRSKAL
ISN C062         21 REWIND UT2                                                     KRSKAL
ISN C063            DO 24 ISMP = 1, SMPLS                                          KRSKAL
ISN C064            REWIND UT1                                                     KRSKAL
ISN C065         22 READ (UT1,END=24) RECORD                                       KRSKAL
ISN C066            IF (ISMP .NE. CURSMP) GO TO 22                                 KRSKAL
ISN C068            ITEMS = 1                                                      KRSKAL
ISN C069            LOCAT(4) = 0                                                   KRSKAL
ISN C070            X(1) = OBS                                                     KRSKAL
ISN C071            IF (WHTRN .EQ. 1) CALL TRANS                                   KRSKAL
ISN C073            IF (ITEMS .LT. 1) GO TO 22                                     KRSKAL
ISN C075            IF (OUT .NE. 0 .AND. DOUT .EQ. 2) CALL WRDATA                  KRSKAL
                C                                                                  KRSKAL
ISN C077            DO 23 I = 1, ITEMS                                             KRSKAL
ISN C078            OBS = X(I)                                                     KRSKAL
ISN C079         23 WRITE (UT2) RECORD                                             KRSKAL
ISN C080            GO TO 22                                                       KRSKAL
                C                                                                  KRSKAL
ISN C081         24 CONTINUE                                                       KRSKAL
                C                                                                  KRSKAL
                C   SORT THE DATA FOR RANKING                                     KRSKAL
                C                                                                  KRSKAL
ISN C082            CALL RANKER (UT4,UT2,LT3,UT4,J)                                KRSKAL
ISN C083            IF (RC .NE. 0) GO TO 92                                        KRSKAL
ISN C085            SUBS(1) = -1                                                   KRSKAL
ISN C086            SUBS(2) = UT4                                                  KRSKAL
ISN C087            IF (WHTRN .EQ. 1) SUBS(1) = 1                                  KRSKAL
```

```
                C                                                                   KRSKAL
                C     DATA HAS BEEN SORTED AT THIS POINT                            KRSKAL
                C                                                                   KRSKAL
ISN C088           31 IF (OUT .EQ. 0 .OR. OCUT .NE. 3) GO TO 36                     KRSKAL
                C                                                                   KRSKAL
                C     DATA IS TO BE WRITTEN OUT AS SORTED                           KRSKAL
                C                                                                   KRSKAL
ISN C091              IUT1 = SUBS(2)                                                KRSKAL
ISN C092              ITEMS = 1                                                     KRSKAL
ISN C093              LOCAT(4) = 0                                                  KRSKAL
                C                                                                   KRSKAL
ISN C094              DO 33 ISMP = 1, SMPLS                                         KRSKAL
ISN C095              REWIND IUT1                                                   KRSKAL
                C                                                                   KRSKAL
ISN C096           32 READ (IUT1,END=33) RECORD                                     KRSKAL
                C                                                                   KRSKAL
ISN C097              IF (ISMP .NE. CURSMP) GO TO 32                                KRSKAL
ISN C099              X(1) = OBS                                                    KRSKAL
ISN C100              CALL WRDATA                                                   KRSKAL
ISN C101              GO TO 32                                                      KRSKAL
                C                                                                   KRSKAL
ISN C102           33 CONTINUE                                                      KRSKAL
                C                                                                   KRSKAL
                C     NOW RANK THE SORTED DATA                                      KRSKAL
                C                                                                   KRSKAL
ISN C103           36 IF (OUT .NE. C .AND. COUT .GE. 2 .AND. OUT .NE. LST) ENDFILE OUT KRSKAL
ISN C105              OUT = 0                                                       KRSKAL
ISN C106              RANK = 0.                                                     KRSKAL
ISN C107              DO 37 I = 1, SMPLS                                            KRSKAL
ISN C108              DATA(I) = 0.                                                  KRSKAL
ISN C109           37 SSIZE(I) = 0                                                  KRSKAL
ISN C110              TIES = 0                                                      KRSKAL
ISN C111              PLCS = 0                                                      KRSKAL
ISN C112              IUT1 = SUBS(2)                                                KRSKAL
ISN C113              REWIND IUT1                                                   KRSKAL
                C                                                                   KRSKAL
ISN 0114              READ (IUT1,END=1234) RECORD                                   KRSKAL
ISN C115              SSIZE(CURSMP) = SSIZE(CURSMP) + 1                             KRSKAL
                C                                                                   KRSKAL
                C     COUNT IT AND SAVE IT                                          KRSKAL
                C                                                                   KRSKAL
ISN 0116           41 XVAL = OBS                                                    KRSKAL
ISN C117              ISMP = CURSMP                                                 KRSKAL
```

```
ISN C118            RANK = RANK + 1.                                    KRSKAL
                C                                                       KRSKAL
ISN C119            READ (IUT1,END=42) RECORD                           KRSKAL
ISN C120            SSIZE(CURSMP) = SSIZE(CURSMP) + 1                   KRSKAL
ISN C121            IF (XVAL .EQ. OBS) GO TO 46                         KRSKAL
                C                                                       KRSKAL
                C     NO TIE HERE - RANK THE OBSERVATION                KRSKAL
                C                                                       KRSKAL
ISN C123            DATA(ISMP) = DATA(ISMP) + RANK                      KRSKAL
ISN C124            GO TO 41                                            KRSKAL
                C                                                       KRSKAL
                C     END OF FILE - PROCESS LAST OBSERVATION            KRSKAL
                C                                                       KRSKAL
ISN C125         42 DATA(ISMP) = DATA(ISMP) + RANK                      KRSKAL
ISN C126            GO TO 61                                            KRSKAL
                C                                                       KRSKAL
                C     PROCESS TIED OBSERVATIONS                         KRSKAL
                C                                                       KRSKAL
ISN C127         46 PLCS = PLCS + 1                                     KRSKAL
ISN C128            I = 2                                               KRSKAL
ISN C129            DO 47 J = 1, SMPLS                                  KRSKAL
ISN C130         47 IWRK(J) = 0                                         KRSKAL
ISN C131            IWRK(ISMP) = 1                                      KRSKAL
                C                                                       KRSKAL
ISN C132         48 IWRK(CURSMP) = IWRK(CLRSMP) + 1                     KRSKAL
ISN C133            READ (IUT1,END=55) RECORD                           KRSKAL
ISN C134            SSIZE(CURSMP) = SSIZE(CURSMP) + 1                   KRSKAL
                C                                                       KRSKAL
ISN C135            IF (OBS .NE. XVAL) GO TO 49                         KRSKAL
ISN C137            I = I + 1                                           KRSKAL
ISN C138            GO TO 48                                            KRSKAL
                C                                                       KRSKAL
                C     END OF TIE COUNTING                               KRSKAL
                C                                                       KRSKAL
ISN C139         49 Z = 0.                                              KRSKAL
ISN C140            DO 50 L = 1, I                                      KRSKAL
ISN C141            Z = Z + RANK                                        KRSKAL
ISN C142         50 RANK = RANK + 1.                                    KRSKAL
                C                                                       KRSKAL
ISN C143            TIES = TIES + I                                     KRSKAL
ISN C144            Z = Z / I                                           KRSKAL
                C                                                       KRSKAL
                C     CALCULATE CONTRIBUTION TO EACH SAMPLE             KRSKAL
```

```
                C                                                                   KRSKAL
ISN C145            DO 51 J = 1, SMPLS                                              KRSKAL
ISN C146         51 DATA(J) = DATA(J) + IWRK(J) * Z                                 KRSKAL
                C                                                                   KRSKAL
ISN C147            RANK = RANK - 1.                                                KRSKAL
ISN C148            GO TO 41                                                        KRSKAL
                C                                                                   KRSKAL
                C     END OF FILE DURING A TIE                                      KRSKAL
                C                                                                   KRSKAL
ISN C149         55 Z = 0.                                                          KRSKAL
ISN C150            DO 56 L = 1, I                                                  KRSKAL
ISN C151            Z = Z + RANK                                                    KRSKAL
ISN C152         56 RANK = RANK + 1.                                                KRSKAL
ISN C153            TIES = TIES + I                                                 KRSKAL
ISN C154            Z = Z / I                                                       KRSKAL
                C                                                                   KRSKAL
ISN C155            DO 57 J = 1, SMPLS                                              KRSKAL
ISN C156         57 DATA(J) = DATA(J) + IWRK(J) * Z                                 KRSKAL
                C                                                                   KRSKAL
                C     END OF RANK SUMMING - CALCULATE THE TEST STATISTIC            KRSKAL
                C                                                                   KRSKAL
ISN C157         61 T = 0.                                                          KRSKAL
ISN C158            ISZ = 0.                                                        KRSKAL
ISN C159            DO 62 I = 1, SMPLS                                              KRSKAL
ISN C160            ISZ = ISZ + SSIZE(I)                                            KRSKAL
ISN C161         62 T = T + DATA(I) * DATA(I) / SSIZE(I)                            KRSKAL
                C                                                                   KRSKAL
ISN C162            T = 12. * T / (ISZ * (ISZ + 1) ) - 3. * (ISZ + 1)               KRSKAL
                C                                                                   KRSKAL
                C     LIST THE SAMPLE BREAKDOWNS                                    KRSKAL
                C                                                                   KRSKAL
ISN C163            WRITE (LST,303) SMPLS,ISZ                                       KRSKAL
ISN C164            WRITE (LST,304) (LABEL(I),I,SSIZE(I),DATA(I), I = 1, SMPLS)     KRSKAL
ISN C165            J = 5 + SMPLS                                                   KRSKAL
ISN C166            CALL OFLOW(J)                                                   KRSKAL
                C                                                                   KRSKAL
ISN C167            IF (TIES .EQ. 0) GO TO 63                                       KRSKAL
ISN C169            WRITE (LST,306) TIES,PLOS                                       KRSKAL
ISN C170            CALL OFLOW(3)                                                   KRSKAL
                C                                                                   KRSKAL
ISN C171         63 IDF = SMPLS - 1                                                 KRSKAL
ISN C172            WRITE (LST,305) T,IDF                                           KRSKAL
                C                                                                   KRSKAL
```

```
                C     CALCULATE THE CRITICAL REGION                                  KRSKAL
                C                                                                    KRSKAL
   ISN C173           DF = IDF                                                       KRSKAL
   ISN C174           ALFHAT = CDTR(T,DF)                                            KRSKAL
                C                                                                    KRSKAL
   ISN C175           I = 1                                                          KRSKAL
   ISN C176           IF (ALPHA .GT. ALFHAT) I = 2                                   KRSKAL
   ISN C178           WRITE (LST,307) ALPHA, ACCREJ(I)                               KRSKAL
   ISN 0179           WRITE (LST,308) ALFHAT                                         KRSKAL
   ISN C180           CALL OFLOW(7)                                                  KRSKAL
                C                                                                    KRSKAL
                C     CHECK FOR SMALL SAMPLE SIZES                                   KRSKAL
                C                                                                    KRSKAL
   ISN C181           IF (SMPLS .GT. 3 .OR. ISZ .GT. 15) GO TO 92                    KRSKAL
   ISN C183           WRITE (LST,290)                                                KRSKAL
   ISN C184           CALL OFLOW(3)                                                  KRSKAL
                C                                                                    KRSKAL
                C     END OF THE KRUSKAL-WALLIS TEST                                 KRSKAL
                C                                                                    KRSKAL
   ISN C185        92 WRITE (LST,309)                                                KRSKAL
   ISN C186           CALL OFLOW(3)                                                  KRSKAL
   ISN C187           RETURN                                                         KRSKAL
                C                                                                    KRSKAL
                C     ERROR - DATA BAD                                               KRSKAL
                C                                                                    KRSKAL
   ISN C188      1234  WRITE (LST,234)                                              KRSKAL
   ISN C189           CALL OFLOW(3)                                                  KRSKAL
   ISN C190           RC = 12                                                        KRSKAL
   ISN C191           GO TO 92                                                       KRSKAL
                C                                                                    KRSKAL
   ISN C192           END                                                            KRSKAL

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     191 ,PROGRAM SIZE =     3682

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE MANWHT                                                MANWHT
              C                                                                     MANWHT
              C    THE MANN-WHITNEY TEST                                            MANWHT
              C                                                                     MANWHT
              C    THE MANN-WHITNEY TEST IS A TEST OF 2 RANDOM SAMPLES (NOT         MANWHT
              C    NECESSARILY THE SAME SIZE) TO DETERMINE IF THE EXPECTED          MANWHT
              C    VALUE OF THE FIRST SAMPLE IS LESS THAN, GREATER THAN, OR         MANWHT
              C    EQUAL TO THE EXPECTED VALUE OF THE SECOND SAMPLE.                MANWHT
              C                                                                     MANWHT
              C    THE CRITICAL REGION FOR THIS TEST IS DETERMINED FROM THE         MANWHT
              C    LARGE SAMPLE APPROXIMATION.  FOR EITHER SAMPLE BEING LESS        MANWHT
              C    THAN 20 OBSERVATIONS, THE CRITICAL REGION GIVEN WILL BE          MANWHT
              C    INCORRECT AND EXACT TABLES SHOULD BE CONSULTED TO                MANWHT
              C    DETERMINE THE ACTUAL CRITICAL REGION.                            MANWHT
              C                                                                     MANWHT
              C    FOR DETAILS ON THE MANN-WHITNEY TEST, SEE CONOVER, W.J.,         MANWHT
              C    PRACTICAL NON-PARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 353-369. MANWHT
              C                                                                     MANWHT
              C                                                                     MANWHT
              C    COMMON BLOCKS                                                    MANWHT
              C                                                                     MANWHT
 ISN 0003          COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                     DECLARE
 ISN 0004          INTEGER*4  ROWS,COLS                                             DECLARE
 ISN 0005          COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                         DECLARE
 ISN 0006          INTEGER*4  DISTN,WHTRN                                           DECLARE
 ISN 0007          COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                      DECLARE
 ISN 0008          REAL*8  BUFFER                                                   DECLARE
 ISN 0009          INTEGER*4  EOD,CDCOL,BUFNO                                       DECLARE
 ISN 0010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD            DECLARE
 ISN 0011          REAL*8   FMT                                                     DECLARE
 ISN 0012          INTEGER*4  DFORM,DOUT,RCBCD                                      DECLARE
 ISN 0013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),      DECLARE
                  1    VALUE(40),IVALUE(40),RC                                      DECLARE
 ISN 0014          REAL*8   PARM,SUBPRM                                             DECLARE
 ISN 0015          INTEGER*4  RC                                                    DECLARE
 ISN 0016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                      DECLARE
 ISN 0017          INTEGER*4  TAIL                                                  DECLARE
 ISN 0018          LOGICAL*4  TESTS,DOALL                                           DECLARE
 ISN 0019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),          DECLARE
                  1    DATE,TIME,PAGE,LINE                                          DECLARE
 ISN 0020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                 DECLARE
```

```
ISN C021          INTEGER*4  PAGE                                                    DECLARE
ISN C022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
ISN C023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE     DECLARE
ISN C024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C025          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C026          COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                          DECLARE
ISN C027          INTEGER*4  ITEMS,LOCAT,RDNO                                       DECLARE
ISN C028          COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                      DECLARE
ISN C029          INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                            DECLARE
ISN C030          EQUIVALENCE (WORK(1),IWRK(1))                                     DECLARE
ISN C031          REAL*8  BLANK /' '/                                               DECLARE
ISN C032          EQUIVALENCE (BLANK,IBLANK)                                        DECLARE
                  C                                                                 MANWHT
                  C     DATA AND CONSTANTS USED                                     MANWHT
                  C                                                                 MANWHT
ISN C033          REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                        MANWHT
ISN C034          REAL*8  XLBL /'X'/, YLBL /'Y'/                                    MANWHT
ISN C035          INTEGER*2  H0 /'H0'/, H1 /'H1'/                                   MANWHT
ISN C036          INTEGER*4  HYPLBL(2,3) /'.EQ.', '.NE.', '.LE.', '.GT.',           MANWHT
                 1    '.GE.', '.LT.'/                                             MANWHT
ISN C037          REAL*4  RECORD(6)                                                 MANWHT
ISN C038          INTEGER*4  CURSMP,CURVAR,CURCNT,TIES,PLCS                         MANWHT
ISN C039          EQUIVALENCE (RECORD(1),CURSMP), (RECORD(2),CURVAR),               MANWHT
                 1    (RECORD(3),CURCNT), (RECORD(6),OBS)                         MANWHT
ISN C040          REAL*8  T1, T2, T3                                                MANWHT
                  C                                                                 MANWHT
                  C     FORMAT STATEMENTS                                           MANWHT
                  C                                                                 MANWHT
ISN C041        234 FORMAT ('-', T12, 'IKMS34I  INVALID SAMPLE SIZE(S)')            MANWHT
ISN C042        29C FORMAT ('-', T12, 'IKMS90I  ***** BECAUSE OF SAMPLE SIZES, ALPHA HMANWHT
                   1AT MAY NOT BE CORRECT.  VERIFY WITH TABLES. *****')             MANWHT
ISN C043        3C1 FORMAT ('-', T57, 'THE MANN-WHITNEY TEST')                      MANWHT
ISN C044        3C2 FORMAT ('0', T12, A2, ': E(', A8, ') ', A4, ' E(', A8, ')')     MANWHT
ISN C045        3C3 FORMAT ('0', T12, 'VERSUS')                                     MANWHT
ISN C046        3C4 FORMAT ('0', (T12, A8, ' HAS', I6, ' OBSERVATIONS'/) )          MANWHT
ISN C047        3C5 FORMAT ('0', T12, 'THERE WAS A TOTAL OF', I5,                   MANWHT
                   1    ' TIED OBSERVATIONS IN', I3, ' PLACES')                     MANWHT
ISN C048        3C6 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5)              MANWHT
ISN C049        3C7 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')      MANWHT
ISN C050        3C8 FORMAT ('C', T12, 'ALPHA HAT =', F6.3)                          MANWHT
ISN C051        3C9 FORMAT ('-', T54, 'END OF THE MANN-WHITNEY TEST')               MANWHT
                  C                                                                 MANWHT
                  C     START BY FIXING THE LABELS AND PRINTING THE HYPOTHESES      MANWHT
```

```
                C                                                                   MANWHT
ISN C052            WKSZE = WKSZE                                                   MANWHT
ISN C053            WRITE (LST,301)                                                 MANWHT
ISN C054            IF (LABEL(1) .EQ. BLANK) LABEL(1) = XLBL                        MANWHT
ISN C056            IF (LABEL(2) .EQ. BLANK) LABEL(2) = YLBL                        MANWHT
                C                                                                   MANWHT
ISN C058            ITL = 1                                                         MANWHT
ISN C059            IF (TAIL) 11, 13, 12                                            MANWHT
ISN C060         11 ITL = 2                                                         MANWHT
ISN C061            GO TO 13                                                        MANWHT
ISN C062         12 ITL = 3                                                         MANWHT
                C                                                                   MANWHT
ISN C063         13 WRITE (LST,302) HO, LABEL(1), HYPLBL(1,ITL), LABEL(2)           MANWHT
ISN C064            WRITE (LST,303)                                                 MANWHT
ISN C065            WRITE (LST,302) H1, LABEL(1), HYPLBL(2,ITL), LABEL(2)           MANWHT
ISN C066            CALL OFLOW(9)                                                   MANWHT
                C                                                                   MANWHT
ISN C067            IF (SUBS(1) .NE. 0) GO TO 31                                    MANWHT
                C                                                                   MANWHT
                C     DATA HAS NOT BEEN PREVIOUSLY SORTED                           MANWHT
                C                                                                   MANWHT
ISN C069            IF (WHTRN .EQ. 1 .OR. (OUT .NE. 0 .AND. DOUT .EQ. 2)) GO TO 21  MANWHT
                C                                                                   MANWHT
                C     SORT THE DATA - NO TRANSFORMATION OR USER OUTPUT NEEDED       MANWHT
                C                                                                   MANWHT
ISN C071            CALL RANKER(UT4,UT1,UT2,UT3,J)                                  MANWHT
ISN C072            IF (RC .NE. 0) GO TO 52                                         MANWHT
ISN C074            SUBS(1) = -1                                                    MANWHT
ISN C075            SUBS(2) = UT4                                                   MANWHT
ISN C076            GO TO 31                                                        MANWHT
                C                                                                   MANWHT
                C     DATA HAS TO BE TRANSFORMED AND/OR WRITTEN OUT                 MANWHT
                C                                                                   MANWHT
ISN C077         21 REWIND UT2                                                      MANWHT
ISN C078            DO 24 ISMP = 1, SMPLS                                           MANWHT
ISN C079            REWIND UT1                                                      MANWHT
                C                                                                   MANWHT
ISN C080         22 READ (UT1,END=24) RECORD                                        MANWHT
ISN C081            IF (ISMP .NE. CURSMP) GO TO 22                                  MANWHT
ISN C083            ITEMS = 1                                                       MANWHT
ISN C084            LOCAT(4) = 0                                                    MANWHT
ISN C085            X(1) = OBS                                                      MANWHT
ISN C086            IF (WHTRN .EQ. 1) CALL TRANS                                    MANWHT
```

```
ISN C088            IF (ITEMS .LT. 1) GO TO 22                                  MANWHT
ISN C090            IF (OUT .NE. 0 .AND. COUT .EQ. 2) CALL WRDATA               MANWHT
                C                                                               MANWHT
ISN C092            DO 23 I = 1, ITEMS                                          MANWHT
ISN C093            OBS = X(I)                                                  MANWHT
ISN C094         23 WRITE (UT2) RECORD                                          MANWHT
ISN C095            GO TO 22                                                    MANWHT
                C                                                               MANWHT
ISN C096         24 CONTINUE                                                    MANWHT
                C                                                               MANWHT
ISN C097            ENDFILE UT2                                                 MANWHT
                C                                                               MANWHT
                C     SORT THE DATA DOWN                                        MANWHT
                C                                                               MANWHT
ISN C098            CALL RANKER(UT4,UT2,UT3,UT4,J)                              MANWHT
ISN C099            IF (RC .NE. 0) GO TO 92                                     MANWHT
ISN C101            SUBS(2) = UT4                                               MANWHT
ISN C102            SUBS(1) = -1                                                MANWHT
ISN C103            IF (WHTRN .EQ. 1) SUBS(1) = 1                               MANWHT
                C                                                               MANWHT
                C     DATA IS SORTED AND IS ON UNIT SUBS(2)                     MANWHT
                C                                                               MANWHT
ISN C105         31 IF (OUT .EQ. 0 .OR. OCUT .NE. 3) GO TO 36                   MANWHT
                C                                                               MANWHT
                C     DATA MUST BE WRITTEN OUT IN SORTED ORDER                  MANWHT
                C                                                               MANWHT
ISN C107            IUT1 = SUBS(2)                                              MANWHT
ISN C108            ITEMS = 1                                                   MANWHT
ISN C109            LOCAT(4) = 0                                                MANWHT
                C                                                               MANWHT
ISN C110            DO 33 ISMP = 1, SMPLS                                       MANWHT
ISN C111            REWIND IUT1                                                 MANWHT
                C                                                               MANWHT
ISN C112         32 READ (IUT1,END=33) RECORD                                   MANWHT
ISN C113            IF (ISMP .NE. CURSMP) GO TO 32                              MANWHT
ISN C115            X(1) = OBS                                                  MANWHT
ISN C116            CALL WRDATA                                                 MANWHT
ISN C117            GO TO 32                                                    MANWHT
                C                                                               MANWHT
ISN C118         33 CONTINUE                                                    MANWHT
                C                                                               MANWHT
                C     NOW PROCESS SORTED DATA TO OBTAIN TEST STATISTIC          MANWHT
                C                                                               MANWHT
```

```
ISN C119          36 IF (OUT .NE. 0 .AND. COUT .GE. 2 .ANC. OUT .NE. LST) ENDFILE OUT  MANWHT
ISN C121             OUT = 0                                                           MANWHT
                C                                                                      MANWHT
ISN C122             RANK = 0.                                                         MANWHT
ISN C123             T = 0.                                                            MANWHT
ISN C124             TIES = 0                                                          MANWHT
ISN C125             PLCS = 0                                                          MANWHT
ISN C126             SSIZE(1) = 0                                                      MANWHT
ISN C127             SSIZE(2) = 0                                                      MANWHT
ISN C128             IUT1 = SUBS(2)                                                    MANWHT
ISN C129             REWIND IUT1                                                       MANWHT
ISN C130             READ (IUT1,END=99) RECORD                                         MANWHT
ISN C131             SSIZE(CURSMP) = SSIZE(CURSMP) + 1                                 MANWHT
                C                                                                      MANWHT
                C    COUNT THE OBSERVATION - SAVE THE VALUE                            MANWHT
                C                                                                      MANWHT
ISN C132          41 XVAL = OBS                                                        MANWHT
ISN C133             ISMP = CURSMP                                                     MANWHT
ISN C134             RANK = RANK + 1.                                                  MANWHT
                C                                                                      MANWHT
                C    READ THE NEXT OBSERVATION - CHECK FOR A TIE                       MANWHT
                C                                                                      MANWHT
ISN C135             READ (IUT1,END=42) RECORD                                         MANWHT
ISN C136             SSIZE(CURSMP) = SSIZE(CURSMP) + 1                                 MANWHT
ISN C137             IF (XVAL .EQ. OBS) GO TO 51                                       MANWHT
                C                                                                      MANWHT
                C    COUNT A NON-TIED OBSERVATION                                      MANWHT
                C                                                                      MANWHT
ISN C139             IF (ISMP .EQ. 1) T = T + RANK                                     MANWHT
ISN C141             GO TO 41                                                          MANWHT
                C                                                                      MANWHT
                C    END OF FILE                                                       MANWHT
                C                                                                      MANWHT
ISN C142          42 IF (ISMP .EQ. 1) T = T + RANK                                     MANWHT
ISN C144             GO TO 61                                                          MANWHT
                C                                                                      MANWHT
                C    COUNT TIED OBSERVATIONS                                           MANWHT
                C                                                                      MANWHT
ISN C145          51 PLCS = PLCS + 1                                                   MANWHT
ISN C146             I = 2                                                             MANWHT
ISN C147             K = 0                                                             MANWHT
ISN C148             IF (ISMP .EQ. 1) K = 1                                            MANWHT
ISN 0150          52 IF (CURSMP .EQ. 1) K = K + 1                                      MANWHT
```

```
ISN C152            READ (IUT1,END=55) RECORD                                   MANWHT
ISN C153            SSIZE(CURSMP) = SSIZE(CURSMP) + 1                           MANWHT
ISN C154            IF (OBS .NE. XVAL) GO TO 53                                 MANWHT
ISN C156            I = I + 1                                                   MANWHT
ISN C157            GO TO 52                                                    MANWHT
                C                                                               MANWHT
                C     END OF SERIES OF TIED OBSERVATIONS                        MANWHT
                C                                                               MANWHT
ISN C158         53 Z = 0.                                                      MANWHT
ISN C159            DO 54 L = 1, I                                              MANWHT
ISN C160            Z = Z + RANK                                                MANWHT
ISN C161         54 RANK = RANK + 1.                                            MANWHT
                C                                                               MANWHT
ISN C162            TIES = TIES + I                                             MANWHT
ISN C163            Z = Z / I                                                   MANWHT
ISN C164            T = T + K * Z                                               MANWHT
ISN C165            RANK = RANK - 1.                                            MANWHT
ISN C166            GO TO 41                                                    MANWHT
                C                                                               MANWHT
                C     END OF FILE WHILE SEARCHING FOR MORE TIES                 MANWHT
                C                                                               MANWHT
ISN C167         55 Z = 0.                                                      MANWHT
ISN C168            DO 56 L = 1, I                                              MANWHT
ISN C169            Z = Z + RANK                                                MANWHT
ISN C170         56 RANK = RANK + 1.                                            MANWHT
ISN C171            TIES = TIES + I                                             MANWHT
ISN C172            Z = Z / I                                                   MANWHT
ISN C173            T = T + K * Z                                               MANWHT
                C                                                               MANWHT
                C     TEST STATISTIC IS CALCULATED - ADJUST IT AND PRINT        MANWHT
                C                                                               MANWHT
ISN C174         61 T = T - SSIZE(1) * (SSIZE(1) + 1) / 2.                      MANWHT
ISN C175            WRITE (LST,304) LABEL(1),SSIZE(1),LABEL(2),SSIZE(2)         MANWHT
ISN C176            CALL OFLOW(3)                                               MANWHT
ISN C177            IF (SSIZE(1) .EQ. 0 .OR. SSIZE(2) .EQ. 0) GO TO 99          MANWHT
                C                                                               MANWHT
ISN C179            IF (TIES .EQ. 0) GO TO 62                                   MANWHT
ISN C181            WRITE (LST,305) TIES,PLCS                                   MANWHT
ISN C182            CALL OFLOW(2)                                               MANWHT
                C                                                               MANWHT
ISN C183         62 WRITE (LST,306) T                                           MANWHT
                C                                                               MANWHT
                C     CALCULATE THE CRITICAL REGION                             MANWHT
```

```
                C                                                                   MANWHT
ISN 0184              T1 = SSIZE(1) * SSIZE(2)                                      MANWHT
ISN 0185              T1 = T1 / 2.                                                  MANWHT
ISN 0186              T2 = T1 * (SSIZE(1) + SSIZE(2) + 1) / 6.                      MANWHT
ISN 0187              T2 = DSQRT(T2)                                                MANWHT
ISN 0188              T3 = T                                                        MANWHT
ISN 0189              XP = (T3 - T1) / T2                                           MANWHT
ISN 0190              ALFHAT = ANDTR(XP)                                            MANWHT
                C                                                                   MANWHT
ISN 0191              GO TO (71, 76, 81), ITL                                       MANWHT
                C                                                                   MANWHT
                C     2-TAILED TEST                                                 MANWHT
                C                                                                   MANWHT
ISN 0192           71 ALFHAT = AMIN1(ALFHAT, 1.-ALFHAT) * 2.                        MANWHT
ISN 0193              IF (ALFHAT .GT. 1) ALFHAT = 1.                                MANWHT
ISN 0195              GO TO 91                                                      MANWHT
                C                                                                   MANWHT
                C     LOWER TAIL TEST                                               MANWHT
                C                                                                   MANWHT
ISN 0196           76 GO TO 91                                                      MANWHT
                C                                                                   MANWHT
                C     UPPER TAIL                                                    MANWHT
                C                                                                   MANWHT
ISN 0197           81 ALFHAT = 1. - ALFHAT                                          MANWHT
                C                                                                   MANWHT
                C     ACCEPT OR REJECT ACCORDINGLY                                  MANWHT
                C                                                                   MANWHT
ISN 0198           91 I = 1                                                         MANWHT
ISN 0199              IF (ALPHA .GT. ALFHAT) I = 2                                  MANWHT
ISN 0201              WRITE (LST,307) ALPHA, ACCREJ(I)                              MANWHT
ISN 0202              WRITE (LST,308) ALFHAT                                        MANWHT
ISN 0203              CALL OFLOW(5)                                                 MANWHT
                C                                                                   MANWHT
                C     CHECK FOR SMALL SAMPLE SIZES                                  MANWHT
                C                                                                   MANWHT
ISN 0204              IF (SSIZE(1) .GT. 20 .AND. SSIZE(2) .GT. 20) GO TO 92         MANWHT
ISN 0206              WRITE (LST,290)                                               MANWHT
ISN 0207              CALL OFLOW(3)                                                 MANWHT
                C                                                                   MANWHT
                C     END OF THE MANN-WHITNEY TEST                                  MANWHT
                C                                                                   MANWHT
ISN 0208           92 WRITE (LST,309)                                               MANWHT
ISN 0209              CALL OFLOW(3)                                                 MANWHT
```

```
   ISN C210            RETURN                                                  MANWHT
               C                                                               MANWHT
               C       ERROR - BAD SAMPLE SIZE                                 MANWHT
               C                                                               MANWHT
   ISN C211         99 WRITE (LST,234)                                         MANWHT
   ISN C212            CALL OFLOW(3)                                           MANWHT
   ISN C213            RC = 12                                                 MANWHT
   ISN C214            GO TO 92                                                MANWHT
               C                                                               MANWHT
   ISN C215            END                                                     MANWHT

*OPTICNS IN EFFECT*       NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*      SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NCMAP,NOEDIT,ID,NOXREF

*STATISTICS*      SOURCE STATEMENTS =      214 ,PROGRAM SIZE =      3586

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            SUBROUTINE MCNMAR                                                    MCNMAR
                    C                                                                    MCNMAR
                    C     THE MC NEMAR TEST FOR SIGNIFICANCE OF CHANGES                  MCNMAR
                    C                                                                    MCNMAR
                    C     THE MC NEMAR TEST IS USED TO CHECK 2 MATCHED SAMPLES OR A      MCNMAR
                    C     BIVARIATE SAMPLE FOR CHANGE WHERE IT IS ASSUMED THAT THE FIRST MCNMAR
                    C     SAMPLE OR VARIATE REPRESENTS A "BEFORE" SITUATION AND THE      MCNMAR
                    C     SECOND REPRESENTS THE "AFTER" SITUATION.  EACH OBSERVATION     MCNMAR
                    C     FROM EACH SAMPLE (OR VARIATE) MUST BE DICHOTOMIZED INTO A      MCNMAR
                    C     ZERO OR A ONE.  THIS CAN EITHER BE DONE BY THE USER IN THE     MCNMAR
                    C     TRANS SUBROUTINE, OR BY SPECIFYING A VALUE TO SPLIT AT ON      MCNMAR
                    C     THE DISTRIB CARD.  IF THIS LEVEL IS NOT SPECIFIED, THE VALUE   MCNMAR
                    C     OF .5 IS ASSUMED, THEREBY MAKING THE ASSUMPTION THAT THE       MCNMAR
                    C     DATA IS ALREADY A ZERO OR A ONE, POSSIBLY AFTER TRANSFORMATION.MCNMAR
                    C     IF A VALUE EQUALS THE SPLIT VALUE, BOTH OBSERVATIONS CORRES-   MCNMAR
                    C     PONDING TO IT ARE THROWN OUT.  THIS IS ACTUALLY AN ERROR       MCNMAR
                    C     THAT THE PROGRAM FORGIVES.                                     MCNMAR
                    C                                                                    MCNMAR
                    C     THE CRITICAL LEVEL IS DETERMINED BY EITHER THE BINOMIAL        MCNMAR
                    C     OR THE CHI-SQUARE DISTRIBUTIONS, DEPENDING ON B + C.           MCNMAR
                    C                                                                    MCNMAR
                    C     FOR DETAILS ON THE MC NEMAR TEST, SEE CONOVER, W.J.,           MCNMAR
                    C     PRACTICAL NON-PARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 206-210.MCNMAR
                    C                                                                    MCNMAR
                    C                                                                    MCNMAR
                    C     COMMON BLOCKS                                                  MCNMAR
                    C                                                                    MCNMAR
ISN C003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                         DECLARE
ISN C004            INTEGER*4  ROWS,COLS                                                 DECLARE
ISN C005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                             DECLARE
ISN C006            INTEGER*4  DISTN,WHTRN                                               DECLARE
ISN C007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                          DECLARE
ISN C008            REAL*8  BUFFER                                                       DECLARE
ISN C009            INTEGER*4  EOD,CDCOL,BUFNO                                           DECLARE
ISN C010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD                DECLARE
ISN C011            REAL*8   FMT                                                         DECLARE
ISN C012            INTEGER*4  DFORM,DOUT,RCBCD                                          DECLARE
ISN C013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),          DECLARE
                   1    VALUE(40),IVALUE(40),RC                                         DECLARE
ISN C014            REAL*8   PARM,SUBPRM                                                 DECLARE
ISN C015            INTEGER*4  RC                                                        DECLARE
```

```
ISN C016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                                DECLARE
ISN C017            INTEGER*4  TAIL                                                           DECLARE
ISN C018            LOGICAL*4  TESTS,DOALL                                                    DECLARE
ISN C019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),                   DECLARE
                   1    DATE,TIME,PAGE,LINE                                                   DECLARE
ISN C020            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                          DECLARE
ISN C021            INTEGER*4  PAGE                                                           DECLARE
ISN C022            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)         DECLARE
ISN C023            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE             DECLARE
ISN C024            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                       DECLARE
ISN C025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                       DECLARE
ISN C026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                                  DECLARE
ISN C027            INTEGER*4  ITEMS,LOCAT,RDNO                                               DECLARE
ISN C028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                              DECLARE
ISN C029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                                    DECLARE
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                                             DECLARE
ISN C031            REAL*8  BLANK /' '/                                                       DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                                DECLARE
                C                                                                             MCNMAR
                C     OTHER DATA USED IN THIS TEST                                            MCNMAR
                C                                                                             MCNMAR
ISN C033            INTEGER*4  A,B,C,D,TIES                                                   MCNMAR
ISN C034            INTEGER*4  HO /'HO'/, H1 /'H1'/, HYPLBL(2) /'.EQ.', '.NE.'/               MCNMAR
ISN C035            REAL*8  XLBL /'X'/, YLBL /'Y'/                                            MCNMAR
ISN C036            INTEGER*4  CURSMP,CURVAR,CURCNT                                           MCNMAR
ISN C037            REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                                MCNMAR
                C                                                                             MCNMAR
                C     FORMAT STATEMENTS                                                       MCNMAR
ISN C038          231 FORMAT ('-', T12, 'IKMS31I  UNEQUAL SAMPLE/VARIATE SIZES AFTER TRAMCNMAR
                     INSFORMATION')                                                           MCNMAR
ISN C039          3C1 FORMAT ('-', T46, 'THE MC NEMAR TEST FOR SIGNIFICANCE OF CHANGE')     MCNMAR
ISN C040          3C2 FORMAT ('0', T12, A2, ': P(', A8, '= 0) ',A4, ' P(',                  MCNMAR
                     1    A8, ' = 0)')                                                        MCNMAR
ISN C041          3C3 FORMAT ('0', T12, 'THE SAMPLE SIZE IS', I6)                           MCNMAR
ISN C042          3C4 FORMAT ('0', T12, 'THERE WERE', I5, ' SAMPLES DISCARDED DUE TO BEIMCNMAR
                     ING EQUAL TO THE DICHOTOMIZATION LEVEL')                                 MCNMAR
ISN C043          3C5 FORMAT ('0', T12, 'VERSUS')                                           MCNMAR
ISN C044          3C6 FORMAT ('0', T12, 'A =', I5, 10X, 'B =', I5, 10X, 'C =',              MCNMAR
                     1    I5, 10X, 'D =', I5)                                              MCNMAR
ISN C045          3C7 FORMAT ('0', T12, 'THE TEST STATISTIC IS', F8.2)                      MCNMAR
ISN C046          3C8 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')            MCNMAR
ISN C047          3C9 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                                MCNMAR
ISN C048          31C FORMAT ('-', T56, 'END OF THE MC NEMAR TEST')                         MCNMAR
```

```
ISN C049        311 FORMAT ('-', T12, 'THE DATA IS DICHOTOMIZED AT', G12.5)          MCNMAR
                C                                                                   MCNMAR
                C     START OF THE TEST - WRITE HEADING, FIX LABELS                 MCNMAR
                C                                                                   MCNMAR
ISN C050              WKSZE = WKSZE                                                 MCNMAR
ISN C051              WRITE (LST,301)                                               MCNMAR
                C                                                                   MCNMAR
ISN C052              IF (LABEL(1) .EQ. BLANK) LABEL(1) = XLBL                      MCNMAR
ISN C054              IF (LABEL(2) .EQ. BLANK) LABEL(2) = YLBL                      MCNMAR
                C                                                                   MCNMAR
ISN C056              WRITE (LST,302) H0, LABEL(1), HYPLBL(1), LABEL(2)             MCNMAR
ISN C057              WRITE (LST,305)                                               MCNMAR
ISN C058              WRITE (LST,302) H1, LABEL(1), HYPLBL(2), LABEL(2)             MCNMAR
ISN C059              CALL OFLOW(9)                                                 MCNMAR
                C                                                                   MCNMAR
                C     FIRST PASS OF THE DATA                                        MCNMAR
                C                                                                   MCNMAR
ISN C060              REWIND UT1                                                    MCNMAR
ISN C061              REWIND UT2                                                    MCNMAR
                C                                                                   MCNMAR
ISN C062           14 READ (UT1,END=21) CURSMP, CURVAR,CURCNT, RDNO, IENT, X(1)     MCNMAR
ISN C063              IF (TYPE .EQ. 6) GO TO 16                                     MCNMAR
                C                                                                   MCNMAR
                C     BIVARIATE SAMPLE                                              MCNMAR
ISN C065              IF (CURVAR .NE. 1) GO TO 14                                   MCNMAR
                C                                                                   MCNMAR
                C     WRITE THE FIRST SAMPLE OR VARIATE ON UT2                      MCNMAR
                C                                                                   MCNMAR
ISN C067           15 WRITE (UT2) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)               MCNMAR
ISN C068              GO TO 14                                                      MCNMAR
                C                                                                   MCNMAR
                C     2 MATCHED SAMPLES                                             MCNMAR
                C                                                                   MCNMAR
ISN C069           16 IF (CURSMP .NE. 1) GO TO 14                                   MCNMAR
ISN C071              ITEMS = 1                                                     MCNMAR
ISN C072              LOCAT(4) = 0                                                  MCNMAR
ISN C073              IF (WHTRN .EQ. 1) CALL TRANS                                  MCNMAR
ISN C075              IF (ITEMS .NE. 1) GO TO 14                                    MCNMAR
ISN C077              IF (OUT .EQ. 0 .OR. DOUT .NE. 2) GO TO 15                     MCNMAR
                C     WRITE OUT DATA FOR USER                                       MCNMAR
ISN C079              CALL WRDATA                                                   MCNMAR
ISN C080              GO TO 15                                                      MCNMAR
                C                                                                   MCNMAR
```

```
                C     END OF PASS 1                                                    MCNMAR
                C                                                                      MCNMAR
ISN C081           21 ENDFILE UT2                                                      MCNMAR
ISN C082              REWIND UT1                                                       MCNMAR
ISN C083              REWIND UT2                                                       MCNMAR
ISN C084              A = 0                                                            MCNMAR
ISN C085              B = 0                                                            MCNMAR
ISN C086              C = 0                                                            MCNMAR
ISN C087              D = 0                                                            MCNMAR
ISN C088              TIES = 0                                                         MCNMAR
ISN C089              NSZ = 0                                                          MCNMAR
ISN C090              IF (DISTN .NE. -1) DPARM1 = .5                                   MCNMAR
                C                                                                      MCNMAR
                C     SECOND PASS OF THE DATA                                          MCNMAR
                C                                                                      MCNMAR
ISN C092           22 READ (UT1,END=41) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(2)            MCNMAR
ISN C093              IF (TYPE .EQ. 6) GO TO 31                                        MCNMAR
                C                                                                      MCNMAR
                C     BIVARIATE SAMPLE                                                 MCNMAR
                C                                                                      MCNMAR
ISN C095              IF (CURVAR .EQ. 1) GO TO 22                                      MCNMAR
ISN C097              READ (UT2,END=99) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)            MCNMAR
ISN C098              ITEMS = 2                                                        MCNMAR
ISN C099              LOCAT(4) = 0                                                     MCNMAR
ISN C100              IF (WHTRN .EQ. 1) CALL TRANS                                     MCNMAR
ISN C102              IF (ITEMS .NE. 2) GO TO 22                                       MCNMAR
ISN C104              IF (CUT .EQ. 0 .OR. DCUT .NE. 2) GO TO 51                        MCNMAR
                C     WRITE OUT THE DATA FOR THE USER                                  MCNMAR
ISN C106              CALL WRDATA                                                      MCNMAR
ISN C107              GO TO 51                                                         MCNMAR
                C     2 MATCHED SAMPLES                                                MCNMAR
ISN C108           31 IF (CURSMP .EQ. 1) GO TO 22                                      MCNMAR
ISN C110              ITEMS = 1                                                        MCNMAR
ISN C111              X(1) = X(2)                                                      MCNMAR
ISN C112              LOCAT(4) = 0                                                     MCNMAR
ISN 0113              IF (WHTRN .EQ. 1) CALL TRANS                                     MCNMAR
ISN C115              IF (ITEMS .NE. 1) GO TO 22                                       MCNMAR
ISN C117              IF (CUT .EQ. 0 .OR. DCUT .NE. 2) GO TO 32                        MCNMAR
                C     WRITE OUT THE SECOND SAMPLE FOR THE USER                         MCNMAR
ISN C119              CALL WRDATA                                                      MCNMAR
                C                                                                      MCNMAR
ISN C120           32 X(2) = X(1)                                                      MCNMAR
ISN 0121              READ (UT2,END=99) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)            MCNMAR
```

```
                C                                                                       MCNMAR
                C                                                                       MCNMAR
ISN C122           51 NSZ = NSZ + 1                                                     MCNMAR
ISN C123              IF (X(1) - DPARM1) 53, 52, 54                                     MCNMAR
                C                                                                       MCNMAR
ISN C124           52 TIES = TIES + 1                                                   MCNMAR
ISN C125              GO TO 22                                                          MCNMAR
                C                                                                       MCNMAR
ISN C126           53 IF (X(2) - DPARM1) 55, 52, 56                                     MCNMAR
                C                                                                       MCNMAR
ISN C127           54 IF (X(2) - DPARM1) 57, 52, 58                                     MCNMAR
                C     BOTH ZERO                                                         MCNMAR
ISN C128           55 A = A + 1                                                         MCNMAR
ISN C129              GO TO 22                                                          MCNMAR
                C     FIRST ZERO, SECOND ONE                                            MCNMAR
ISN C130           56 B = B + 1                                                         MCNMAR
ISN C131              GO TO 22                                                          MCNMAR
                C     SECOND ZERO, FIRST ONE                                            MCNMAR
ISN C132           57 C = C + 1                                                         MCNMAR
ISN C133              GO TO 22                                                          MCNMAR
                C     BOTH ONE                                                          MCNMAR
ISN C134           58 D = D + 1                                                         MCNMAR
ISN C135              GO TO 22                                                          MCNMAR
                C                                                                       MCNMAR
                C     END OF SECOND PASS - PERFORM TEST                                 MCNMAR
                C                                                                       MCNMAR
ISN C136           41 IF (OUT .NE. 0 .AND. COUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT  MCNMAR
ISN C138              OUT = 0                                                           MCNMAR
ISN C139              WRITE (LST,311) DPARM1                                            MCNMAR
ISN C140              WRITE (LST,303) NSZ                                               MCNMAR
ISN C141              CALL OFLOW(5)                                                     MCNMAR
ISN C142              IF (TIES .EQ. 0) GO TO 42                                         MCNMAR
ISN C144              WRITE (LST,304) TIES                                              MCNMAR
ISN C145              CALL OFLOW(2)                                                     MCNMAR
                C                                                                       MCNMAR
ISN C146           42 N = B + C                                                         MCNMAR
ISN C147              WRITE (LST,306) A, B, C, D                                        MCNMAR
ISN C148              IF (N .LE. 20) GO TO 43                                           MCNMAR
                C                                                                       MCNMAR
ISN C150              T = (B - C) ** 2                                                  MCNMAR
ISN C151              T = T / N                                                         MCNMAR
ISN 0152              GO TO 44                                                          MCNMAR
                C                                                                       MCNMAR
```

```
ISN C153          43 T = B                                                    MCNMAR
                C                                                             MCNMAR
ISN C154          44 WRITE (LST,307) T                                        MCNMAR
ISN C155             CALL OFLOW(4)                                            MCNMAR
ISN C156             IF (N .GT. 20) GO TO 61                                  MCNMAR
                C                                                             MCNMAR
                C     DETERMINE THE CRITICAL REGION WITH THE BINOMIAL DISTRIBUTION MCNMAR
                C                                                             MCNMAR
ISN C158             IT = T                                                   MCNMAR
ISN C159             NIT = N - IT                                             MCNMAR
ISN C160             IT = MIN0(IT,NIT)                                        MCNMAR
ISN C161             ALFHAT = BINOM(N,.5,IT)                                  MCNMAR
ISN C162             ALFHAT = 2. * AMIN1(ALFHAT, 1.-ALFHAT)                   MCNMAR
ISN C163             IF (ALFHAT .GT. 1.) ALFHAT = 1.                          MCNMAR
ISN C165             I = 1                                                    MCNMAR
ISN C166             IF (ALPHA .GE. ALFHAT) I = 2                             MCNMAR
ISN C168             GO TO 91                                                 MCNMAR
                C                                                             MCNMAR
                C     DETERMINE THE CRITICAL REGION ACCORDING TO THE CHI-SQUARE DISTN MCNMAR
ISN C169          61 ALFHAT = CDTR(T,1.)                                      MCNMAR
ISN C170             IF (RC .NE. 0) GO TO 98                                  MCNMAR
ISN C172             ALFHAT = AMIN1(ALFHAT, 1.-ALFHAT)                        MCNMAR
ISN C173             I = 1                                                    MCNMAR
ISN C174             IF (ALPHA .GT. ALFHAT) I = 2                             MCNMAR
                C                                                             MCNMAR
                C     PRINT THE CONCLUSION REACHED                            MCNMAR
                C                                                             MCNMAR
ISN C176          91 WRITE (LST,308) ALPHA, ACCREJ(I)                         MCNMAR
ISN C177             WRITE (LST,309) ALFHAT                                   MCNMAR
ISN C178             CALL OFLOW(5)                                            MCNMAR
                C                                                             MCNMAR
                C     RETURN TO USER                                          MCNMAR
                C                                                             MCNMAR
ISN C179          98 WRITE (LST,310)                                          MCNMAR
ISN C180             CALL OFLOW(3)                                            MCNMAR
ISN C181             RETURN                                                   MCNMAR
                C                                                             MCNMAR
                C     USER GENERATED ERROR IN THE DATA                        MCNMAR
                C                                                             MCNMAR
ISN 0182          99 WRITE (LST,231)                                          MCNMAR
ISN C183             CALL OFLOW(3)                                            MCNMAR
ISN 0184             RC = 12                                                  MCNMAR
ISN C185             GO TO 98                                                 MCNMAR
```

```
                C                                                                    MCNMAR
   ISN C186           END                                                            MCNMAR
*OPTICNS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45
*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
*STATISTICS*     SCURCE STATEMENTS =     185 ,PROGRAM SIZE =     3014
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** ENC OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE MEDIAN                                                MEDIAN
              C                                                                      MEDIAN
              C    THE MEDIAN TEST                                                   MEDIAN
              C                                                                      MEDIAN
              C    THE MEDIAN IS A TEST OF "C" RANDOM SAMPLES TO DETERMINE           MEDIAN
              C    WHETHER OR NOT EACH SAMPLE HAS THE SAME MEDIAN.  THIS IS          MEDIAN
              C    DONE BY COUNTING THE NUMBER OF OBSERVATIONS IN EACH SAMPLE        MEDIAN
              C    ABOVE AND BELOW (OR EQUAL TO) THE GRAND MEDIAN AND CALCULATING    MEDIAN
              C    A TEST STATISTIC WHICH IS DISTRIBUTED ACCORDING TO THE            MEDIAN
              C    CHI-SQUARE DISTRIBUTION WITH "C" - 1 DEGREES OF FREEDOM           MEDIAN
              C    UNDER A TRUE NULL HYPOTHESIS.                                     MEDIAN
              C                                                                      MEDIAN
              C    THE MEDIAN TEST CAN ALSO BE PERFORMED ON DATA OF THE RCB          MEDIAN
              C    TYPE (OBSERVATIONS ASSIGNED TO BLOCK, TREATMENTS, AND             MEDIAN
              C    REPLICATIONS).  THREE MEDIAN TESTS ARE PERFORMED ON THIS          MEDIAN
              C    TYPE OF DATA, ONE FOR EACH CLASSIFICATION.                        MEDIAN
              C                                                                      MEDIAN
              C    FOR EXACT DETAILS ON THE MEDIAN TEST, SEE CONOVER, W.J.,          MEDIAN
              C    PRACTICAL NON-PARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 266-275.  MEDIAN
              C                                                                      MEDIAN
              C                                                                      MEDIAN
              C    COMMON BLOCKS                                                     MEDIAN
              C                                                                      MEDIAN
 ISN 0003          COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                      DECLARE
 ISN 0004          INTEGER*4  ROWS,COLS                                              DECLARE
 ISN 0005          COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                          DECLARE
 ISN 0006          INTEGER*4  DISTN,WHTRN                                            DECLARE
 ISN 0007          COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                       DECLARE
 ISN 0008          REAL*8  BUFFER                                                    DECLARE
 ISN 0009          INTEGER*4  EOD,CDCOL,BUFNO                                        DECLARE
 ISN 0010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD             DECLARE
 ISN 0011          REAL*8   FMT                                                      DECLARE
 ISN 0012          INTEGER*4  DFORM,DOUT,RCBCD                                       DECLARE
 ISN 0013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),       DECLARE
                  1    VALUE(40),IVALUE(40),RC                                       DECLARE
 ISN 0014          REAL*8   PARM,SUBPRM                                              DECLARE
 ISN 0015          INTEGER*4  RC                                                     DECLARE
 ISN 0016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                       DECLARE
 ISN 0017          INTEGER*4  TAIL                                                   DECLARE
 ISN 0018          LOGICAL*4  TESTS,DOALL                                            DECLARE
 ISN 0019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),           DECLARE
```

```
                1    DATE,TIME,PAGE,LINE                                                DECLARE
ISN C020         REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                       DECLARE
ISN C021         INTEGER*4  PAGE                                                       DECLARE
ISN C022         COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)     DECLARE
ISN C023         INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE        DECLARE
ISN C024         COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                   DECLARE
ISN C025         INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                   DECLARE
ISN C026         COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                              DECLARE
ISN C027         INTEGER*4  ITEMS,LOCAT,RDNO                                           DECLARE
ISN C028         COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                          DECLARE
ISN C029         INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                                DECLARE
ISN C030         EQUIVALENCE (WORK(1),IWRK(1))                                         DECLARE
ISN C031         REAL*8  BLANK /' '/                                                   DECLARE
ISN C032         EQUIVALENCE (BLANK,IBLANK)                                            DECLARE
                C                                                                      MEDIAN
                C     OTHER DATA USED                                                  MEDIAN
                C                                                                      MEDIAN
ISN C033         REAL*8  BLKLBL /'BLOCKS'/, TRTLBL /'TRTMNTS'/, REPLBL /'REPS'/       MEDIAN
ISN C034         REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                            MEDIAN
ISN C035         INTEGER*4  CURSMP,CURVAR,CURCNT,TOT,A,B,IRCD(6)                       MEDIAN
ISN C036         REAL*4  RECORD(6)                                                     MEDIAN
ISN C037         EQUIVALENCE (CURSMP,RECORD(1)), (CURVAR,RECORD(2)),                   MEDIAN
                1    (CURCNT,RECORD(3)), (IRDNO,RECORD(5)), (RECORD(1),IRCD(1))       MEDIAN
                C                                                                      MEDIAN
                C     FORMAT STATEMENTS                                                MEDIAN
                C                                                                      MEDIAN
ISN C038        294 FORMAT ('-', T12, 'IKMS941  PREMATURE END OF FILE ON UNIT', I3)    MEDIAN
ISN C039        301 FORMAT ('-', T60, 'THE MEDIAN TEST')                               MEDIAN
ISN 0040        302 FORMAT ('0', T12, 'HO: THE MEDIANS OF ALL', I3,                    MEDIAN
                   1    ' SAMPLES ARE THE SAME'/ '0', T12, 'VERSUS'/ '0', T12,        MEDIAN
                   2    'H1: THE MEDIANS OF AT LEAST 2 SAMPLES ARE NOT THE SAME')     MEDIAN
ISN C041        303 FORMAT ('-', T12, 'HO: THE MEDIANS OF ALL', I3, 1X, A8,            MEDIAN
                   1    ' ARE THE SAME'/ '0', T12, 'VERSUS'/ '0', T12,                MEDIAN
                   2    'H1: THE MEDIANS OF AT LEAST 2 ', A8,                         MEDIAN
                   3    ' ARE NOT THE SAME')                                          MEDIAN
ISN C042        304 FORMAT ('0', T12, 'THERE IS A TOTAL OF', I6,                       MEDIAN
                   1    ' OBSERVATIONS WITH THE FOLLOWING BREAKDOWN:'/                MEDIAN
                   2    '0', ' SAMPLE NUMBER', T32, 'LESS THAN OR EQUAL',             MEDIAN
                   3    T62, 'GREATER THAN', T82, 'TOTAL SIZE'/                       MEDIAN
                   4    (' ', A8, I5, T38, I5, T65, I5, T84, I5) )                    MEDIAN
ISN C043        305 FORMAT ('0', T12, 'THE GRAND MEDIAN IS', G16.8)                    MEDIAN
ISN C044        306 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5, ' WITH',        MEDIAN
                   1    I3, ' DEGREES OF FREEDOM')                                    MEDIAN
```

```
ISN C045        307 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')        MEDIAN
ISN C046        308 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                            MEDIAN
ISN C047        309 FORMAT ('-', T57, 'END OF THE MEDIAN TEST')                       MEDIAN
ISN C048        310 FORMAT ('-', T12, 'THERE IS A TOTAL OF', I6, ' OBSERVATIONS')     MEDIAN
                C                                                                     MEDIAN
                C     PRINT THE TEST HEADING AND FIX POSSIBLE MISSING LABELS          MEDIAN
                C                                                                     MEDIAN
ISN C049              WKSZE = WKSZE                                                   MEDIAN
ISN C050              WRITE (LST,301)                                                 MEDIAN
ISN C051              CALL OFLOW(3)                                                   MEDIAN
                C                                                                     MEDIAN
ISN C052              IF (BLOCK .EQ. BLANK) BLOCK = BLKLBL                            MEDIAN
ISN C054              IF (TREAT .EQ. BLANK) TREAT = TRTLBL                            MEDIAN
ISN C056              IF (REPLI .EQ. BLANK) REPLI = REPLBL                            MEDIAN
                C                                                                     MEDIAN
                C     SPLIT HERE ON SEVERAL RANDOM SAMPLES OR RCB TYPE DATA           MEDIAN
                C                                                                     MEDIAN
ISN C058              IF (TYPE .EQ. 7) GO TC 1001                                     MEDIAN
                C                                                                     MEDIAN
                C     "C" RANDOM SAMPLES                                              MEDIAN
                C                                                                     MEDIAN
ISN C060              WRITE (LST,302) SMPLS                                           MEDIAN
ISN C061              CALL OFLOW(6)                                                   MEDIAN
                C                                                                     MEDIAN
                C     CHECK FOR POSSIBLE TRANSFORMATION OR SAVING OF DATA             MEDIAN
                C                                                                     MEDIAN
ISN C062              IF (WHTRN .NE. 1 .AND. (OUT .EQ. 0 .OR. DOUT .NE. 2) ) GO TO 21 MEDIAN
                C                                                                     MEDIAN
                C     MUST TRANSFORM THE DATA OR SAVE THE DATA OR BOTH BY SEPARATE    MEDIAN
                C     SAMPLES, RATHER THAN BY ORDER READ IN                           MEDIAN
                C                                                                     MEDIAN
ISN C064              REWIND UT2                                                      MEDIAN
                C                                                                     MEDIAN
ISN C065              DO 14 I = 1, SMPLS                                              MEDIAN
ISN C066              REWIND UT1                                                      MEDIAN
                C                                                                     MEDIAN
ISN C067           11 READ (UT1,END=14) RECORD                                        MEDIAN
ISN C068              IF (CURSMP .NE. I) GO TO 11                                     MEDIAN
                C                                                                     MEDIAN
ISN C070              ITEMS = 1                                                       MEDIAN
ISN C071              LOCAT(4) = 0                                                    MEDIAN
ISN C072              X(1) = RECORD(6)                                                MEDIAN
ISN C073              IF (WHTRN .EQ. 1) CALL TRANS                                    MEDIAN
```

```
ISN C075            IF (ITEMS .LT. 1) GO TO 11                                    MEDIAN
ISN C077            IF (OUT .EQ. 0 .OR. OCUT .NE. 2) GO TO 12                     MEDIAN
                C   WRITE OUT ALL THE ITEMS THE USER SUPPLIED                     MEDIAN
ISN C079            CALL WRDATA                                                   MEDIAN
                C                                                                 MEDIAN
                C   SAVE ALL OF THE TRANSFORMED DATA ON UT2                       MEDIAN
                C                                                                 MEDIAN
ISN C080         12 DO 13 J = 1, ITEMS                                            MEDIAN
ISN C081            RECORD(6) = X(J)                                              MEDIAN
ISN C082         13 WRITE (UT2) RECORD                                            MEDIAN
ISN C083            GO TO 11                                                      MEDIAN
                C                                                                 MEDIAN
ISN C084         14 CONTINUE                                                      MEDIAN
                C                                                                 MEDIAN
                C   END OF TRANSFORM AND/OR SAVE DATA PASS                        MEDIAN
                C                                                                 MEDIAN
ISN C085            ENDFILE UT2                                                   MEDIAN
ISN C086            IF (OUT .NE. 0 .AND. COUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT MEDIAN
                C                                                                 MEDIAN
                C   SORT THE DATA DOWN COMPLETELY                                 MEDIAN
                C                                                                 MEDIAN
ISN C088            CALL RANKER (UT4,UT2,LT3,UT4,TOT)                             MEDIAN
                C   SORT THE TRANSFORMED DATA AND HAVE IT PUT ON UT4              MEDIAN
ISN C089            SUBS(1) = -1                                                  MEDIAN
ISN C090            IF (WHTRN .EQ. 1) SUBS(1) = 1                                 MEDIAN
ISN C092            SUBS(2) = UT4                                                 MEDIAN
ISN C093            GO TO 22                                                      MEDIAN
                C                                                                 MEDIAN
                C   NO TRANSFORMATION OR DATA SAVE - SORT DIRECTLY ONTO UT4       MEDIAN
                C                                                                 MEDIAN
ISN C094         21 CALL RANKER (UT4,UT1,LT2,UT3,TOT)                             MEDIAN
ISN C095            SUBS(1) = -1                                                  MEDIAN
ISN C096            SUBS(2) = UT4                                                 MEDIAN
                C                                                                 MEDIAN
                C   CHECK FOR SAVING OF DATA AFTER RANKING                       MEDIAN
                C                                                                 MEDIAN
ISN C097         22 IF (OUT .EQ. 0 .OR. OCUT .NE. 3) GO TO 31                     MEDIAN
                C                                                                 MEDIAN
                C   SAVE THE DATA FOR THE USER RANKED WITHIN SAMPLES              MEDIAN
                C                                                                 MEDIAN
ISN C099            ITEMS = 1                                                     MEDIAN
ISN C100            LOCAT(4) = 0                                                  MEDIAN
ISN C101            DO 24 I = 1, SMPLS                                            MEDIAN
```

```
ISN C102            REWIND UT4                                                  MEDIAN
                C                                                               MEDIAN
ISN C103          23 READ (UT4,END=24) RECORD                                   MEDIAN
ISN C104            X(1) = RECORD(6)                                            MEDIAN
ISN C105            CALL WRDATA                                                 MEDIAN
ISN C106            GO TO 23                                                    MEDIAN
                C                                                               MEDIAN
ISN C107          24 CONTINUE                                                   MEDIAN
ISN C108            IF (OUT .NE. LST) ENDFILE OUT                               MEDIAN
                C                                                               MEDIAN
                C     PROCESS DATA FOR OBTAIN THE GRAND MEDIAN                  MEDIAN
ISN C110          31 OUT = 0                                                    MEDIAN
ISN C111            ITOT = TOT / 2                                              MEDIAN
ISN C112            REWIND UT4                                                  MEDIAN
                C                                                               MEDIAN
ISN C113            DO 32 I = 1, ITOT                                           MEDIAN
ISN C114            READ (UT4,END=1294) RECORD                                  MEDIAN
ISN C115          32 CONTINUE                                                   MEDIAN
                C                                                               MEDIAN
                C     HALF OF THE TOTAL SAMPLE HAS BEEN READ - GET NEXT ONE     MEDIAN
                C                                                               MEDIAN
ISN C116            GMEDN = RECORD(6)                                           MEDIAN
ISN C117            READ (UT4,END=1294) RECORD                                  MEDIAN
ISN C118            IF (ITOT*2 .EQ. TOT) GO TO 33                               MEDIAN
                C                                                               MEDIAN
                C     TOTAL SIZE IS ODD - USE MIDDLE OBSERVATION                MEDIAN
                C                                                               MEDIAN
ISN C120            GMEDN = RECORD(6)                                           MEDIAN
ISN C121            GO TO 34                                                    MEDIAN
                C                                                               MEDIAN
                C     TOTAL SIZE IS EVEN - AVERAGE THE 2 MIDDLE VALUES          MEDIAN
                C                                                               MEDIAN
ISN C122          33 GMEDN = (GMEDN + RECORD(6)) / 2.                           MEDIAN
                C                                                               MEDIAN
                C     INITIALIZE THE COUNTERS NEEDED TO PERFORM THE TEST        MEDIAN
                C                                                               MEDIAN
ISN C123          34 REWIND UT4                                                 MEDIAN
ISN C124            NSZ = 0                                                     MEDIAN
ISN C125            A = 0                                                       MEDIAN
ISN C126            B = 0                                                       MEDIAN
ISN C127            DO 35 I = 1, SMPLS                                          MEDIAN
ISN C128            SSIZE(I) = 0                                                MEDIAN
ISN 0129            IWRK(I) = 0                                                 MEDIAN
```

```
ISN C130          35 IWRK(SMPLS+I) = 0                                          MEDIAN
                C                                                               MEDIAN
                C     READ THE CATA AND CLASSIFY IT AS ABOVE OR BELOW THE MEDIAN MEDIAN
                C                                                               MEDIAN
ISN C131          36 READ (UT4,END=41) RECORD                                   MEDIAN
ISN C132             NSZ = NSZ + 1                                              MEDIAN
ISN C133             SSIZE(CURSMP) = SSIZE(CURSMP) + 1                          MEDIAN
                C                                                               MEDIAN
ISN C134             IF (RECORD(6) .LE. GMEDN) GO TO 37                         MEDIAN
                C                                                               MEDIAN
                C     GREATER THAN THE MEDIAN                                   MEDIAN
                C                                                               MEDIAN
ISN C136             A = A + 1                                                  MEDIAN
ISN C137             IWRK(CURSMP) = IWRK(CLRSMP) + 1                            MEDIAN
ISN C138             GO TO 36                                                   MEDIAN
                C                                                               MEDIAN
                C     LESS THAN OR EQUAL TO THE GRAND MEDIAN                    MEDIAN
                C                                                               MEDIAN
ISN C139          37 B = B + 1                                                  MEDIAN
ISN C140             IWRK(CURSMP+SMPLS) = IWRK(CURSMP+SMPLS) + 1                MEDIAN
ISN C141             GO TO 36                                                   MEDIAN
                C                                                               MEDIAN
                C     CALCULATE THE TEST STATISTIC                              MEDIAN
                C                                                               MEDIAN
ISN C142          41 T = 0                                                      MEDIAN
ISN C143             DO 42 I = 1, SMPLS                                         MEDIAN
ISN C144             01I2 = IWRK(I) * IWRK(I)                                   MEDIAN
ISN C145          42 T = T + 01I2 / SSIZE(I)                                    MEDIAN
                C                                                               MEDIAN
ISN C146             T = T * (NSZ * NSZ) / (A * B)                              MEDIAN
ISN C147             T = T - (NSZ * A) / B                                      MEDIAN
ISN C148             DF = SMPLS - 1                                             MEDIAN
                C                                                               MEDIAN
                C     PRINT OUT THE INFORMATION SO FAR                          MEDIAN
                C                                                               MEDIAN
ISN C149             WRITE (LST,304) NSZ, (LABEL(I), I, IWRK(SMPLS+I), IWRK(I), MEDIAN
                    1    SSIZE(I), I = 1, SMPLS)                               MEDIAN
ISN C150             J = SMPLS + 3                                              MEDIAN
ISN C151             CALL CFLOW(J)                                              MEDIAN
                C                                                               MEDIAN
ISN C152             WRITE (LST,305) GMEDN                                      MEDIAN
ISN C153             IDF = DF                                                   MEDIAN
ISN C154             WRITE (LST,306) T, IDF                                     MEDIAN
```

```
ISN C155            CALL OFLOW(4)                                          MEDIAN
                C                                                          MEDIAN
                C   DETERMINE THE CRITICAL REGION AND GIVE THE RESULTS     MEDIAN
                C                                                          MEDIAN
ISN C156            ALFHAT = CDTR(T,DF)                                    MEDIAN
ISN C157            I = 1                                                  MEDIAN
ISN C158            IF (ALPHA .GT. ALFHAT) I = 2                           MEDIAN
                C                                                          MEDIAN
ISN C160            WRITE (LST,307) ALPHA,ACCREJ(I)                        MEDIAN
ISN C161            WRITE (LST,308) ALFHAT                                 MEDIAN
ISN C162            CALL OFLOW(5)                                          MEDIAN
                C                                                          MEDIAN
                C   END OF THE TEST(S) - RETURN                            MEDIAN
                C                                                          MEDIAN
ISN C163         91 WRITE (LST,309)                                        MEDIAN
ISN C164            CALL OFLOW(3)                                          MEDIAN
ISN C165            RETURN                                                 MEDIAN
                C                                                          MEDIAN
                C   ERROR - END OF FILE IN THE WRONG PLACE                 MEDIAN
                C                                                          MEDIAN
ISN C166       1294 WRITE (LST,294) UT4                                    MEDIAN
ISN C167            RC = 16                                                MEDIAN
ISN C168            GO TO 91                                               MEDIAN
                C                                                          MEDIAN
                C                                                          MEDIAN
                C                                                          MEDIAN
                C   MEDIAN TESTS ON BLOCKS, TREATMENTS, AND REPLICATIONS  MEDIAN
                C                                                          MEDIAN
                C   THE TERM REPLICATION IS A MISNOMER IN MANY CASES SINCE IT  MEDIAN
                C   COULD ACTUALLY BE ANOTHER FACTOR                       MEDIAN
                C                                                          MEDIAN
                C   IMMEDIATELY SORT ALL THE DATA BY SAMPLE VALUE          MEDIAN
                C                                                          MEDIAN
                C   THE USER IS NOT ALLOWED TO TRANSFORM OR SAVE SEPARATED  MEDIAN
                C   OR SORTED RCB TYPE DATA                                MEDIAN
                C                                                          MEDIAN
ISN C169       1001 CALL RANKER (UT4,UT1,LT2,UT3,TOT)                      MEDIAN
ISN C170            SUBS(1) = -1                                           MEDIAN
ISN C171            SUBS(2) = UT4                                          MEDIAN
                C                                                          MEDIAN
                C   INITIALIZE THE PARAMETERS FOR 3 MEDIAN TESTS          MEDIAN
                C                                                          MEDIAN
ISN C172            SSIZE(1) = BLK                                         MEDIAN
```

```
ISN C173            SSIZE(2) = TRT                                          MEDIAN
ISN C174            SSIZE(3) = REP                                          MEDIAN
ISN C175            LABEL(1) = BLOCK                                        MEDIAN
ISN C176            LABEL(2) = TREAT                                        MEDIAN
ISN C177            LABEL(3) = REPLI                                        MEDIAN
ISN C178            NSZ = BLK * TRT * REP                                   MEDIAN
ISN C179            WRITE (LST,310) NSZ                                     MEDIAN
ISN C180            CALL OFLOW(3)                                           MEDIAN
                C                                                           MEDIAN
                C     PERFORM THE TEST 3 TIMES                              MEDIAN
                C                                                           MEDIAN
ISN C181            DO 1099 IWH = 1, 3                                      MEDIAN
                C     IF THERE IS ONLY 1 LEVEL OF THE FACTOR, SKIP TEST     MEDIAN
ISN C182            IF (SSIZE(IWH) .LE. 1) GO TO 1099                       MEDIAN
                C                                                           MEDIAN
                C     PRINT THE HYPOTHSES FOR THIS TEST                     MEDIAN
                C                                                           MEDIAN
ISN C184            WRITE (LST,303) SSIZE(IWH), LABEL(IWH), LABEL(IWH)      MEDIAN
ISN C185            CALL OFLOW(7)                                           MEDIAN
ISN C186            IF (IWH - 2) 1011, 1012, 1013                           MEDIAN
ISN C187     1011   ISZ1 = TRT                                              MEDIAN
ISN C188            ISZ2 = REP                                              MEDIAN
ISN C189            GO TO 1014                                              MEDIAN
ISN C190     1012   ISZ1 = BLK                                              MEDIAN
ISN C191            ISZ2 = REP                                              MEDIAN
ISN C192            GO TO 1014                                              MEDIAN
ISN C193     1013   ISZ1 = BLK                                              MEDIAN
ISN C194            ISZ2 = TRT                                              MEDIAN
                C                                                           MEDIAN
ISN C195     1014   ISZ = ISZ1 * ISZ2                                       MEDIAN
ISN C196            IMID = SSIZE(IWH) / 2                                   MEDIAN
ISN C197            IMID1 = IMID + 1                                        MEDIAN
ISN C198            ISZA = 2 * ISZ                                          MEDIAN
ISN C199            DO 1015 I = 1, ISZ                                      MEDIAN
ISN C200     1015   IWRK(I) = 0                                             MEDIAN
                C                                                           MEDIAN
                C     READ THE DATA AND SAVE MIDDLE VALUES FOR THE MEDIANS  MEDIAN
                C                                                           MEDIAN
ISN C201            REWIND UT4                                              MEDIAN
                C                                                           MEDIAN
ISN C202     1020   READ (UT4,END=1031) RECORD                              MEDIAN
ISN C203            IF (IWH - 2) 1021, 1022, 1023                           MEDIAN
                C                                                           MEDIAN
```

```
                C     PICK UP THE IMPORTANT SUBSCRIPTS                                  MEDIAN
                C                                                                       MEDIAN
ISN C204         1021 IX1 = CURVAR                                                      MEDIAN
ISN C205              IX2 = CURCNT                                                      MEDIAN
ISN C206              GO TO 1024                                                        MEDIAN
ISN C207         1022 IX1 = CURSMP                                                      MEDIAN
ISN C208              IX2 = CURCNT                                                      MEDIAN
ISN C209              GO TO 1024                                                        MEDIAN
ISN 0210         1023 IX1 = CURSMP                                                      MEDIAN
ISN C211              IX2 = CURVAR                                                      MEDIAN
                C                                                                       MEDIAN
                C     CHECK IF VALUE IN QUESTION IS A MIDDLE VALUE                      MEDIAN
                C                                                                       MEDIAN
ISN 0212         1024 IWHR = (IX1 - 1) * ISZ2 + IX2                                     MEDIAN
ISN C213              IWRK(IWHR) = IWRK(IWHR) + 1                                       MEDIAN
ISN 0214              IF (IWRK(IWHR) .EQ. IMID) WORK(ISZ+IWHR) = RECORD(6)              MEDIAN
ISN C216              IF (IWRK(IWHR) .EQ. IMID1) WORK(ISZA+IWHR) = RECORD(6)            MEDIAN
ISN 0218              GO TO 1020                                                        MEDIAN
                C                                                                       MEDIAN
                C     CALCULATE THE MEDIANS FROM THE VALUES SAVED                       MEDIAN
                C                                                                       MEDIAN
ISN C219         1031 IF (2*IMID .NE. SSIZE(IWH)) GO TO 1033                            MEDIAN
                C     AVERAGE THE MIDDLE VALUES                                         MEDIAN
ISN C221              DO 1032 I = 1, ISZ                                                MEDIAN
ISN C222         1032 WORK(ISZA+I) = (WORK(ISZA+I) + WORK(ISZ+I)) / 2.                  MEDIAN
                C                                                                       MEDIAN
                C     SET UP THE CONTINGENCY TABLE                                      MEDIAN
                C                                                                       MEDIAN
ISN C223         1033 ISZB = ISZ * 3                                                    MEDIAN
ISN C224              NSIZE = SSIZE(IWH)                                                MEDIAN
ISN C225              J = 2 * NSIZE                                                     MEDIAN
ISN C226              DO 1034 I = 1, J                                                  MEDIAN
ISN C227         1034 IWRK(ISZB+I) = 0                                                  MEDIAN
ISN C228              A = 0                                                             MEDIAN
ISN C229              B = 0                                                             MEDIAN
ISN C230              REWIND UT4                                                        MEDIAN
                C                                                                       MEDIAN
                C     READ THE DATA AND CLASSIFY INTO THE TABLE                         MEDIAN
                C                                                                       MEDIAN
ISN 0231         1040 READ (UT4,END=1051) RECORD                                        MEDIAN
ISN C232              IF (IWH - 2) 1041, 1042, 1043                                     MEDIAN
                C                                                                       MEDIAN
                C     GET THE PROPER SUBSCRIPTS FOR CLASSIFICATION                      MEDIAN
```

```
                C                                                                MEDIAN
ISN C233         1041 IX1 = CURVAR                                               MEDIAN
ISN C234              IX2 = CURCNT                                               MEDIAN
ISN C235              IX3 = CURSMP                                               MEDIAN
ISN C236              GO TO 1044                                                 MEDIAN
ISN C237         1042 IX1 = CURSMP                                               MEDIAN
ISN C238              IX2 = CURCNT                                               MEDIAN
ISN C239              IX3 = CURVAR                                               MEDIAN
ISN C240              GO TO 1044                                                 MEDIAN
ISN C241         1043 IX1 = CURSMP                                               MEDIAN
ISN C242              IX2 = CURVAR                                               MEDIAN
ISN C243              IX3 = CURCNT                                               MEDIAN
                C                                                                MEDIAN
                C     FIND THE MEDIAN AND COMPARE THIS VALUE                     MEDIAN
                C                                                                MEDIAN
ISN C244         1044 IWHR = (IX1 - 1) * ISZ2 + IX2                              MEDIAN
ISN C245              IF (RECORD(6) .LE. WORK(ISZA+IWHR)) GO TO 1045             MEDIAN
                C     GREATER THAN ITS CORRESPONDING MEDIAN                      MEDIAN
ISN 0247              IWRK(ISZB+IX3) = IWRK(ISZB+IX3) + 1                        MEDIAN
ISN C248              A = A + 1                                                  MEDIAN
ISN C249              GO TO 1040                                                 MEDIAN
                C     LESS THAN OR EQUAL TO THE MEDIAN                           MEDIAN
ISN 0250         1045 IWRK(ISZB+NSIZE+IX3) = IWRK(ISZB+NSIZE+IX3) + 1            MEDIAN
ISN C251              B = B + 1                                                  MEDIAN
ISN 0252              GO TO 1040                                                 MEDIAN
                C                                                                MEDIAN
                C     CALCULATE THE TEST STATISTIC                               MEDIAN
                C                                                                MEDIAN
ISN C253         1051 T = 0.                                                     MEDIAN
ISN C254              DO 1052 I = 1, NSIZE                                       MEDIAN
ISN C255              OII2 = IWRK(ISZB+I) * IWRK(ISZB+I)                         MEDIAN
ISN C256         1052 T = T + OII2 / ISZ                                         MEDIAN
                C                                                                MEDIAN
ISN C257              T = T * (NSZ * NSZ) / (A * B)                              MEDIAN
ISN C258              T = T - (NSZ * A) / B                                      MEDIAN
ISN C259              DF = NSIZE - 1                                             MEDIAN
                C                                                                MEDIAN
                C     PRINT THE TEST STATISTIC                                   MEDIAN
                C                                                                MEDIAN
ISN C260              IDF = DF                                                   MEDIAN
ISN C261              WRITE (LST,306) T, IDF                                     MEDIAN
ISN C262              CALL OFLOW(2)                                              MEDIAN
                C                                                                MEDIAN
```

```
                C     DETERMINE THE CRITICAL REGION AND ACCEPT OR REJECT                MEDIAN
                C                                                                       MEDIAN
   ISN C263             ALFHAT = CDTR(T,DF)                                             MEDIAN
   ISN C264             I = 1                                                           MEDIAN
   ISN 0265             IF (ALPHA .GT. ALFHAT) I = 2                                    MEDIAN
                C                                                                       MEDIAN
   ISN C267             WRITE (LST,307) ALPHA,ACCREJ(I)                                 MEDIAN
   ISN 0268             WRITE (LST,308) ALFHAT                                          MEDIAN
   ISN C269             CALL OFLOW(5)                                                   MEDIAN
                C                                                                       MEDIAN
                C     END OF THE TEST FOR THIS FACTOR                                   MEDIAN
                C                                                                       MEDIAN
   ISN C270      1099 CONTINUE                                                          MEDIAN
                C                                                                       MEDIAN
                C     END OF ALL THREE TESTS                                            MEDIAN
                C                                                                       MEDIAN
   ISN C271             GO TO 91                                                        MEDIAN
                C                                                                       MEDIAN
   ISN C272             END                                                             MEDIAN

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     271 ,PROGRAM SIZE =     5422

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            SUBROUTINE MOOD                                                    MOOD
                    C                                                                  MOOD
                    C     MOOD'S TEST OF VARIANCES                                     MOOD
                    C                                                                  MOOD
                    C     THE MOOD TEST OF VARIANCES TESTS THE VARIANCES OF 2 RANDOM   MOOD
                    C     SAMPLES (NOT NECESSARILY THE SAME SIZES).  THE TEST CAN      MOOD
                    C     DETERMINE IF THE VARIANCE OF THE FIRST SAMPLE IS LESS THAN,  MOOD
                    C     GREATER THAN, OR EQUAL TO THE VARIANCE OF THE SECOND SAMPLE. MOOD
                    C                                                                  MOOD
                    C     THE LARGE SAMPLE APPROXIMATION IS USED TO DETERMINE THE      MOOD
                    C     CRITICAL REGION.  THIS MAKES THE CRITICAL REGION CALCULATED  MOOD
                    C     POSSIBLY INVALID.  SINCE THE QUESTION OF WHAT CONSTITUTES    MOOD
                    C     A SMALL SAMPLE IS OPEN TO DEBATE, THIS PROGRAM WILL          MOOD
                    C     CALCULATE THE CRITICAL REGION FOR ALL SAMPLE SIZES,          MOOD
                    C     BUT WILL GIVE AN APPROPRIATE WARNING IF EITHER SAMPLE        MOOD
                    C     CONTAINS LESS THAN 20 OBSERVATIONS.                          MOOD
                    C                                                                  MOOD
                    C     FOR MORE DETAILS ON MOOD'S TEST, SEE CONOVER, W.J.,          MOOD
                    C     PRACTICAL NON-PARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 370-371. MOOD
                    C                                                                  MOOD
                    C                                                                  MOOD
                    C     COMMON BLOCKS                                                MOOD
                    C                                                                  MOOD
ISN C003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                       DECLARE
ISN C004            INTEGER*4  ROWS,COLS                                               DECLARE
ISN C005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                           DECLARE
ISN C006            INTEGER*4  DISTN,WHTRN                                             DECLARE
ISN C007            COMMON /FLAG/  EOD,COCOL,BUFFER(10,2),BUFNO                        DECLARE
ISN C008            REAL*8  BUFFER                                                     DECLARE
ISN C009            INTEGER*4  EOD,COCOL,BUFNO                                         DECLARE
ISN C010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD              DECLARE
ISN C011            REAL*8   FMT                                                       DECLARE
ISN C012            INTEGER*4  DFORM,DOUT,RCBCD                                        DECLARE
ISN C013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),        DECLARE
                   1    VALUE(40),IVALUE(40),RC                                        DECLARE
ISN C014            REAL*8   PARM,SUBPRM                                               DECLARE
ISN C015            INTEGER*4  RC                                                      DECLARE
ISN C016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                        DECLARE
ISN C017            INTEGER*4  TAIL                                                    DECLARE
ISN C018            LOGICAL*4  TESTS,DOALL                                             DECLARE
ISN C019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),            DECLARE
```

```
                     1    DATE,TIME,PAGE,LINE                                          DECLARE
ISN C020              REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                DECLARE
ISN C021              INTEGER*4  PAGE                                                  DECLARE
ISN C022              COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
ISN C023              INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE    DECLARE
ISN C024              COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT              DECLARE
ISN C025              INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT              DECLARE
ISN C026              COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                         DECLARE
ISN C027              INTEGER*4  ITEMS,LOCAT,RDNO                                      DECLARE
ISN C028              COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                     DECLARE
ISN C029              INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                           DECLARE
ISN C030              EQUIVALENCE (WORK(1),IWRK(1))                                    DECLARE
ISN C031              REAL*8  BLANK /' '/                                              DECLARE
ISN C032              EQUIVALENCE (BLANK,IBLANK)                                       DECLARE
                    C                                                                  MOOD
                    C     OTHER DATA AND CONSTANTS                                     MOOD
                    C                                                                  MOOD
ISN C033              REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                       MOOD
ISN C034              INTEGER*2  H0 /'H0'/, H1 /'H1'/                                  MOOD
ISN C035              INTEGER*4  HYPLBL(2,3) /'.EQ.', '.NE.', '.LE.', '.GT.',          MOOD
                     1    '.GE.', '.LT.'/                                             MOOD
ISN C036              REAL*8  XLBL /'X'/, YLBL /'Y'/                                   MOOD
ISN C037              REAL*4  RECORD(6)                                                MOOD
ISN C038              INTEGER*4  CURSMP, CURVAR, CURCNT, TIES, PLCS, TOT               MOOD
ISN C039              EQUIVALENCE (RECORD(1),CURSMP), (RECORD(2),CURVAR),              MOOD
                     1    (RECORD(3),CURCNT), (RECORD(6),OBS)                         MOOD
ISN C040              REAL*8  T1, T2, T3                                               MOOD
                    C                                                                  MOOD
                    C     FORMAT STATEMENTS                                            MOOD
                    C                                                                  MOOD
ISN C041          234 FORMAT ('-', T12, 'IKMS34I  INVALID SAMPLE SIZE(S)')             MOOD
ISN C042          290 FORMAT ('-', T12, 'IKMS90I  ***** BECAUSE OF SAMPLE SIZES, ALPHA HMOOD
                     1AT MAY NOT BE CORRECT.  VERIFY WITH TABLES. *****')             MOOD
ISN C043          301 FORMAT ('-', T56, 'MOOD''S TEST OF VARIANCES')                   MOOD
ISN C044          302 FORMAT ('0', T12, A2, ': VAR(', A8, ') ', A4, ' VAR(', A8, ')')  MOOD
ISN C045          303 FORMAT ('0', T12, 'VERSUS')                                      MOOD
ISN C046          304 FORMAT ('0', (T12, A8, ' HAS', I6, ' OBSERVATIONS'/ ) )          MOOD
ISN C047          305 FORMAT ('0', T12, 'THERE WAS A TOTAL OF', I5,                    MOOD
                     1    ' TIED OBSERVATIONS IN', I3, ' PLACES')                     MOOD
ISN C048          306 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5)               MOOD
ISN C049          307 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')       MOOD
ISN C050          308 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                           MOOD
ISN C051          309 FORMAT ('-', T53, 'END OF MOOD''S TEST OF VARIANCES')            MOOD
```

```
                C                                                                MOOD
                C     START BY FIXING THE LABELS AND PRINTING THE HYPOTHESES     MOOD
                C                                                                MOOD
ISN C052              WKSZE = WKSZE                                              MOOD
ISN C053              WRITE (LST,301)                                            MOOD
ISN C054              IF (LABEL(1) .EQ. BLANK) LABEL(1) = XLBL                   MOOD
ISN C056              IF (LABEL(2) .EQ. BLANK) LABEL(2) = YLBL                   MOOD
                C                                                                MOOD
ISN C058              ITL = 1                                                    MOOD
ISN C059              IF (TAIL) 11, 13, 12                                       MOOD
ISN C060           11 ITL = 2                                                    MOOD
ISN C061              GO TO 13                                                   MOOD
ISN C062           12 ITL = 3                                                    MOOD
                C                                                                MOOD
ISN C063           13 WRITE (LST,302) HO, LABEL(1), HYPLBL(1,ITL), LABEL(2)     MOOD
ISN C064              WRITE (LST,303)                                            MOOD
ISN C065              WRITE (LST,302) H1, LABEL(1), HYPLBL(2,ITL), LABEL(2)      MOOD
ISN C066              CALL OFLOW(9)                                              MOOD
                C                                                                MOOD
ISN C067              IF (SUBS(1) .NE. 0) GO TO 31                               MOOD
                C                                                                MOOD
                C     DATA WILL HAVE TO BE WORKED OVER IN SOME MANNER            MOOD
                C                                                                MOOD
ISN C069              IF (WHTRN .EQ. 1 .OR. (OUT .NE. 0 .AND. DOUT .EQ. 2)) GO TO 21  MOOD
                C                                                                MOOD
                C     ALL THAT IS NEEDED IS SORTING OF THE DATA                  MOOD
                C                                                                MOOD
ISN C071              CALL RANKER(UT4,UT1,UT2,UT3,TOT)                           MOOD
ISN C072              IF (RC .NE. 0) GO TO 92                                    MOOD
ISN C074              SUBS(1) = -1                                               MOOD
ISN C075              SUBS(2) = UT4                                              MOOD
ISN C076              GO TO 31                                                   MOOD
                C                                                                MOOD
                C     MUST PLAY WITH DATA FOR TRANSFORMATION OF WRITE OUT        MOOD
                C                                                                MOOD
ISN C077           21 REWIND UT2                                                 MOOD
ISN C078              SSIZE(1) = 0                                               MOOD
ISN C079              SSIZE(2) = 0                                               MOOD
ISN C080              DO 24 ISMP = 1, SMPLS                                      MOOD
ISN C081              REWIND UT1                                                 MOOD
                C                                                                MOOD
ISN 0082           22 READ (UT1,END=24) RECORD                                   MOOD
ISN C083              IF (ISMP .NE. CURSMP) GO TO 22                             MOOD
```

```
ISN C085            ITEMS = 1                                                  MOOD
ISN C086            LOCAT(4) = 0                                               MOOD
ISN C087            X(1) = OBS                                                 MOOD
ISN C088            IF (WHTRN .EQ. 1) CALL TRANS                               MOOD
ISN C090            IF (ITEMS .LT. 1) GO TO 22                                 MOOD
ISN C092            IF (OUT .NE. 0 .AND. OOUT .EQ. 2) CALL WRDATA              MOOD
                C                                                              MOOD
ISN C094            DO 23 I = 1, ITEMS                                         MOOD
ISN C095            OBS = X(I)                                                 MOOD
ISN C096            SSIZE(CURSMP) = SSIZE(CURSMP) + 1                          MOOD
ISN C097         23 WRITE (UT2) RECORD                                         MOOD
ISN C098            GO TO 22                                                   MOOD
                C                                                              MOOD
ISN C099         24 CONTINUE                                                   MOOD
                C                                                              MOOD
ISN C100            ENDFILE UT2                                                MOOD
                C                                                              MOOD
                C     SORT THE DATA IN TOTALITY                                MOOD
                C                                                              MOOD
ISN C101            CALL RANKER(UT4,UT2,UT3,UT4,TOT)                           MOOD
ISN C102            IF (RC .NE. C) GO TO 92                                    MOOD
ISN C104            SUBS(1) = -1                                               MOOD
ISN C105            SUBS(2) = UT4                                              MOOD
ISN C106            IF (WHTRN .EQ. 1) SUBS(1) = 1                              MOOD
                C                                                              MOOD
                C     DATA HAS BEEN COMPLETELY SORTED AT THIS POINT            MOOD
                C                                                              MOOD
ISN C108         31 IF (OUT .EQ. 0 .OR. OOUT .NE. 3) GO TO 36                  MOOD
ISN 0110            IUT1 = SUBS(2)                                             MOOD
ISN C111            ITEMS = 1                                                  MOOD
ISN 0112            LOCAT(4) = 0                                               MOOD
ISN C113            SSIZE(1) = 0                                               MOOD
ISN C114            SSIZE(2) = 0                                               MOOD
                C                                                              MOOD
ISN C115            DO 33 ISMP = 1, SMPLS                                      MOOD
ISN 0116            REWIND IUT1                                                MOOD
                C                                                              MOOD
ISN C117         32 READ (IUT1,END=33) RECORD                                  MOOD
ISN 0118            IF (ISMP .NE. CURSMP) GO TO 32                             MOOD
ISN 0120            SSIZE(CURSMP) = SSIZE(CURSMP) + 1                          MOOD
ISN C121            X(1) = OBS                                                 MOOD
ISN 0122            CALL WRDATA                                                MOOD
ISN 0123            GO TO 32                                                   MOOD
```

```
                C                                                                   MOOD
ISN 0124           33 CONTINUE                                                      MOOD
                C                                                                   MOOD
                C     READ THE DATA TO CALCLLATE THE TEST STATISTIC                 MOOD
                C                                                                   MOOD
ISN C125           36 IF (OUT. NE. 0 .AND. COUT .GE. 2 .AND. OUT .NE. LST) ENDFILE OUT  MOOD
ISN C127              OUT = 0                                                       MOOD
                C                                                                   MOOD
ISN 0128              RANK = 0.                                                     MOOD
ISN C129              T = 0.                                                        MOOD
ISN C130              TIES = 0                                                      MOOD
ISN C131              PLCS = 0                                                      MOOD
ISN C132              DIFF = SSIZE(1) + SSIZE(2) + 1                                MOOD
ISN C133              DIFF = DIFF / 2.                                              MOOD
ISN C134              IUT1 = SUBS(2)                                                MOOD
ISN C135              REWIND IUT1                                                   MOOD
ISN C136              READ (IUT1,END=99) RECORD                                     MOOD
                C                                                                   MOOD
                C     SAVE THE VALUE OF THE OBSERVATION AND SAMPLE NUMBER           MOOD
                C                                                                   MOOD
ISN C137           41 XVAL = OBS                                                    MOOD
ISN 0138              ISMP = CURSMP                                                 MOOD
ISN C139              RANK = RANK + 1.                                              MOOD
                C                                                                   MOOD
ISN 0140              READ (IUTI,END=42) RECORD                                     MOOD
ISN C141              IF (XVAL .EQ. OBS) GO TO 51                                   MOOD
                C                                                                   MOOD
                C     NON-TIED OBSERVATION                                          MOOD
                C                                                                   MOOD
ISN C143              IF (ISMP .EQ. 1) T = T + (RANK - DIFF) ** 2                   MOOD
ISN C145              GO TO 41                                                      MOOD
                C                                                                   MOOD
                C     END OF FILE - COUNT LAST OBSERVATION                          MOOD
                C                                                                   MOOD
ISN C146           42 IF (ISMP .EQ. 1) T = T + (RANK - DIFF) ** 2                   MOOD
ISN 0148              GO TO 61                                                      MOOD
                C                                                                   MOOD
                C     OBSERVATIONS TIED                                             MOOD
                C                                                                   MOOD
ISN 0149           51 PLCS = PLCS + 1                                               MOOD
ISN C150              I = 2                                                         MOOD
ISN C151              K = 0                                                         MOOD
ISN C152              IF (ISMP .EQ. 1) K = 1                                        MOOD
```

```
                C                                                                MOOD
ISN C154           52 IF (CURSMP .EQ. 1) K = K + 1                               MOOD
                C                                                                MOOD
                C     READ THE NEXT OBSERVATION AND CHECK FOR IT BEING TIED      MOOD
                C                                                                MOOD
ISN C156           READ (IUT1,END=55) RECORD                                     MOOD
ISN C157           IF (XVAL .NE. OBS) GO TO 53                                   MOOD
ISN C159           I = I + 1                                                     MOOD
ISN C160           GO TO 52                                                      MOOD
                C                                                                MOOD
                C     CALCULATE CONTRIBUTION OF TIED VALUES                      MOOD
                C                                                                MOOD
ISN C161           53 Z = 0.                                                     MOOD
ISN C162              DO 54 L = 1, I                                             MOOD
ISN C163              Z = Z + RANK                                               MOOD
ISN C164           54 RANK = RANK + 1.                                           MOOD
                C                                                                MOOD
ISN C165              TIES = TIES + I                                            MOOD
ISN C166              Z = Z / I                                                  MOOD
ISN C167              T = T + K * (Z - DIFF) ** 2                                MOOD
ISN C168              RANK = RANK - 1                                            MOOD
ISN C169              GO TO 41                                                   MOOD
                C                                                                MOOD
                C     END OF FILE WHILE LOOKING FOR MORE TIES                    MOOD
                C                                                                MOOD
ISN C170           55 Z = 0.                                                     MOOD
ISN C171              DO 56 L = 1, I                                             MOOD
ISN C172              Z = Z + RANK                                               MOOD
ISN C173           56 RANK = RANK + 1                                            MOOD
                C                                                                MOOD
ISN C174              TIES = TIES + I                                            MOOD
ISN C175              Z = Z / I                                                  MOOD
ISN C176              T = T + K * (Z - DIFF) ** 2                                MOOD
                C                                                                MOOD
                C     TEST STATISTIC HAS BEEN CALCULATED                         MOOD
                C                                                                MOOD
ISN C177           61 WRITE (LST,304) LABEL(1),SSIZE(1),LABEL(2),SSIZE(2)        MOOD
ISN C178              CALL OFLOW(2)                                              MOOD
ISN C179              IF (SSIZE(1) .EQ. 0 .OR. SSIZE(2) .EQ. 0) GO TO 99         MOOD
                C                                                                MOOD
ISN C181              IF (TIES .EQ. 0) GO TO 62                                  MOOD
ISN C183              WRITE (LST,305) TIES, PLCS                                 MOOD
ISN C184              CALL OFLOW(2)                                              MOOD
```

```
                C                                                                   MOOD
ISN 0185           62 WRITE (LST,306) T                                             MOOD
                C                                                                   MOOD
                C     CALCULATE THE CRITICAL REGION                                 MOOD
                C                                                                   MOOD
ISN 0186              T1 = SSIZE(1) * (SSIZE(1) + SSIZE(2) + 1)                     MOOD
                     1    * (SSIZE(1) + SSIZE(2) - 1)                             MOOD
ISN 0187              T1 = T1 / 12.                                                 MOOD
ISN 0188              T2 = SSIZE(1) * SSIZE(2) * (SSIZE(1) + SSIZE(2) + 1)          MOOD
                     1    * (SSIZE(1) + SSIZE(2) + 2) * (SSIZE(1) + SSIZE(2) - 2)  MOOD
ISN 0189              T2 = DSQRT(T2 / 180.DC)                                       MOOD
ISN 0190              T3 = T                                                        MOOD
ISN 0191              XP = (T3 - T1) / T2                                           MOOD
ISN 0192              ALFHAT = ANDTR(XP)                                            MOOD
                C                                                                   MOOD
ISN 0193              GO TO (71, 76, 81), ITL                                       MOOD
                C                                                                   MOOD
                C     2-TAILED TEST                                                 MOOD
                C                                                                   MOOD
ISN 0194           71 ALFHAT = 2. * AMIN1(ALFHAT, 1.-ALFHAT)                        MOOD
ISN 0195              IF (ALFHAT .GT. 1.) ALFHAT = 1.                               MOOD
ISN 0197              GO TO 91                                                      MOOD
                C                                                                   MOOD
                C     LOWER TAILED TEST                                             MOOD
                C                                                                   MOOD
ISN 0198           76 GO TO 91                                                      MOOD
                C                                                                   MOOD
                C     UPPER TAILED TEST                                             MOOD
                C                                                                   MOOD
ISN 0199           81 ALFHAT = 1. - ALFHAT                                          MOOD
                C                                                                   MOOD
                C     ACCEPT OR REJECT ACCORDINGLY                                  MOOD
                C                                                                   MOOD
ISN 0200           91 I = 1                                                         MOOD
ISN 0201              IF (ALPHA .GT. ALFHAT) I = 2                                  MOOD
ISN 0203              WRITE (LST,307) ALPHA, ACCREJ(I)                              MOOD
ISN 0204              WRITE (LST,308) ALFHAT                                        MOOD
ISN 0205              CALL OFLOW(5)                                                 MOOD
                C                                                                   MOOD
ISN 0206              IF (SSIZE(1) .GT. 20 .AND. SSIZE(2) .GT. 20) GO TO 92         MOOD
ISN 0208              WRITE (LST,290)                                               MOOD
ISN 0209              CALL OFLOW(3)                                                 MOOD
                C                                                                   MOOD
```

```
                C     END OF MCOD'S TEST                                        MOOD
                C                                                               MOOD
   ISN C210        92 WRITE (LST,309)                                           MOOD
   ISN C211           CALL OFLOW(3)                                             MOOD
   ISN C212           RETURN                                                    MOOD
                C                                                               MOOD
                C     ERROR - BAD SAMPLE SIZE                                   MOOD
                C                                                               MOOD
   ISN C213        99 WRITE (LST,234)                                           MOOD
   ISN C214           CALL OFLOW(3)                                             MOOD
   ISN C215           RC = 12                                                   MOOD
   ISN C216           GO TO 92                                                  MOOD
                C                                                               MOOD
   ISN C217           END                                                       MOOD

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     216 ,PROGRAM SIZE =     3694

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
  ISN C002            SUBROUTINE OFLOW (/NLINE/)                                        OFLOW
                C                                                                       OFLOW
                C     THIS SUBROUTINE ADDS THE QUANTITY PASSED IN THE CALL STATEMENT    OFLOW
                C     TO THE LINE COUNT, 'LINE', IN THE COMMON  BLOCK /PRETTY/.         OFLOW
                C     IT WILL THEN TEST FOR POSSIBLE PAGE OVERFLOW, I.E., LINE IS       OFLOW
                C     GREATER THAN OR EQUAL TO MAXLIN, WHICH IS SET TO THE NUMBER       OFLOW
                C     OF LINES DESIRES ON A PAGE.  IF PAGE OVERFLOW HAS OCCURRED, THIS  OFLOW
                C     SUBROUTINE WILL CALL THE SUBROUTINE 'HEAD', TO PRINT THE PAGE     OFLOW
                C     HEADING AND RESET THE LINE COUNT.                                 OFLOW
                C     THE VALUE OF ZERO, PASSED AS THE QUANTITY TO BE ADDED TO THE      OFLOW
                C     LINE COUNT HAS A SPECIAL MEANING.  IF, WHEN NLINE (THE PASSED     OFLOW
                C     QUANTITY) IS ZERO, THE CURRENT LINE COUNT (LINE) IS ALSO ZERO,    OFLOW
                C     THE PAGE HEADING IS PRINTED UNCONDITIONALLY.                      OFLOW
                C                                                                       OFLOW
                C     COMMON BLOCKS                                                     OFLOW
                C                                                                       OFLOW
  ISN C003            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),           OFLOW
                     1    DATE,TIME,PAGE,LINE                                          OFLOW
  ISN C004            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                OFLOW
  ISN C005            INTEGER*4  PAGE                                                   OFLOW
                C                                                                       OFLOW
                C     DEFINE THE MAXIMUM DESIRED LINES PER PAGE                         OFLOW
                C                                                                       OFLOW
  ISN C006            INTEGER*4 MAXLIN /55/                                             OFLOW
                C                                                                       OFLOW
                C     CHECK FOR NLINE = 0 AND LINE = 0                                  OFLOW
                C                                                                       OFLOW
  ISN C007            IF (NLINE .EQ. 0 .AND. LINE .EQ. 0) GO TO 1                       OFLOW
                C                                                                       OFLOW
                C     ADD NLINE TO LINE AND CHECK FOR PAGE OVERFLOW                     OFLOW
                C                                                                       OFLOW
  ISN C009            LINE = LINE + NLINE                                               OFLOW
  ISN C010            IF (LINE .LT. MAXLIN) GO TO 2                                     OFLOW
                C                                                                       OFLOW
                C     NEW PAGE HEADING ROUTINE NEEDED - CALL HEAD                       OFLOW
                C                                                                       OFLOW
  ISN C012          1 CALL HEAD                                                       OFLOW
                C                                                                       OFLOW
  ISN C013          2 RETURN                                                          OFLOW
                C                                                                       OFLOW
  ISN C014            END                                                               OFLOW
```

```
*OPTICNS IN EFFECT*       NAME=  MAIN,OPT=01,LINECNT=45
*OPTICNS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
*STATISTICS*      SOURCE STATEMENTS =       13 ,PROGRAM SIZE =      262
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            FUNCTION POIS(X,N)                                          POIS
                C                                                               POIS
                C   THIS FUNCTION CALCULATES THE PROBABILITY OF N OR FEWER       POIS
                C   EVENTS OCCURING FROM A POPULATION DISTRIBUTED               POIS
                C   ACCORDING TO THE POISSON DISTRIBUTION WITH MEAN X.          POIS
                C                                                               POIS
                C   BECAUSE OF THE METHOD OF CALCULATION OF THE PROBABILITY,    POIS
                C   THIS FUNCTION WILL BE EXTREMELY SLOW FOR LARGE              POIS
                C   VALUES OF N.                                               POIS
                C                                                               POIS
ISN C003            REAL*8  P,EMX,XX,NFACT,XN,P1                                POIS
                C                                                               POIS
ISN C004            EMX = 0.D0                                                  POIS
ISN C005            P = 0.D0                                                    POIS
                C                                                               POIS
                C   CHECK FOR NEGATIVE ARGUMENTS                                POIS
                C                                                               POIS
ISN C006            IF (N .LT. 0 .OR. X .LT. 0.) GO TO 9                        POIS
ISN C008            XX = X                                                      POIS
ISN C009            EMX = DEXP(-XX)                                             POIS
ISN C010            IF (N - 1) 1, 2, 3                                          POIS
                C                                                               POIS
                C   N IS ZERO                                                   POIS
                C                                                               POIS
ISN C011          1 P = 1.D0                                                    POIS
ISN C012            GO TO 9                                                     POIS
                C                                                               POIS
                C   N IS ONE                                                    POIS
                C                                                               POIS
ISN C013          2 P = XX + 1.D0                                               POIS
ISN C014            GO TO 9                                                     POIS
                C                                                               POIS
                C   N IS GREATER THAN ONE                                       POIS
                C                                                               POIS
ISN C015          3 P = XX + 1.D0                                               POIS
ISN C016            NFACT = 1.D0                                                POIS
ISN C017            XN = XX                                                     POIS
                C                                                               POIS
ISN C018            DO 4 I = 2, N                                               POIS
ISN C019            NFACT = NFACT * I                                           POIS
ISN C020            XN = XN * XX                                                POIS
```

```
   ISN C021            P1 = P + XN / NFACT                                          POIS
   ISN C022            IF (DABS(P1-P) .LT. .001D0) GO TO 9                          POIS
   ISN C024          4 P = P1                                                       POIS
                     C                                                              POIS
                     C     END OF CALCULATION OF MULTIPLIER - CALCULATE PROBABILITY  POIS
                     C                                                              POIS
   ISN C025          9 POIS = EMX * P                                               POIS
                     C                                                              POIS
   ISN C026            RETURN                                                       POIS
                     C                                                              POIS
   ISN C027            END                                                          POIS

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      26 ,PROGRAM SIZE =      612

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
  ISN C002            SUBROUTINE PRLBLS                                                    PRLBLS
                C                                                                          PRLBLS
                C     THIS SUBROUTINE IS USED TO PRINT THE LABELS THAT THE USER            PRLBLS
                C     SPECIFIED, AS AN ECHO CHECK OF THOSE SPECIFICATIONS.                 PRLBLS
                C                                                                          PRLBLS
                C                                                                          PRLBLS
                C     COMMON BLOCKS                                                        PRLBLS
                C                                                                          PRLBLS
  ISN C003            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),              PRLBLS
                     1    DATE,TIME,PAGE,LINE                                             PRLBLS
  ISN C004            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                   PRLBLS
  ISN C005            INTEGER*4  PAGE                                                    PRLBLS
  ISN C006            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)  PRLBLS
  ISN C007            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE      PRLBLS
  ISN C008            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                PRLBLS
  ISN C009            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                PRLBLS
  ISN C010            REAL*8  BLANK /' '/                                                PRLBLS
  ISN C011            EQUIVALENCE (BLANK,IBLANK)                                         PRLBLS
                C                                                                          PRLBLS
                C     FORMAT STATEMENTS                                                    PRLBLS
                C                                                                          PRLBLS
  ISN C012        139 FORMAT (' ', T12, 'NO LABELS PROVIDED')                            PRLBLS
  ISN C013        140 FORMAT ('-', T12, 'THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLPRLBLS
                     1LOWS:' / ' ')                                                      PRLBLS
  ISN C014        141 FORMAT (' ', T12, 'THE SAMPLE IS CALLED ', A8)                     PRLBLS
  ISN C015        142 FORMAT (' ', T12, 'SAMPLE', I3, ' IS CALLED ', A8)                 PRLBLS
  ISN C016        144 FORMAT (' ', T12, 'VARIATE', I3, ' IS CALLED ', A8)                PRLBLS
  ISN C017        147 FORMAT (' ', T12, 'BLOCKS ARE CALLED ', A8/                        PRLBLS
                     1    ' ', T12, 'TREATMENTS ARE CALLED ', A8)                       PRLBLS
  ISN C018        148 FORMAT (' ', T12, 'REPLICATIONS ARE CALLED ', A8)                  PRLBLS
                C                                                                          PRLBLS
                C     WRITE HEADING AND GO TO RCUTINE BASED ON DATA TYPE                   PRLBLS
                C                                                                          PRLBLS
  ISN C019            WRITE (LST,140)                                                    PRLBLS
  ISN C020            CALL OFLOW (3)                                                     PRLBLS
  ISN C021            GO TO (1, 2, 3, 4, 5, 6, 7, 8), TYPE                               PRLBLS
                C                                                                          PRLBLS
                C     1 RANDOM SAMPLE                                                      PRLBLS
                C                                                                          PRLBLS
  ISN C022          1 IF (LABEL(1) .EQ. BLANK) GO TO 10                                  PRLBLS
  ISN C024            WRITE (LST,141) LABEL(1)                                           PRLBLS
```

```
ISN C025            CALL OFLOW (1)                                          PRLBLS
ISN C026            GO TO 11                                                PRLBLS
                C                                                           PRLBLS
                C     2 RANDOM SAMPLES                                      PRLBLS
                C                                                           PRLBLS
ISN C027          2 DO 21 I = 1, SMPLS                                      PRLBLS
ISN C028            IF (LABEL(I) .NE. BLANK) GO TO 22                       PRLBLS
ISN C030         21 CONTINUE                                                PRLBLS
ISN C031            GO TO 10                                                PRLBLS
                C                                                           PRLBLS
ISN C032         22 DO 23 I = 1, SMPLS                                      PRLBLS
ISN C033            IF (LABEL(I) .EQ. BLANK) GO TO 23                       PRLBLS
ISN C035            WRITE (LST,142) I, LABEL(I)                             PRLBLS
ISN C036            CALL OFLOW (1)                                          PRLBLS
ISN C037         23 CONTINUE                                                PRLBLS
ISN C038            GO TO 11                                                PRLBLS
                C                                                           PRLBLS
                C     K RANDOM SAMPLES - USE ROUTINE FOR 2 RANDOM SAMPLES   PRLBLS
                C                                                           PRLBLS
ISN C039          3 GO TO 2                                                 PRLBLS
                C                                                           PRLBLS
                C     BIVARIATE SAMPLE                                      PRLBLS
                C                                                           PRLBLS
ISN C040          4 DO 41 I = 1, VARITE                                     PRLBLS
ISN C041            IF (LABEL(I) .NE. BLANK) GO TO 42                       PRLBLS
ISN C043         41 CONTINUE                                                PRLBLS
ISN C044            GO TO 10                                                PRLBLS
                C                                                           PRLBLS
ISN C045         42 DO 43 I = 1, VARITE                                     PRLBLS
ISN C046            IF (LABEL(I) .EQ. BLANK) GO TO 43                       PRLBLS
ISN C048            WRITE (LST,144) I, LABEL(I)                             PRLBLS
ISN C049            CALL OFLOW (1)                                          PRLBLS
ISN C050         43 CONTINUE                                                PRLBLS
ISN C051            GO TO 11                                                PRLBLS
                C                                                           PRLBLS
                C     MULTIVARIATE SAMPLE - USE ROUTINE FOR BIVARIATE SAMPLE PRLBLS
                C                                                           PRLBLS
ISN C052          5 GO TO 4                                                 PRLBLS
                C                                                           PRLBLS
                C     2 MATCHED SAMPLES - USE ROUTINE FOR 2 RANDOM SAMPLES  PRLBLS
                C                                                           PRLBLS
ISN C053          6 GO TO 2                                                 PRLBLS
                C                                                           PRLBLS
```

```
                C      RANDOMIZED COMPLETE BLOCK DATA                                  PRLBLS
                C                                                                      PRLBLS
    ISN C054          7 IF (BLOCK .EQ. BLANK .AND. TREAT .EQ. BLANK                    PRLBLS
                     1    .AND. REPLI .EQ. BLANK) GO TO 10                            PRLBLS
    ISN C056            WRITE (LST,147) BLOCK, TREAT                                   PRLBLS
    ISN C057            CALL OFLOW (2)                                                 PRLBLS
                C                                                                      PRLBLS
    ISN C058            IF (REPLI .EQ. BLANK) GO TO 11                                 PRLBLS
    ISN C060            WRITE (LST,148) REPLI                                          PRLBLS
    ISN C061            CALL OFLOW (1)                                                 PRLBLS
    ISN C062            GO TO 11                                                       PRLBLS
                C                                                                      PRLBLS
                C      BALANCED INCOMPLETE BLOCK DESIGN TYPE DATA                      PRLBLS
                C               USE ROUTINE FOR RCB                                    PRLBLS
                C                                                                      PRLBLS
    ISN C063          8 GO TO 7                                                        PRLBLS
                C                                                                      PRLBLS
                C      NO LABELS WERE PROVIDED                                         PRLBLS
                C                                                                      PRLBLS
    ISN C064         1C WRITE (LST,139)                                                PRLBLS
    ISN C065            CALL OFLOW (1)                                                 PRLBLS
                C                                                                      PRLBLS
                C      RETURN TO CALLING PROGRAM                                       PRLBLS
                C                                                                      PRLBLS
    ISN C066         11 RETURN                                                         PRLBLS
                C                                                                      PRLBLS
    ISN C067            END                                                            PRLBLS

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      66 ,PROGRAM SIZE =     1214

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C0002           SUBROUTINE RANKER (TO,FROM,WORK1,WORK2,TOTAL)                                  RANKER
ISN C0003           INTEGER*4  TO,FROM,WORK1,WORK2,TOTAL                                           RANKER
                C                                                                                  RANKER
                C     THIS SUBROUTINE IS USED TO SORT THE DATA SO THAT IT CAN                      RANKER
                C     EASILY BE RANKED FOR MANY OF THE TESTS THAT ARE PERFORMED                    RANKER
                C     IN THIS PROGRAM.                                                             RANKER
                C                                                                                  RANKER
                C     THIS PROGRAM ASSUMES THAT THE DATA EXISTS ON AN INTERMEDIATE                 RANKER
                C     WORK UNIT AS 6 WORD RECORDS.  THE DATA IS SORTED ON THE                      RANKER
                C     SIXTH WORD ONLY, BUT ALL 6 WORDS ARE WRITTEN OUT ON THE                      RANKER
                C     SPECIFIED OUTPUT INTERMEDIATE WORK UNIT.                                     RANKER
                C                                                                                  RANKER
                C     "FROM" IS THE UNIT NUMBER WHERE THE DATA CURRENTLY RESIDES.                  RANKER
                C            THE DATA IS LEFT INTACT ON THIS UNIT.                                 RANKER
                C     "TO" IS THE UNIT ON WHICH THE SORTED OUTPUT IS PLACED.                       RANKER
                C     "WORK1" AND "WORK2" ARE 2 UNITS WHICH CAN BE USED BY THIS                    RANKER
                C              SUBROUTINE FOR INTERMEDIATE MERGING OF THE DATA                     RANKER
                C              IF THE DATA CANNOT FIT TOTALLY IN CORE.  "TO"                       RANKER
                C              CAN BE EQUAL TO EITHER 1 OF THESE 2 UNITS IF NECESSARY.             RANKER
                C              NO OTHER DUPLICATIONS ARE ALLOWED.                                  RANKER
                C     "TOTAL" IS THE NUMBER OF ITEMS THAT WERE SORTED.  THIS                       RANKER
                C              WILL MAKE FINDING THE MEDIAN SIMPLE.                                RANKER
                C                                                                                  RANKER
                C                                                                                  RANKER
                C     COMMON BLOCKS                                                                RANKER
                C                                                                                  RANKER
ISN C0004           COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                                   DECLARE
ISN C0005           INTEGER*4  ROWS,COLS                                                           DECLARE
ISN C0006           COMMON /DIST/  DISTN,CPARM1,CPARM2,WHTRN                                       DECLARE
ISN C0007           INTEGER*4  DISTN,WHTRN                                                         DECLARE
ISN C0008           COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                                    DECLARE
ISN C0009           REAL*8  BUFFER                                                                 DECLARE
ISN C0010           INTEGER*4  EOD,CDCOL,BUFNO                                                     DECLARE
ISN C0011           COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD                          DECLARE
ISN C0012           REAL*8   FMT                                                                   DECLARE
ISN C0013           INTEGER*4  DFORM,DOUT,RCBCD                                                    DECLARE
ISN C0014           COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),                    DECLARE
                   1    VALUE(40),IVALUE(40),RC                                                    DECLARE
ISN C0015           REAL*8   PARM,SUBPRM                                                           DECLARE
ISN C0016           INTEGER*4  RC                                                                  DECLARE
ISN C0017           COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                                    DECLARE
```

```
ISN C018            INTEGER*4  TAIL                                                    DECLARE
ISN C019            LOGICAL*4  TESTS,DOALL                                            DECLARE
ISN C020            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),           DECLARE
                   1     DATE,TIME,PAGE,LINE                                          DECLARE
ISN C021            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                  DECLARE
ISN C022            INTEGER*4  PAGE                                                   DECLARE
ISN C023            COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
ISN C024            INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE     DECLARE
ISN C025            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C026            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C027            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                          DECLARE
ISN C028            INTEGER*4  ITEMS,LOCAT,RDNO                                       DECLARE
ISN C029            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                      DECLARE
ISN C030            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                            DECLARE
ISN C031            EQUIVALENCE (WORK(1),IWRK(1))                                     DECLARE
ISN C032            REAL*8  BLANK /' '/                                               DECLARE
ISN C033            EQUIVALENCE (BLANK,IBLANK)                                        DECLARE
                C                                                                     RANKER
                C     OTHER DATA DECLARATIONS                                         RANKER
                C                                                                     RANKER
ISN C034            REAL*4  RECORD(6)                                                 RANKER
ISN C035            INTEGER*4  WK1, WK2, WKOUT                                        RANKER
                C                                                                     RANKER
                C     FORMAT STATEMENTS FOR POSSIBLE ERRORS                           RANKER
                C                                                                     RANKER
ISN C036          295 FORMAT ('-', T12, 'IKMS95I  NO DATA TO SORT')                   RANKER
ISN C037          296 FORMAT ('-', T12, 'IKMS96I  INVALID UNIT PARAMETERS IN CALL TO RANRANKER
                   1KER')                                                             RANKER
                C                                                                     RANKER
                C     CHECK UNIT PARAMETERS SPECIFIED                                 RANKER
                C                                                                     RANKER
ISN C038            TOTAL = 0                                                         RANKER
ISN C039            IF (TO .EQ. FROM) GO TO 1296                                      RANKER
ISN C041            IF (FROM .EQ. WORK1 .OR. FROM .EQ. WORK2) GO TO 1296              RANKER
ISN C043            IF (WORK1 .EQ. WORK2) GO TO 1296                                  RANKER
                C                                                                     RANKER
                C     CHECK FOR OUTPUT UNIT EQUALLY 1 OF THE WORK UNITS               RANKER
                C                                                                     RANKER
ISN C045            WK1 = WORK1                                                       RANKER
ISN C046            WK2 = WORK2                                                       RANKER
ISN C047            IF (TO .EQ. WORK1) GO TO 12                                       RANKER
ISN C049            IF (TO .NE. WORK2) GO TO 13                                       RANKER
                C     TO = WORK2 - FIRST MERGE UNIT WILL BE WORK1                     RANKER
```

```
ISN C0051          11 WKOUT = TO                                                RANKER
ISN C0052             GO TO 13                                                  RANKER
                C     TO = WORK1 - BEST FIRST MERGE UNIT IS WORK2               RANKER
ISN C0053          12 WKOUT = TO                                                RANKER
ISN C0054             WK1 = WORK2                                               RANKER
ISN C0055             WK2 = WORK1                                               RANKER
                C                                                               RANKER
                C     INITIALIZE FOR SORT/MERGE                                 RANKER
                C                                                               RANKER
ISN C0056          13 MERGE = 0                                                 RANKER
ISN C0057             MAX = WKSZE / 6                                           RANKER
ISN C0058             LIM = MAX * 6                                             RANKER
ISN C0059             REWIND FROM                                               RANKER
                C                                                               RANKER
                C     READ INPUT UNTIL EOF OR UNTIL THE WORK ARRAY IS FILLED    RANKER
                C                                                               RANKER
ISN C0060          16 DO 22 INCORE = 1, MAX                                     RANKER
ISN C0061             READ (FROM,END=31) RECORD                                 RANKER
ISN C0062             LOC = (INCORE - 1) * 6                                    RANKER
                C                                                               RANKER
ISN C0063             DO 21 I = 1, 6                                            RANKER
ISN C0064          21 WORK(LOC+I) = RECORD(I)                                   RANKER
                C                                                               RANKER
ISN C0065          22 CONTINUE                                                  RANKER
ISN C0066             INCORE = MAX + 1                                          RANKER
                C                                                               RANKER
                C     END OF FILE OR WORK IS FILLED                             RANKER
                C                                                               RANKER
ISN C0067          31 INCORE = INCORE - 1                                       RANKER
ISN C0068             IF (INCORE - 1) 81, 41, 32                                RANKER
                C                                                               RANKER
                C     SORT THE DATA BY ITEM 6 IN CORE                           RANKER
                C                                                               RANKER
ISN C0069          32 I = 1                                                     RANKER
                C                                                               RANKER
ISN C0070         321 II = 2 * I                                                RANKER
ISN C0071             M = II - 1                                                RANKER
ISN C0072             I = II                                                    RANKER
ISN C0073             IF (I .LE. INCORE) GO TO 321                              RANKER
                C                                                               RANKER
ISN C0075          33 M = M / 2                                                 RANKER
ISN C0076             IF (M .LE. 0) GO TO 41                                    RANKER
                C                                                               RANKER
```

```
ISN C078          34 K = INCORE - M                                          RANKER
ISN C079             J = 1                                                   RANKER
                  C                                                          RANKER
ISN C080          35 I = J                                                   RANKER
                  C                                                          RANKER
ISN C081          36 IF (I .LT. 1) GO TO 38                                  RANKER
                  C                                                          RANKER
ISN C083          37 L = I + M                                               RANKER
ISN C084             LOCL = (L - 1) * 6                                      RANKER
ISN C085             LOCI = (I - 1) * 6                                      RANKER
ISN C086             IF (WORK(LOCL+6) .LT. WORK(LOCI+6)) GO TO 39            RANKER
                  C                                                          RANKER
ISN C088          38 J = J + 1                                               RANKER
ISN C089             IF (J .GT. K) GO TO 33                                  RANKER
ISN C091             GO TO 35                                                RANKER
                  C                                                          RANKER
ISN C092          39 DO 40 LJ = 1,6                                          RANKER
ISN C093             TEMP = WORK(LOCI+LJ)                                    RANKER
ISN C094             WORK(LOCI+LJ) = WORK(LOCL+LJ)                           RANKER
ISN C095          40 WORK(LOCL+LJ) = TEMP                                    RANKER
                  C                                                          RANKER
ISN C096             I = I - M                                               RANKER
ISN C097             GO TO 36                                                RANKER
                  C                                                          RANKER
                  C     END OF INTERNAL SORT - CHECK FOR MERGE WITH PREVIOUS DATA RANKER
                  C                                                          RANKER
ISN 0098          41 IF (MERGE .NE. 0) GO TO 51                              RANKER
                  C                                                          RANKER
                  C     NO OTHER SORTED DATA STORED AT THIS TIME             RANKER
                  C                                                          RANKER
ISN C100             IF (INCORE .LT. MAX) GO TO 46                           RANKER
                  C                                                          RANKER
                  C     CORE COMPLETELY FILLED - MORE DATA ON INPUT          RANKER
                  C                                                          RANKER
ISN C102             MERGE = WK1                                             RANKER
ISN C103             ISUB1 = 0                                               RANKER
ISN C104             REWIND MERGE                                            RANKER
ISN C105             DO 42 I = 1, INCORE                                     RANKER
ISN C106             WRITE (MERGE) (WORK(ISUB1+J), J = 1, 6)                 RANKER
ISN C107          42 ISUB1 = ISUB1 + 6                                       RANKER
                  C                                                          RANKER
                  C     EXCHANGE UNIT NUMBERS - GO TO READ MORE DATA         RANKER
                  C                                                          RANKER
```

```
ISN C108            ENDFILE MERGE                                                  RANKER
ISN C109            WK1 = WK2                                                      RANKER
ISN C110            WK2 = MERGE                                                    RANKER
ISN C111            GO TO 16                                                       RANKER
                C                                                                  RANKER
                C     ALL DATA IS IN CORE - NO MORE LEFT ON INPUT UNIT              RANKER
                C                                                                  RANKER
ISN C112         46 ISUB1 = 0                                                      RANKER
ISN C113            REWIND TO                                                      RANKER
                C                                                                  RANKER
ISN C114            DO 47 I = 1, INCORE                                            RANKER
ISN C115            WRITE (TO) (WORK(ISUB1+J), J = 1, 6)                           RANKER
ISN C116            TOTAL = TOTAL + 1                                              RANKER
ISN C117         47 ISUB1 = ISUB1 + 6                                              RANKER
ISN C118            GO TO 96                                                       RANKER
                C                                                                  RANKER
                C     DATA IN CORE AND DATA ON UNIT "MERGE" FROM LAST PASS         RANKER
                C                                                                  RANKER
ISN C119         51 IF (INCORE .LT. MAX) GO TO 61                                  RANKER
                C                                                                  RANKER
                C     MORE DATA LEFT ON INPLT - PERFORM AN INTERMEDIATE MERGE       RANKER
                C                                                                  RANKER
ISN C121            REWIND MERGE                                                   RANKER
ISN C122            REWIND WK1                                                     RANKER
ISN C123            READ (MERGE) RECORD                                            RANKER
ISN C124            ISUB1 = 0                                                      RANKER
                C                                                                  RANKER
                C     MERGE IN CORE DATA WITH INTERMEDIATE OUTPUT DATA              RANKER
                C                                                                  RANKER
ISN C125            DO 54 I = 1, INCORE                                            RANKER
                C                                                                  RANKER
ISN C126         52 IF (WORK(ISUB1+6) .LT. RECORD(6)) GO TO 53                     RANKER
                C     INTERMEDIATE RECORD LOW (OR EQUAL)                           RANKER
ISN C128            WRITE (WK1) RECORD                                             RANKER
ISN C129            READ (MERGE,END=55) RECORD                                     RANKER
ISN C130            GO TO 52                                                       RANKER
                C     IN CORE ENTRY LOW                                            RANKER
ISN C131         53 WRITE (WK1) (WORK(ISUB1+J), J = 1, 6)                          RANKER
ISN C132         54 ISUB1 = ISUB1 + 6                                              RANKER
                C                                                                  RANKER
ISN C133            GO TO 57                                                       RANKER
                C                                                                  RANKER
                C     END OF INTERMEDIATE OUTPUT DATA - DUMP OUT REST OF DATA IN CORE RANKER
```

```
                C                                                                RANKER
ISN 0134           55 DO 56 K = I, INCORE                                        RANKER
ISN 0135              WRITE (WK1) (WORK(ISUB1+J), J = 1, 6)                      RANKER
ISN 0136           56 ISUB1 = ISUB1 + 6                                          RANKER
                C                                                                RANKER
ISN 0137              GO TO 59                                                   RANKER
                C                                                                RANKER
                C     END OF IN CORE DATA - COPY INTERMEDIATE TO NEW OUTPUT       RANKER
                C                                                                RANKER
ISN 0138           57 WRITE (WK1) RECORD                                         RANKER
ISN 0139              READ (MERGE,END=59) RECORD                                 RANKER
ISN 0140              GO TO 57                                                   RANKER
                C                                                                RANKER
                C     END OF INTERMEDIATE MERGE                                  RANKER
                C                                                                RANKER
ISN 0141           59 ENDFILE WK1                                                RANKER
ISN 0142              WK2 = MERGE                                                RANKER
ISN 0143              MERGE = WK1                                                RANKER
ISN 0144              WK1 = WK2                                                  RANKER
ISN 0145              GO TO 16                                                   RANKER
                C                                                                RANKER
                C     DATA IN CORE AND ON MERGE - NO MORE LEFT ON THE INPUT       RANKER
                C                                                                RANKER
ISN 0146           61 IF (MERGE .EQ. TO) GO TO 71                                RANKER
                C                                                                RANKER
                C     MAKE THE FINAL MERGE DIRECTLY TO THE OUTPUT UNIT            RANKER
                C                                                                RANKER
ISN 0148              REWIND MERGE                                               RANKER
ISN 0149              REWIND TO                                                  RANKER
ISN 0150              READ (MERGE) RECORD                                        RANKER
ISN 0151              ISUB1 = 0                                                  RANKER
                C                                                                RANKER
ISN 0152              DO 64 I = 1, INCORE                                        RANKER
                C                                                                RANKER
ISN 0153           62 IF (WORK(ISUB1+6) .LT. RECORD(6)) GO TO 63                 RANKER
                C     INTERMEDIATE DATA ITEM LOW (OR EQUAL)                      RANKER
ISN 0155              WRITE (TO) RECORD                                          RANKER
ISN 0156              TOTAL = TOTAL + 1                                          RANKER
ISN 0157              READ (MERGE,END=65) RECORD                                 RANKER
ISN 0158              GO TO 62                                                   RANKER
                C     IN CORE ITEM LOW                                           RANKER
ISN 0159           63 WRITE (TO) (WORK(ISUB1+J), J = 1, 6)                       RANKER
ISN 0160              TOTAL = TOTAL + 1                                          RANKER
```

```
                C                                                                   RANKER
ISN C161           64 ISUB1 = ISUB1 + 6                                             RANKER
                C                                                                   RANKER
ISN C162              GO TO 67                                                      RANKER
                C                                                                   RANKER
                C     END OF INTERMEDIATE MERGE DATA - DUMP DATA FROM CORE          RANKER
                C                                                                   RANKER
ISN C163           65 DO 66 K = I, INCORE                                           RANKER
ISN C164              WRITE (TO) (WORK(ISUB1+J), J = 1, 6)                          RANKER
ISN C165              TOTAL = TOTAL + 1                                             RANKER
ISN C166           66 ISUB1 = ISUB1 + 6                                             RANKER
ISN C167              GO TO 69                                                      RANKER
                C                                                                   RANKER
                C     END OF IN CORE DATA - COPY PREVIOUS MERGE TO OUTPUT           RANKER
                C                                                                   RANKER
ISN C168           67 WRITE (TO) RECORD                                             RANKER
ISN C169              TOTAL = TOTAL + 1                                             RANKER
ISN C170              READ (MERGE,END=69) RECORD                                    RANKER
ISN C171              GO TO 67                                                      RANKER
                C                                                                   RANKER
                C     ALL DATA MERGED UNTO THE OUTPUT UNIT                          RANKER
                C                                                                   RANKER
ISN C172           69 GO TO 96                                                      RANKER
                C     NO MORE DATA ON INPUT, BUT INTERMEDIATE DATA IS CURRENTLY     RANKER
                C     ON THE OUTPUT UNIT                                            RANKER
                C                                                                   RANKER
ISN C173           71 REWIND MERGE                                                  RANKER
ISN 0174              REWIND WK1                                                    RANKER
                C     COPY OLD MERGE TO OTHER WORK UNIT                             RANKER
ISN C175           72 READ (MERGE,END=73) RECORD                                    RANKER
ISN C176              WRITE (WK1) RECORD                                            RANKER
ISN 0177              GO TO 72                                                      RANKER
                C                                                                   RANKER
                C     END OF COPY OF MERGED DATA                                    RANKER
                C                                                                   RANKER
ISN C178           73 ENDFILE WK1                                                   RANKER
ISN C179              WK2 = MERGE                                                   RANKER
ISN C180              MERGE = WK1                                                   RANKER
ISN C181              WK1 = WK2                                                     RANKER
ISN C182              GO TO 61                                                      RANKER
                C                                                                   RANKER
                C     THE RARE INSTANCE WHEN ALL THE DATA IS OUT ON AN INTERMEDIATE RANKER
                C     UNIT AND NONE IS IN CORE WHEN EOF IS REACHED ON INPUT         RANKER
```

```
                C                                                                    RANKER
ISN 0183           81 IF (MERGE .EQ. 0) GO TO 1295                                   RANKER
                C                                                                    RANKER
                C     READ THE MERGE UNIT AND WRITE IT OUT IF UNITS ARE NOT EQUAL    RANKER
                C                                                                    RANKER
ISN 0185              REWIND MERGE                                                   RANKER
ISN 0186              IF (MERGE .NE. TO) REWIND TO                                   RANKER
                C                                                                    RANKER
ISN 0188           82 READ (MERGE,END=96) RECORD                                     RANKER
ISN 0189              TOTAL = TOTAL + 1                                              RANKER
ISN 0190              IF (MERGE .NE. TO) WRITE (TO) RECORD                           RANKER
ISN 0192              GO TO 82                                                       RANKER
                C                                                                    RANKER
                C     END OF THE SORT/MERGE                                          RANKER
                C                                                                    RANKER
ISN 0193           96 IF (MERGE .NE. TO) ENDFILE TO                                  RANKER
ISN 0195              REWIND TO                                                      RANKER
ISN 0196              RETURN                                                         RANKER
                C                                                                    RANKER
                C     NO RECORDS ON UNIT "TO"                                        RANKER
                C                                                                    RANKER
ISN 0197         1295 WRITE (LST,295)                                                RANKER
ISN 0198              GO TO 99                                                       RANKER
                C                                                                    RANKER
                C     CONFLICTING I/O UNITS SPECIFIED                                RANKER
                C                                                                    RANKER
ISN 0199         1296 WRITE (LST,296)                                                RANKER
                C                                                                    RANKER
ISN 0200           99 CALL OFLOW(3)                                                  RANKER
ISN 0201              RC = 12                                                        RANKER
ISN 0202              MERGE = TO                                                     RANKER
ISN 0203              GO TO 96                                                       RANKER
                C                                                                    RANKER
ISN 0204              END                                                            RANKER

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     203 ,PROGRAM SIZE =     2742

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            SUBROUTINE RDDATA                                                    RDDATA
                C                                                                        RDDATA
                C     THIS SUBROUTINE IS USED TO PERFORM THE ACTUAL READING OF THE       RDDATA
                C     DATA FOR THE EXPERIMENT BEING TESTED.  IT READS IT FORMATTED       RDDATA
                C     OR UNFORMATTED, ACCORDING TO THE USER REQUEST, AND WRITES          RDDATA
                C     THE DATA OUT ON WORK UNIT UT1 AND ON INTERMEDIATE UNIT             RDDATA
                C     OUT IF THE USER HAS SPECIFIED SAVING OF THE RAW DATA.              RDDATA
                C     IF THE USER SPECIFIED RAW DATA TRANSFORMATION, A CALL IS           RDDATA
                C     MADE TO THE USER SUPPLIED TRANSFORMATION SUBROUTINE, TRANS,        RDDATA
                C     WITH THE PERTINENT DATA INFORMATION IN THE COMMON BLOCK           RDDATA
                C     /USER/.  TRANS CAN INCREASE OR DECREASE THE NUMBER                 RDDATA
                C     OF DATA ITEMS TO BE PROCESSED, AS WELL AS THEIR VALUES, AND        RDDATA
                C     CAN EVEN INDICATE THAT THIS PARTICULAR SET OF OBSERVATIONS         RDDATA
                C     SHOULD BE DELETED COMPLETELY.                                      RDDATA
                C                                                                        RDDATA
                C     THE DATA IS WRITTEN OUT ONTO WORK UNIT UT1, ONE ITEM AT A          RDDATA
                C     TIME, WITH SUFFICIENT LOCATION INFORMATION TO BE ABLE TO           RDDATA
                C     IDENTIFY WHICH SAMPLE OR VARIATE OR BLOCK-TREATMENT-REP           RDDATA
                C     THE OBSERVATION CAME FROM, AS WELL AS INFORMATION PINPOINTING      RDDATA
                C     WHERE THE DATA CAME FROM (READ NUMBER AND ITEM NUMBER).            RDDATA
                C                                                                        RDDATA
                C     THIS SUBROUTINE CALLS THE SUBROUTINE RDFMT TO READ THE USER'S      RDDATA
                C     FORMAT STATEMENT IF A FORMAT STATEMENT IS NEEDED TO READ           RDDATA
                C     THE DATA.                                                          RDDATA
                C                                                                        RDDATA
                C     DATA IS READ FROM THE UNIT 'IPT'.  END OF DATA IS INDICATED        RDDATA
                C     BY EITHER AN ACTUAL END OF FILE SIGNAL OR BY A CONVERSION          RDDATA
                C     ERROR WHEN THE 'END OF DATA' CARD IS READ.  A CONVERSION           RDDATA
                C     ERROR THAT IS NOT CAUSED BY THE 'END OF DATA' CARD WILL            RDDATA
                C     RESULT IN TERMINATION OF THE READING OPERATION AND CONTROL         RDDATA
                C     WILL BE PASSED BACK TO THE CALLING PROGRAM.                        RDDATA
                C                                                                        RDDATA
                C                                                                        RDDATA
                C     COMMON BLOCKS                                                      RDDATA
                C                                                                        RDDATA
ISN C003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                         DECLARE
ISN C004            INTEGER*4  ROWS,COLS                                                 DECLARE
ISN C005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                             DECLARE
ISN C006            INTEGER*4  DISTN,WHTRN                                               DECLARE
ISN C007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                          DECLARE
ISN C008            REAL*8  BUFFER                                                       DECLARE
```

```
ISN C009          INTEGER*4  EOD,CDCOL,BUFNO                                    DECLARE
ISN C010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD          DECLARE
ISN C011          REAL*8   FMT                                                   DECLARE
ISN C012          INTEGER*4  DFORM,DOUT,RCBCD                                    DECLARE
ISN C013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),    DECLARE
                 1    VALUE(40),IVALUE(40),RC                                   DECLARE
ISN C014          REAL*8   PARM,SUBPRM                                           DECLARE
ISN C015          INTEGER*4  RC                                                  DECLARE
ISN C016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                    DECLARE
ISN C017          INTEGER*4  TAIL                                                DECLARE
ISN C018          LOGICAL*4  TESTS,DOALL                                         DECLARE
ISN C019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),        DECLARE
                 1    DATE,TIME,PAGE,LINE                                       DECLARE
ISN C020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME               DECLARE
ISN C021          INTEGER*4  PAGE                                                DECLARE
ISN C022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
ISN C023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE  DECLARE
ISN C024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT            DECLARE
ISN C025          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT            DECLARE
ISN C026          COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                       DECLARE
ISN C027          INTEGER*4  ITEMS,LOCAT,RDNO                                    DECLARE
ISN C028          COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                   DECLARE
ISN C029          INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                         DECLARE
ISN C030          EQUIVALENCE (WORK(1),IWRK(1))                                  DECLARE
ISN C031          REAL*8  BLANK /' '/                                            DECLARE
ISN C032          EQUIVALENCE (BLANK,IBLANK)                                     DECLARE
                  C                                                              RDDATA
                  C     OTHER DATA USED IN THIS SUBROUTINE                       RDDATA
                  C                                                              RDDATA
ISN C033          INTEGER*2  PTR(80), BPTR, EPTR                                 RDDATA
ISN C034          DATA  BPTR /' '/, EPTR /'$'/                                   RDDATA
ISN C035          REAL*8   RCBLBL(2,5) /'BLOCKS', ' ', 'TREATMEN', 'TS',         RDDATA
                 1    'REPLICAT', 'IONS', 'NO CONTR', 'OL', ' ', ' '/           RDDATA
ISN C036          REAL*8  OUTLBL(4,4) /'AS IT CO', 'MES IN', ' ', ' ',           RDDATA
                 1    'AFTER US', 'ER TRANS', 'FORMATIO', 'N', 'SEPARATE',      RDDATA
                 2    'LY BY SA', 'MPLES', ' ', 'AFTER BE', 'ING RANK',         RDDATA
                 3    'ED', ' '/                                                RDDATA
ISN C037          INTEGER*4  LSUB(3), NSUBS                                      RDDATA
ISN C038          REAL*8  TRNLBL(3,2) /'ON THE I', 'NPUT DAT', 'A',              RDDATA
                 1    'AFTER SE', 'PARATION', ' OF DATA'/                       RDDATA
ISN C039          INTEGER*4  CURSMP, CURCNT, CURVAR, CURBLK, CURTRT, CURREP,     RDDATA
                 1    TOTSIZ, TCTITM, TCTIBD                                    RDDATA
                  C                                                              RDDATA
```

```
                C     FORMAT STATEMENTS                                                 RDDATA
                C                                                                       RDDATA
ISN C040          150 FORMAT ('-', T12, 'INPUT OF DATA')                                RDDATA
ISN C041          151 FORMAT ('0', T12, 'THERE IS 1 DATA ITEM PER READ')                RDDATA
ISN C042          152 FORMAT ('0', T12, 'THERE ARE', I3, ' DATA ITEMS PER READ')        RDDATA
ISN C043          153 FORMAT (' ', T12, 'INPUT IS FROM UNIT', I3)                       RDDATA
ISN C044          154 FORMAT ('0', T12, 'DATA IS UNFORMATTED')                          RDDATA
ISN C045          155 FORMAT (' ', T12, 'DATA IS CODED BY ', 6A8)                       RDDATA
ISN C046          156 FORMAT ('-', T12, 'END OF INPUT OF DATA')                         RDDATA
ISN C047          157 FORMAT ('0', T12, 'THE DATA IS WRITTEN OUT ON UNIT',              RDDATA
                     1    I3, 1X, 4A8)                                                RDDATA
ISN C048          158 FORMAT ('-', T12, 'USER TRANSFORMATION IS SPECIFIED ', 3A8)       RDDATA
ISN C049          159 FORMAT (' ')                                                      RDDATA
ISN C050          220 FORMAT ('-', T12, 'IKMS20I  CONVERSION ERROR OCCURRED IN COLUMN', RDDATA
                     1    I3/ ' ', T12, 10A8/ ' ', T12, 80A1)                         RDDATA
ISN C051          221 FORMAT ('-', T12, 'IKMS21I  END OF DATA CARD MISSING.  ASSUMED PRERDDATA'
                    1SENT.')                                                            RDDATA
ISN C052          224 FORMAT ('-', T12, 'IKMS24I  INCORRECTLY LABELLED DATA', 4I5)      RDDATA
ISN C053          225 FORMAT ('-', T12, 'IKMS25I  INPUT SAMPLE SIZES DON''T MATCH SPECIFRDDATA
                    1IED SIZES.  INPUT SIZES USED.')                                    RDDATA
ISN C054          226 FORMAT ('-', T12, 'IKMS26I  INCOMPLETE BLOCKS IN RCB DATA')       RDDATA
ISN C055          227 FORMAT ('-', T12, 'IKMS27I  MISSING REPLICATIONS IN BIBD DATA')   RDDATA
                C                                                                       RDDATA
                C     LIST THE READ SPECIFICATIONS                                      RDDATA
                C                                                                       RDDATA
ISN C056                WKSZE = WKSZE                                                   RDDATA
ISN C057                WRITE (LST,150)                                                 RDDATA
ISN C058                CALL OFLOW(3)                                                   RDDATA
ISN C059                IF (IPR .GT. 1) GO TO 12                                        RDDATA
ISN C061                WRITE (LST,151)                                                 RDDATA
ISN C062                GO TO 13                                                        RDDATA
ISN C063             12 WRITE (LST,152) IPR                                             RDDATA
ISN C064             13 CALL OFLOW(2)                                                   RDDATA
ISN C065                WRITE (LST,153) IPT                                             RDDATA
ISN C066                CALL OFLOW(1)                                                   RDDATA
ISN C067                IF (DFORM .EQ. 1) GO TO 14                                      RDDATA
                C     CALL SUBROUTINE TO READ THE USER'S FORMAT                         RDDATA
ISN C069                CALL RDFMT                                                      RDDATA
ISN C070                IF (RC .NE. 0) RETURN                                           RDDATA
ISN C072                GO TO 15                                                        RDDATA
                C     NOTE THAT THE DATA IS UNFORMATTED                                 RDDATA
ISN C073             14 WRITE (LST,154)                                                 RDDATA
ISN C074                CALL OFLOW(2)                                                   RDDATA
```

```
ISN C075          15 NSUBS = 0                                                  RDDATA
ISN C076             IF (TYPE .LT. 7) GO TO 41                                  RDDATA
                C                                                               RDDATA
                C    RCB OR BIBD DATA                                           RDDATA
                C                                                               RDDATA
ISN C078             LSUB(2) = 5                                                RDDATA
ISN C079             LSUB(3) = 5                                                RDDATA
ISN C080             GO TO (31, 32, 33, 34, 35, 36, 37, 39, 39), RCBCD          RDDATA
                C                                                               RDDATA
                C    SET UP LABELLING SUBSCRIPTS FOR PRINTING MESSAGE           RDDATA
                C                                                               RDDATA
ISN C081          31 LSUB(1) = 1                                                RDDATA
ISN C082             NSUBS = 1                                                  RDDATA
ISN C083             GO TO 40                                                   RDDATA
                C                                                               RDDATA
ISN C084          32 LSUB(1) = 2                                                RDDATA
ISN C085             NSUBS = 1                                                  RDDATA
ISN C086             GO TO 40                                                   RDDATA
                C                                                               RDDATA
ISN C087          33 LSUB(1) = 1                                                RDDATA
ISN C088             LSUB(2) = 2                                                RDDATA
ISN C089             NSUBS = 2                                                  RDDATA
ISN C09C             GO TO 40                                                   RDDATA
                C                                                               RDDATA
ISN C091          34 LSUB(1) = 3                                                RDDATA
ISN C092             NSUBS = 1                                                  RDDATA
ISN C093             GO TO 40                                                   RDDATA
                C                                                               RDDATA
ISN C094          35 LSUB(1) = 1                                                RDDATA
ISN C095             LSUB(2) = 3                                                RDDATA
ISN C096             NSUBS = 2                                                  RDDATA
ISN C097             GO TO 40                                                   RDDATA
                C                                                               RDDATA
ISN C098          36 LSUB(1) = 2                                                RDDATA
ISN C099             LSUB(2) = 3                                                RDDATA
ISN C100             NSUBS = 2                                                  RDDATA
ISN C101             GO TO 40                                                   RDDATA
                C                                                               RDDATA
ISN C102          37 LSUB(1) = 1                                                RDDATA
ISN C103             LSUB(2) = 2                                                RDDATA
ISN C104             LSUB(3) = 3                                                RDDATA
ISN C105             NSUBS = 3                                                  RDDATA
ISN C106             GO TO 40                                                   RDDATA
```

```
                C                                                                       RDDATA
ISN C107           39 LSUB(1) = 4                                                       RDDATA
ISN C108              NSUBS = 0                                                         RDDATA
                C                                                                       RDDATA
                C     WRITE AN APPROPRIATE MESSAGE                                      RDDATA
                C                                                                       RDDATA
ISN C109           4C WRITE (LST,155) (RCBLBL(1,LSUB(I)), RCBLBL(2,LSUB(I)), I = 1, 3)  RDDATA
ISN C110              CALL OFLOW (1)                                                    RDDATA
                C                                                                       RDDATA
                C     CONTINUE PRINTING MESSAGES                                        RDDATA
                C                                                                       RDDATA
ISN C111           41 IF (OUT .EQ. 0) GO TO 51                                          RDDATA
ISN C113              WRITE (LST,157) OUT, (OUTLBL(I,DOUT+1), I = 1, 4)                 RDDATA
ISN C114              CALL OFLOW(2)                                                     RDDATA
                C                                                                       RDDATA
ISN C115           51 I = 1                                                             RDDATA
ISN C116              IF (WHTRN) 53, 54, 52                                             RDDATA
ISN C117           52 I = 2                                                             RDDATA
ISN C118           53 WRITE (LST,158) (TRNLBL(J,I), J = 1, 3)                           RDDATA
ISN C119              CALL OFLOW(3)                                                     RDDATA
                C                                                                       RDDATA
                C     INITIALIZE INPUT COUNTERS AND POINTERS                            RDDATA
                C                                                                       RDDATA
ISN C120           54 WRITE (LST,159)                                                   RDDATA
ISN C121              CALL OFLOW(1)                                                     RDDATA
ISN C122              IF (OUT .NE. 0 .AND. OUT .NE. LST) REWIND OUT                     RDDATA
ISN C124              REWIND UT1                                                        RDDATA
ISN C125              EOD = 0                                                           RDDATA
ISN C126              TOTITM = 0                                                        RDDATA
ISN C127              IF (TYPE .GT. 3) GO TO 57                                         RDDATA
                C                                                                       RDDATA
                C     1 OR MORE RANDOM SAMPLES                                          RDDATA
                C                                                                       RDDATA
ISN C129           55 TOTSIZ = 0                                                        RDDATA
ISN C130              DO 56 I = 1, SMPLS                                                RDDATA
ISN C131              TOTSIZ = TOTSIZ + SSIZE(I)                                        RDDATA
ISN C132           56 IWRK(I) = 0                                                       RDDATA
ISN C133              CURSMP = 1                                                        RDDATA
ISN C134              CURVAR = 1                                                        RDDATA
ISN C135              CURCNT = 0                                                        RDDATA
ISN C136              GO TO 65                                                          RDDATA
                C                                                                       RDDATA
ISN C137           57 IF (TYPE - 6) 58, 55, 60                                          RDDATA
```

```
                C                                                                       RDDATA
                C     BIVARIATE OR MULTIVARIATE DATA                                    RDDATA
                C                                                                       RDDATA
ISN C138           58 TOTSIZ = SSIZE(1) * VARITE                                        RDDATA
ISN C139              DO 59 I = 1, VARITE                                               RDDATA
ISN C140              SSIZE(I) = SSIZE(1)                                               RDDATA
ISN 0141           59 IWRK(I) = 0                                                       RDDATA
ISN C142              CURVAR = 1                                                        RDDATA
ISN C143              CURSMP = 1                                                        RDDATA
ISN C144              CURCNT = 0                                                        RDDATA
ISN C145              GO TO 65                                                          RDDATA
                C                                                                       RDDATA
                C     BCD OR BIBD DATA                                                  RDDATA
                C                                                                       RDDATA
ISN C146           60 CURBLK = 1                                                        RDDATA
ISN C147              CURTRT = 1                                                        RDDATA
ISN C148              CURREP = 1                                                        RDDATA
ISN C149              IF (REP .EQ. 0) REP = 1                                           RDDATA
ISN C151              TOTSIZ = BLK * TRT * REP                                          RDDATA
ISN C152              TOTIBD = BLK * TRTB                                               RDDATA
ISN C153              CURCNT = 0                                                        RDDATA
ISN C154              DO 61 I = 1, TOTSIZ                                               RDDATA
ISN C155           61 IWRK(I) = 0                                                       RDDATA
ISN 0156              GO TO 65                                                          RDDATA
                C                                                                       RDDATA
ISN C157           65 RDNO = 0                                                          RDDATA
ISN 0158              IF (IPT .NE. RDR .AND. DFORM .EQ. 1) REWIND IPT                   RDDATA
                C                                                                       RDDATA
                C     CHECK HOW TO READ THE DATA AND READ IT                            RDDATA
                C                                                                       RDDATA
ISN 0160           71 IF (NSUBS .EQ. 0) GO TO 73                                        RDDATA
                C     DATA HAS LOCATION PARAMETERS WITH IT                              RDDATA
ISN 0162              IF (DFORM .NE. 1) GO TO 72                                        RDDATA
                C     UNFORMATTED INPUT WITH SUBSCRIPTS                                 RDDATA
ISN C164              READ (IPT,END=99) (SUBS(I), I = 1, NSUBS), (DATA(I), I = 1, IPR)  RDDATA
ISN C165              GO TO 81                                                          RDDATA
                C     FORMATTED WITH SUBSCRIPTS                                         RDDATA
ISN C166           72 READ (IPT,FMT,END=99) (SUBS(I), I = 1, NSUBS),                    RDDATA
                  1     (DATA(I), I = 1, IPR)                                         RDDATA
ISN 0167              IF (EOD) 98, 81, 97                                               RDDATA
                C                                                                       RDDATA
                C     DATA DOES NOT HAVE ANY LOCATION PARAMETERS                        RDDATA
                C                                                                       RDDATA
```

```
ISN C168        73 IF (OFORM .NE. 1) GO TO 74                                    RDDATA
                C     UNFORMATTED DATA WITH NO SUBSCRIPTS                        RDDATA
ISN C170              READ (IPT,END=99) (DATA(I), I = 1, IPR)                    RDDATA
ISN C171              GO TO 81                                                   RDDATA
                C     FORMATTED DATA WITHOUT SUBSCRIPTS                          RDDATA
ISN C172        74 READ (IPT,FMT,END=99) (DATA(I), I = 1, IPR)                   RDDATA
ISN C173              IF (EOD) 98, 81, 97                                        RDDATA
                C                                                                RDDATA
                C     SIGNAL PHYSICAL END OF DATA                                RDDATA
                C                                                                RDDATA
ISN C174        99 IF (IPT .EQ. RDR) EOD = 2                                     RDDATA
ISN C176              IF (OFORM .EQ. 1) GO TO 97                                 RDDATA
                C     FORGIVE HIM FOR FORGETTING THE 'END OF DATA' CARD          RDDATA
ISN C178              WRITE (LST,221)                                            RDDATA
ISN C179              CALL OFLOW(3)                                              RDDATA
ISN 0180              GO TO 97                                                   RDDATA
                C                                                                RDDATA
                C     REAL LIVE DATA CONVERSION ERROR                            RDDATA
                C                                                                RDDATA
ISN C181        98 DO 981 I = 1, 80                                              RDDATA
ISN C182       981 PTR(I) = BPTR                                                 RDDATA
ISN C183              PTR(CDCOL) = EPTR                                          RDDATA
ISN C184              WRITE (LST,220) CDCOL, (BUFFER(I,BUFNO), I = 1, 10), PTR   RDDATA
ISN C185              RC = 12                                                    RDDATA
                C     ENDFILE USER OUTPUT AT POINT OF CONVERSION ERROR           RDDATA
ISN 0186              IF (OUT .NE. 0 .AND. OUT .NE. LST .AND. DOUT .LE. 1) ENDFILE OUT RDDATA
ISN C188              GO TO 4001                                                 RDDATA
                C                                                                RDDATA
                C     NORMAL END OF DATA                                         RDDATA
                C                                                                RDDATA
ISN 0189        97 IF (EOD .LT. 3) GO TO 971                                     RDDATA
ISN C191              WRITE (LST,221)                                            RDDATA
ISN C192              CALL OFLOW(3)                                              RDDATA
ISN C193       971 IF (OUT .NE. 0 .AND. DOUT .LE. 1 .AND. OUT .NE. LST) ENDFILE OUT RDDATA
ISN C195              ENDFILE UT1                                                RDDATA
ISN C196              GO TO 3001                                                 RDDATA
                C                                                                RDDATA
                C     PROCESS THE DATA THE WAY IT BELONGS                        RDDATA
                C                                                                RDDATA
ISN 0197        81 ITEMS = IPR                                                   RDDATA
ISN C198              RDNO = RDNO + 1                                            RDDATA
ISN C199              LOCAT(1) = SUBS(1)                                         RDDATA
ISN C200              LOCAT(2) = SUBS(2)                                         RDDATA
```

```
ISN C201            LOCAT(3) = SUBS(3)                                          RDDATA
ISN C202            LOCAT(4) = NSUBS                                            RDDATA
ISN C203            DO 82 I = 1, ITEMS                                          RDDATA
ISN C204         82 X(I) = DATA(I)                                              RDDATA
                 C                                                              RDDATA
                 C      CHECK FOR RAW DATA OUT                                  RDDATA
                 C                                                              RDDATA
ISN C205            IF (OUT .NE. 0 .AND. COUT .EQ. 0) CALL WRDATA               RDDATA
                 C                                                              RDDATA
                 C      CHECK FOR RAW TRANSFORMATION                            RDDATA
                 C                                                              RDDATA
ISN C207            IF (WHTRN .EQ. -1) CALL TRANS                               RDDATA
                 C                                                              RDDATA
                 C      CHECK FOR DELETED ITEM                                  RDDATA
                 C                                                              RDDATA
ISN 0209            IF (ITEMS .LE. 0) GO TO 2001                                RDDATA
                 C                                                              RDDATA
                 C      CHECK FOR OUTPUT OF TRANSFORMED RAW DATA                RDDATA
                 C                                                              RDDATA
ISN 0211            IF (OUT .NE. 0 .AND. COUT .EQ. 1) CALL WRDATA               RDDATA
                 C                                                              RDDATA
                 C      COUNT ITEMS AND GO TO PROCESS DATA INTO THE PROPER SAMPLE RDDATA
ISN C213            TOTITM = TOTITM + ITEMS                                     RDDATA
ISN C214            GO TO (1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008), TYPE RDDATA
                 C                                                              RDDATA
                 C      1 OR MORE RANDOM SAMPLES                                RDDATA
                 C                                                              RDDATA
ISN C215         1001 DO 1041 IENT = 1, ITEMS                                   RDDATA
                 C      CHECK IF IT SHOULD GO INTO THE CURRENT SAMPLE           RDDATA
ISN C216         1011 IF (CURCNT .LT. IPS(CURSMP)) GO TO 1031                   RDDATA
ISN C218         1021 CURSMP = CURSMP + 1                                       RDDATA
ISN C219            IF (CURSMP .GT. SMPLS) CURSMP = 1                           RDDATA
ISN C221            CURCNT = 0                                                  RDDATA
ISN C222            IF (SSIZE(CURSMP) .EQ. 0) GO TO 1011                        RDDATA
                 C      PROCESS THIS ITEM IN THIS SAMPLE                        RDDATA
ISN C224         1031 CURCNT = CURCNT + 1                                       RDDATA
ISN C225            IWRK(CURSMP) = IWRK(CURSMP) + 1                             RDDATA
ISN C226            WRITE (UT1) CURSMP,CURVAR,IWRK(CURSMP),RDNO,IENT,X(IENT)    RDDATA
                 C      SAMPLE NUMBER, VARIATE NUMBER, ITEM NUMBER WITHIN SAMPLE NUMBER RDDATA
                 C      READ NUMBER, ITEM WITHIN READ NUMBER, DATA VALUE        RDDATA
ISN C227         1041 CONTINUE                                                  RDDATA
ISN 0228            GO TO 2001                                                  RDDATA
                 C                                                              RDDATA
```

```
ISN 0229      1002 GO TO 1001                                                  RDDATA
ISN 0230      1003 GO TO 1001                                                  RDDATA
              C                                                                RDDATA
              C     BIVARIATE OR MULTIVARIATE SAMPLE                           RDDATA
              C                                                                RDDATA
ISN 0231      1004 DO 1044 IENT = 1, ITEMS                                     RDDATA
              C     SEE 1 OR MORE RANDOM SAMPLE FOR COMMENTS                   RDDATA
ISN 0232      1014 IF (CURCNT .LT. IPS(CURVAR)) GO TO 1034                     RDDATA
ISN 0234      1024 CURVAR = CURVAR + 1                                         RDDATA
ISN 0235           IF (CURVAR .GT. VARITE) CURVAR = 1                         RDDATA
ISN 0237           CURCNT = 0                                                 RDDATA
ISN 0238           IF (SSIZE(CURVAR) .EQ. 0) GO TO 1014                       RDDATA
              C                                                                RDDATA
ISN 0240      1034 CURCNT = CURCNT + 1                                         RDDATA
ISN 0241           IWRK(CURVAR) = IWRK(CURVAR) + 1                            RDDATA
ISN 0242           WRITE (UT1) CURSMP,CURVAR,IWRK(CURVAR),RDNO,IENT,X(IENT)   RDDATA
              C     SAMPLE NUMBER, VARIATE NUMBER, ITEM WITHIN VARIATE NUMBER  RDDATA
              C     READ NUMBER, ITEM WITHIN READ NUMBER, DATA VALUE           RDDATA
ISN 0243      1044 CONTINUE                                                    RDDATA
ISN 0244           GO TO 2001                                                 RDDATA
              C                                                                RDDATA
ISN 0245      1005 GO TO 1004                                                  RDDATA
              C                                                                RDDATA
              C     2 MATCHED SAMPLES - TREAT AS 2 RANDOM SAMPLES              RDDATA
              C                                                                RDDATA
ISN 0246      1006 GO TO 1001                                                  RDDATA
              C                                                                RDDATA
              C     RCB (AND BIBD) TYPE DATA WITH OR WITHOUT LOCATION PARAMETERS RDDATA
              C                                                                RDDATA
ISN 0247      1007 GO TO (1017, 1027, 1037, 1047, 1057, 1067, 1077, 1097, 1097), RDDATA
                  1    RCBCD                                                   RDDATA
              C                                                                RDDATA
              C     SET LOCATION PARAMETERS AS SPECIFIED BY THE USER           RDDATA
              C                                                                RDDATA
ISN 0248      1017 CURBLK = LOCAT(1)                                           RDDATA
ISN 0249           GO TO 1107                                                 RDDATA
              C                                                                RDDATA
ISN 0250      1027 CURTRT = LOCAT(1)                                           RDDATA
ISN 0251           GO TO 1107                                                 RDDATA
              C                                                                RDDATA
ISN 0252      1037 CURBLK = LOCAT(1)                                           RDDATA
ISN 0253           CURTRT = LOCAT(2)                                          RDDATA
ISN 0254           GO TO 1107                                                 RDDATA
```

```
                C                                                                       RDDATA
ISN 0255         1047 CURREP = LOCAT(1)                                                 RDDATA
ISN 0256              GO TO 1107                                                        RDDATA
                C                                                                       RDDATA
ISN 0257         1057 CURBLK = LOCAT(1)                                                 RDDATA
ISN 0258              CURREP = LOCAT(2)                                                 RDDATA
ISN 0259              GO TO 1107                                                        RDDATA
                C                                                                       RDDATA
ISN 0260         1067 CURTRT = LOCAT(1)                                                 RDDATA
ISN 0261              CURREP = LOCAT(2)                                                 RDDATA
ISN 0262              GO TO 1107                                                        RDDATA
                C                                                                       RDDATA
ISN 0263         1077 CURBLK = LOCAT(1)                                                 RDDATA
ISN 0264              CURTRT = LOCAT(2)                                                 RDDATA
ISN 0265              CURREP = LOCAT(3)                                                 RDDATA
ISN 0266              GO TO 1107                                                        RDDATA
                C                                                                       RDDATA
ISN 0267         1097 GO TO 1107                                                        RDDATA
                C                                                                       RDDATA
                C     CHECK LIMITS ON THE LOCATIONS                                     RDDATA
                C                                                                       RDDATA
ISN 0268         1107 IF (CURBLK .GT. BLK) CURBLK = 1                                   RDDATA
ISN 0270              IF (CURTRT .GT. TRT) CURTRT = 1                                   RDDATA
ISN 0272              IF (CURREP .GT. REP) CURREP = 1                                   RDDATA
                C                                                                       RDDATA
                C     FIND THE HOLES AND PLUG THEM WITH THE DATA                        RDDATA
                C                                                                       RDDATA
ISN 0274              DO 1197 IENT = 1, ITEMS                                           RDDATA
ISN 0275              CURCNT = CURCNT + 1                                               RDDATA
                C     CALCULATE THE LOCATION OF THIS ITEM                               RDDATA
ISN 0276              LOC = BLK * TRT * (CURREP - 1) + BLK * (CURTRT - 1) + CURBLK      RDDATA
                C     CHECK FOR USED LOCATION AND SET USED IF NOT                       RDDATA
ISN 0277              IF (IWRK(LOC) .NE. 0) GO TO 2224                                  RDDATA
ISN 0279              IWRK(LOC) = 1                                                     RDDATA
ISN 0280              WRITE (UT1) CURBLK,CURTRT,CURREP,RDNO,IENT,X(IENT)                RDDATA
                C     BLOCK NUMBER, TRESTMENT NUMBER, REPLICATION NUMBER,               RDDATA
                C     READ NUMBER, ITEM WITHIN READ NUMBER, DATA VALUE                  RDDATA
                C                                                                       RDDATA
                C     GO TO ADVANCE SUBSCRIPTS ACCCRDING TO USER DESIRES                RDDATA
                C                                                                       RDDATA
ISN 0281              GO TO (1217, 1227, 1237, 1247, 1257, 1267, 1277, 1297, 1297),     RDDATA
                     1    RC8CD                                                         RDDATA
                C                                                                       RDDATA
```

```
ISN C282      1217 CURREP = CURREP + 1                                  RDDATA
ISN C283           IF (CURREP .LE. REP) GO TC 1197                      RDDATA
ISN C285           CURREP = 1                                           RDDATA
ISN C286           CURTRT = CURTRT + 1                                  RDDATA
ISN C287           IF (CURTRT .GT. TRT) CURTRT = 1                      RDDATA
ISN C289           GO TO 1197                                           RDDATA
              C                                                         RDDATA
ISN C290      1227 CURREP = CURREP + 1                                  RDDATA
ISN C291           IF (CURREP .LE. REP) GO TC 1197                      RDDATA
ISN C293           CURREP = 1                                           RDDATA
ISN C294           CURBLK = CURBLK + 1                                  RDDATA
ISN C295           IF (CURBLK .GT. BLK) CURBLK = 1                      RDDATA
ISN C297           GO TO 1197                                           RDDATA
              C                                                         RDDATA
ISN C298      1237 CURREP = CURREP + 1                                  RDDATA
ISN C299           IF (CURREP .GT. REP) CURREP = 1                      RDDATA
ISN C301           GO TO 1197                                           RDDATA
              C                                                         RDDATA
ISN C302      1247 CURTRT = CURTRT + 1                                  RDDATA
ISN C303           IF (CURTRT .LE. TRT) GO TC 1197                      RDDATA
ISN C305           CURTRT = 1                                           RDDATA
ISN C306           CURBLK = CURBLK + 1                                  RDDATA
ISN C307           IF (CURBLK .GT. BLK) CURBLK = 1                      RDDATA
ISN C309           GO TO 1197                                           RDDATA
              C                                                         RDDATA
ISN C310      1257 CURTRT = CURTRT + 1                                  RDDATA
ISN C311           IF (CURTRT .GT. TRT) CURTRT = 1                      RDDATA
ISN C313           GO TO 1197                                           RDDATA
              C                                                         RDDATA
ISN C314      1267 CURBLK = CURBLK + 1                                  RDDATA
ISN C315           IF (CURBLK .GT. BLK) CURBLK = 1                      RDDATA
ISN C317           GO TO 1197                                           RDDATA
              C                                                         RDDATA
ISN C318      1277 GO TO 1197                                           RDDATA
              C                                                         RDDATA
ISN C319      1297 CURREP = CURREP + 1                                  RDDATA
ISN C320           IF (CURREP .LE. REP) GO TC 1197                      RDDATA
ISN C322           CURREP = 1                                           RDDATA
ISN C323           CURTRT = CURTRT + 1                                  RDDATA
ISN C324           IF (CURTRT .LE. TRT) GO TO 1197                      RDDATA
ISN C326           CURTRT = 1                                           RDDATA
ISN C327           CURBLK = CURBLK + 1                                  RDDATA
ISN C328           IF (CURBLK .GT. BLK) CURBLK = 1                      RDDATA
```

```
                C                                                                RDDATA
ISN C330         1197 CONTINUE                                                   RDDATA
                C                                                                RDDATA
ISN C331              GO TO 2001                                                 RDDATA
                C                                                                RDDATA
ISN C332         1008 GO TO 1007                                                 RDDATA
                C                                                                RDDATA
                C     END OF PROCESSING THIS ITEM                                RDDATA
                C                                                                RDDATA
ISN C333         2001 GO TO 71                                                   RDDATA
                C                                                                RDDATA
                C     END OF INPUT DATA - CHECK FOR ERRORS                       RDDATA
                C                                                                RDDATA
ISN C334         3001 IF (ITEMS .LE. 0) ITEMS = IPR                              RDDATA
ISN C336              IF (TOTSIZ .EQ. 0) GO TO 3005                              RDDATA
ISN C338              IF (TOTITM .EQ. TOTSIZ) GO TO 3007                         RDDATA
ISN C340              IF (TYPE - 7) 3002, 3003, 3004                             RDDATA
                C                                                                RDDATA
                C     SAMPLE SIZES DON'T MATCH SO TELL USER                      RDDATA
                C                                                                RDDATA
ISN C341         3002 WRITE (LST,225)                                            RDDATA
ISN C342              CALL OFLOW(3)                                              RDDATA
ISN C343              GO TO 3005                                                 RDDATA
                C                                                                RDDATA
                C     RCB DATA INCOMPLETE                                        RDDATA
                C                                                                RDDATA
ISN C344         3003 WRITE (LST,226)                                            RDDATA
ISN C345              RC = 12                                                    RDDATA
ISN C346              GO TO 4001                                                 RDDATA
                C                                                                RDDATA
                C     CHECK ITEMS IN A BIBD TYPE DATA                            RDDATA
                C                                                                RDDATA
ISN C347         3004 IF (TOTITM .EQ. TOTIBD) GO TO 3007                         RDDATA
ISN C349              WRITE (LST,227)                                            RDDATA
ISN C350              RC = 12                                                    RDDATA
ISN C351              GO TO 4001                                                 RDDATA
                C                                                                RDDATA
                C     SET THE SAMPLE SIZES AS READ                               RDDATA
                C                                                                RDDATA
ISN C352         3005 J = SMPLS * VARITE                                         RDDATA
ISN C353              DO 3006 I = 1, J                                           RDDATA
ISN C354         3006 SSIZE(I) = IWRK(I)                                         RDDATA
                C                                                                RDDATA
```

```
                C     DATA IS TENATIVELY OK, SO RETURN CODE IS ZERO                    RDDATA
                C                                                                      RDDATA
   ISN C355      3007 RC = 0                                                           RDDATA
   ISN C356           GO TO 4002                                                      RDDATA
                C                                                                      RDDATA
                C     CHECK OVERFLOW BECAUSE ERROR MESSAGE WAS PRINTED                 RDDATA
                C                                                                      RDDATA
   ISN C357      4001 CALL OFLOW(3)                                                    RDDATA
                C                                                                      RDDATA
                C     WRITE END MESSAGE AND RETURN TO CALLER                           RDDATA
                C                                                                      RDDATA
   ISN C358      4002 WRITE (LST,156)                                                  RDDATA
   ISN C359           CALL OFLOW(3)                                                   RDDATA
                C                                                                      RDDATA
   ISN C360           RETURN                                                          RDDATA
                C                                                                      RDDATA
                C     DUPLICATE RCB DATA ITEM                                          RDDATA
                C                                                                      RDDATA
   ISN C361      2224 WRITE (LST,224) RDNO, CURBLK, CURTRT, CURREP                     RDDATA
   ISN C362           RC = 12                                                         RDDATA
   ISN C363           GO TO 4001                                                      RDDATA
                C                                                                      RDDATA
   ISN C364           END                                                             RDDATA

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     363 ,PROGRAM SIZE =     5808

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE RDFMT                                                    RDFMT
                    C                                                                   RDFMT
                    C     THIS SUBRCUTINE READS THE USER FORMAT STATEMENT ONE CARD AT   RDFMT
                    C     A TIME UNTIL ALL PARENTHESES BALANCE OR UNTIL 10 CARDS ARE READ. RDFMT
                    C     IT CALLS THE ROUTINE 'CNTFMT' TO PERFORM THE ACTUAL PARENTHESES RDFMT
                    C     COUNT.  THAT ROUTINE IS WRITTEN IN ASSEMBLY LANGUAGE TO       RDFMT
                    C     FACILITATE THE ACTUAL COUNTING AND ALLOW THE FORMAT TO BE READ RDFMT
                    C     INTO A REAL*8 ARRAY, WHICH IS HOW IT IS ACTUALLY REFERENCED.  RDFMT
                    C                                                                   RDFMT
                    C                                                                   RDFMT
                    C     COMMON BLOCK DECLARATIONS                                     RDFMT
                    C                                                                   RDFMT
ISN 0003            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                         RDFMT
ISN 0004            REAL*8  BUFFER                                                      RDFMT
ISN 0005            INTEGER*4  EOD,CDCOL,BUFNO                                          RDFMT
ISN 0006            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD               RDFMT
ISN 0007            REAL*8   FMT                                                        RDFMT
ISN 0008            INTEGER*4  DFORM,DOUT,RCBCD                                         RDFMT
ISN 0009            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),         RDFMT
                   1    VALUE(40),IVALUE(40),RC                                         RDFMT
ISN 0010            REAL*8   PARM,SUBPRM                                                RDFMT
ISN 0011            INTEGER*4  RC                                                       RDFMT
ISN 0012            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),             RDFMT
                   1    DATE,TIME,PAGE,LINE                                             RDFMT
ISN 0013            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                    RDFMT
ISN 0014            INTEGER*4  PAGE                                                     RDFMT
ISN 0015            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                 RDFMT
ISN 0016            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                 RDFMT
ISN 0017            REAL*8 BLANK /' '/                                                  RDFMT
ISN 0018            EQUIVALENCE (BLANK,IBLANK)                                          RDFMT
                    C                                                                   RDFMT
                    C     FORMAT STATEMENTS                                             RDFMT
                    C                                                                   RDFMT
ISN 0019          101 FORMAT (10A8)                                                     RDFMT
ISN 0020          106 FORMAT ('0', T12, 'INPUT FORMAT STATEMENT'/                       RDFMT
                   1    (' ', T12, 10A8))                                              RDFMT
ISN 0021          206 FORMAT ('-', T12, 'IKMS06I  EXCESSIVE RIGHT PARENTHESES')        RDFMT
ISN 0022          207 FORMAT ('-', T12, 'IKMS07I  EXCESSIVE LEFT PARENTHESES')         RDFMT
ISN 0023          223 FORMAT ('-', T12, 'IKMS23I  END OF FILE WHILE READING FORMAT STATERDFMT
                   1MENT')                                                              RDFMT
                    C                                                                   RDFMT
```

```
                C     BLANK OUT THE FORMAT                                          RDFMT
                C                                                                   RDFMT
ISN C024              DO 1 I = 1, 100                                               RDFMT
ISN C025            1 FMT(I) = BLANK                                                RDFMT
ISN C026              RC = 0                                                        RDFMT
                C                                                                   RDFMT
                C     READ THE FORMAT STATEMENT 1 CARD AT A TIME                    RDFMT
                C                                                                   RDFMT
ISN C027              DO 2 I = 1, 91, 10                                            RDFMT
ISN C028              K = I + 9                                                     RDFMT
ISN C029              READ (RDR,101,END=9) (FMT(J), J = I, K)                       RDFMT
ISN C030              CALL CNTFMT (FMT,K)                                           RDFMT
ISN C031              IF (K) 3, 3, 2                                                RDFMT
ISN C032            2 CONTINUE                                                      RDFMT
                C                                                                   RDFMT
                C     DETERMINE THE LENGTH OF THE FORMAT STATEMENT                  RDFMT
                C                                                                   RDFMT
ISN C033            3 DO 4 ISZ = 1, 100                                             RDFMT
ISN C034              IF (FMT(ISZ) .EQ. BLANK) GO TO 10                             RDFMT
ISN C036            4 CONTINUE                                                      RDFMT
ISN C037              ISZ = 101                                                     RDFMT
ISN C038           10 ISZ = ISZ - 1                                                 RDFMT
                C     WRITE OUT THE FORMAT STATEMENT                                RDFMT
ISN C039              WRITE (LST,106) (FMT(I), I = 1, ISZ)                          RDFMT
ISN C040              ILN = 2 + (ISZ+9)/10                                          RDFMT
ISN C041              CALL OFLOW (ILN)                                              RDFMT
                C                                                                   RDFMT
                C     CHECK FOR ERRORS                                              RDFMT
                C                                                                   RDFMT
ISN C042              IF (K) 5, 8, 7                                                RDFMT
                C                                                                   RDFMT
ISN C043            5 WRITE (LST,206)                                               RDFMT
ISN C044              RC = 8                                                        RDFMT
                C                                                                   RDFMT
                C     RETURN TO CALLING PROGRAM                                     RDFMT
                C                                                                   RDFMT
ISN C045            6 CALL OFLOW(3)                                                 RDFMT
                C                                                                   RDFMT
ISN C046            8 RETURN                                                        RDFMT
                C                                                                   RDFMT
ISN C047            7 WRITE (LST,207)                                               RDFMT
ISN C048              RC = 8                                                        RDFMT
ISN C049              GO TO 6                                                       RDFMT
```

```
                C                                                                RDFMT
                C     END OF FILE WHILE READING THE FORMAT                       RDFMT
                C                                                                RDFMT
   ISN C050          9 WRITE (LST,223)                                           RDFMT
   ISN C051            RC = 12                                                   RDFMT
   ISN C052            GO TO 6                                                   RDFMT
                C                                                                RDFMT
   ISN C053            END                                                       RDFMT

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      52 ,PROGRAM SIZE =      984

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002            SUBROUTINE RDPARM                                                RDPARM
                C                                                                     RDPARM
                C     THIS SUBROUTINE PERFORMS THE INITIALIZATION OF ALL PARAMETER     RDPARM
                C     INFORMATION AND SUBSEQUENT READING OF PARAMETER CARDS            RDPARM
                C     TO FILL IN THE PARAMETER INFORMATION AS SPECIFIED BY THE USER.   RDPARM
                C     THIS SUBROUTINE DOES NOT CHECK FOR CONFLICTING OR AMBIGUOUS      RDPARM
                C     PARAMETER SPECIFICATIONS, BUT MERELY CHECKS THE VALIDITY OF      RDPARM
                C     SPECIFICATIONS WITHIN THE CONTEXT OF THE PARAMETER CARD ON       RDPARM
                C     WHICH A PARTICULAR ITEM APPEARS.                                 RDPARM
                C                                                                     RDPARM
                C     ALL DATA PASSED TO AND FROM THIS SUBROUTINE, AS WELL AS THAT     RDPARM
                C     PASSED TO LOWER LEVEL ROUTINES, ARE CARRIED IN COMMON BLOCKS,    RDPARM
                C     TO AVIOD UNNECESSARILY LONG PARAMETER LISTS IN CALL STATEMENTS.  RDPARM
                C                                                                     RDPARM
                C     THIS SUBROUTINE CALLS THE SUBROUTINE 'DECODE', WHICH ANALYZES    RDPARM
                C     THE PARAMETERS ON A STATEMENT AND BREAKS THEM UP FOR EASY        RDPARM
                C     DIGESTION BY FORTRAN.  THAT SUBROUTINE IS WRITTEN IN ASSEMBLY    RDPARM
                C     LANGUAGE, AND HENCE IS MACHINE (S/360) DEPENDENT.                RDPARM
                C                                                                     RDPARM
                C                                                                     RDPARM
                C     COMMON DECLARATIONS                                              RDPARM
                C                                                                     RDPARM
 ISN 0003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                     DECLARE
 ISN 0004            INTEGER*4  ROWS,COLS                                             DECLARE
 ISN 0005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                         DECLARE
 ISN 0006            INTEGER*4  DISTN,WHTRN                                           DECLARE
 ISN 0007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                      DECLARE
 ISN 0008            REAL*8  BUFFER                                                   DECLARE
 ISN 0009            INTEGER*4  EOD,CDCOL,BUFNO                                       DECLARE
 ISN 0010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD            DECLARE
 ISN 0011            REAL*8   FMT                                                     DECLARE
 ISN 0012            INTEGER*4  DFORM,DOUT,RCBCD                                      DECLARE
 ISN 0013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),      DECLARE
                    1    VALUE(40),IVALUE(40),RC                                      DECLARE
 ISN 0014            REAL*8   PARM,SUBPRM                                             DECLARE
 ISN 0015            INTEGER*4  RC                                                    DECLARE
 ISN 0016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                      DECLARE
 ISN 0017            INTEGER*4  TAIL                                                  DECLARE
 ISN 0018            LOGICAL*4  TESTS,DOALL                                           DECLARE
 ISN 0019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),          DECLARE
                    1    DATE,TIME,PAGE,LINE                                          DECLARE
```

```
ISN C020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                    DECLARE
ISN C021          INTEGER*4  PAGE                                                     DECLARE
ISN C022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)  DECLARE
ISN C023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE      DECLARE
ISN C024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C025          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C026          COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                           DECLARE
ISN C027          INTEGER*4  ITEMS,LOCAT,RDNO                                        DECLARE
ISN C028          COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                       DECLARE
ISN C029          INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                             DECLARE
ISN C030          EQUIVALENCE (WORK(1),IWRK(1))                                      DECLARE
ISN C031          REAL*8  BLANK /' '/                                                DECLARE
ISN C032          EQUIVALENCE (BLANK,IBLANK)                                         DECLARE
                C                                                                    RDPARM
                C     FORMAT DECLARATIONS                                            RDPARM
                C                                                                    RDPARM
ISN C033          1C1 FORMAT (10A8)                                                  RDPARM
ISN C034          1C3 FORMAT ('-', T12, 'PARAMETER ANALYSIS')                        RDPARM
ISN C035          1C4 FORMAT ('0', T12, 10A8)                                        RDPARM
ISN C036          1C5 FORMAT ('-', T12, 'NO PARAMETER SYNTAX ERRORS DETECTED.')      RDPARM
ISN C037          20C FORMAT ('-', T12, 'IKMS00I  END OF IKMSTAT RUN')               RDPARM
ISN C038          2C1 FORMAT (' ', T12, 'IKMS01I  INVALID PARAMETER CARD.')          RDPARM
ISN C039          2C2 FORMAT (' ', T12, 'IKMS02I  INVALID SPECIFICATION IN PARAMETER', RDPARM
                     1    I3)                                                       RDPARM
ISN C040          203 FORMAT ('-', T12, 'IKMS03I  END OF FILE ENCOUNTERED WHILE READING RDPARM
                     1PARAMETERS.')                                                  RDPARM
ISN C041          204 FORMAT ('-', T12, 'IKMS04I  1 PARAMETER SYNTAX ERROR DETECTED.') RDPARM
ISN C042          2C5 FORMAT ('-', T12, 'IKMS05I', I4,                               RDPARM
                     1    ' PARAMETER SYNTAX ERRORS DETECTED.')                      RDPARM
                C                                                                    RDPARM
                C     PARAMETER ANALYSIS CONSTANTS                                   RDPARM
                C                                                                    RDPARM
ISN C043          REAL*8  DTYPES(10), DSTYPS(4), CROW, CCOL, CROWS, CCOLS, TVAL      RDPARM
ISN C044          REAL*8  CBLK, CTRT, CREP, CNONE, CRAW, CRANK                       RDPARM
ISN C045          REAL*8  DTAIL(3), ALLC, TSTNAM(25)                                 RDPARM
ISN C046          REAL*4  TRC(20)                                                    RDPARM
ISN C047          INTEGER*4  TSTNUM(25)                                              RDPARM
                C                                                                    RDPARM
                C     INITIALIZATION OF PARAMETER ANALYSIS CONSTANTS                 RDPARM
                C                                                                    RDPARM
                C     TYPES OF DATA FROM THE DATA PARAMETER                          RDPARM
ISN C048          DATA  DTYPES / '1RS', '2RS', 'KRS', 'BIVAR', 'MULVAR', '2MS',      RDPARM
                 1    'RCB', 'BIBD', 'UNDEF', 'UNDEF' /                             RDPARM
```

```
                C     RCB OR BIBD DATA LABELLING FOR INPUT FROM DATA STATEMENT          RDPARM
                C     ALSO TRANSFORMATION SPECIFICATIONS FROM THE TRANS PARAMETER CARD  RDPARM
ISN C049              DATA  CBLK/'BLOCK'/, CTRT/'TREAT'/, CREP/'REP'/,                  RDPARM
                     1    CNONE/'NONE'/, CRAW/'RAW'/, CRANK/'SAMPLE'/                  RDPARM
                C     TYPES OF DATA DISTRIBUTIONS FROM THE DISTRIB STATEMENT            RDPARM
ISN C050              DATA  DSTYPS / 'NORMAL', 'POISSON', 'UNIFORM', 'BINOMIAL'/        RDPARM
                C     ROW OR COLUMN LABELS FROM THE CLASSES CARDS                       RDPARM
ISN C051              DATA  CROW /'ROW'/, CROWS/'ROWS'/, CCOL/'COLUMN'/,CCOLS/'COLUMNS'/RDPARM
                C     TAILS TO BE TESTED FORM THE TESTS CARD                            RDPARM
ISN C052              DATA  DTAIL / 'LOWER', '2-TAIL', 'UPPER'/, ALLC /'ALL'/           RDPARM
                C     NUMBER OF CURRENTLY VALID TESTS IN TSTNAM & TSTNUM                RDPARM
ISN C053              DATA  NTSTS /20/                                                  RDPARM
                C     KEYWORD ABBREVIATIONS OF TESTS CURRENTLY IMPLEMENTED              RDPARM
ISN C054              DATA  TSTNAM / 'SIGN', 'MCNEMAR', 'CHIPROB', 'CHINDEP',           RDPARM
                     1    'MEDIAN', 'SIGNRANK', 'M-W', 'MANN-WHI', 'TNEY', 'MOOD',     RDPARM
                     2    'RHO', 'TREND', 'TAU', 'K-W', 'KRUSKAL-', 'WALLIS',          RDPARM
                     3    'FRIEDMAN', 'CHIGOF', 'GOODFIT', 'DURBIN',                   RDPARM
                     4    ' ', ' ', ' ', ' ', ' ' /                                    RDPARM
                C     NUMBER CODES CORRESPONDING TO THE TESTS IN TSTNAM                 RDPARM
ISN C055              DATA  TSTNUM / 1, 2, 3, 4,                                        RDPARM
                     1    5, 6, 7, 7, 7, 8,                                            RDPARM
                     2    9, 10, 11, 12, 12, 12,                                       RDPARM
                     3    13, 14, 14, 15,                                              RDPARM
                     4    0, 0, 0, 0, 0 /                                              RDPARM
ISN C056              REAL*8  INTRNL /'INTERNAL'/, EXTRN /'EXTERNAL'/                   RDPARM
ISN C057              REAL*8  TRAW /'RAW'/, TTRAN /'TRANSFOR'/, TMED /'MED'/            RDPARM
ISN C058              REAL*8  TSMPL /'SAMPLE'/, TRANK /'RANKED'/                        RDPARM
ISN C059              REAL*8  DMED /'MEDIAN'/, DICHOT /'LEVEL'/                         RDPARM
                C                                                                       RDPARM
                C     MISCELLANEOUS DATA ITEMS NEEDED                                   RDPARM
                C                                                                       RDPARM
ISN C060              REAL*8  ENDOF /'END OF D'/                                        RDPARM
ISN C061              INTEGER*4  PERR                                                   RDPARM
                C                                                                       RDPARM
                C     INITIALIZE (RESET) ALL THE PARAMETER SPECIFICATIONS               RDPARM
                C                                                                       RDPARM
ISN C062              WKSZE = WKSZE                                                     RDPARM
ISN C063           1C TYPE = 0                                                          RDPARM
ISN C064              SMPLS = 0                                                         RDPARM
ISN C065              VARITE = 0                                                        RDPARM
ISN C066              BLK = 0                                                           RDPARM
ISN C067              TRT = 0                                                           RDPARM
ISN C068              REP = 0                                                           RDPARM
```

```
ISN C069            BLKT = 0                                                    RDPARM
ISN C070            TRTB = 0                                                    RDPARM
ISN C071            IPR = 1                                                     RDPARM
ISN C072            IPT = RDR                                                   RDPARM
ISN C073            OUT = 0                                                     RDPARM
ISN C074            DFORM = 0                                                   RDPARM
ISN C075            DOUT = 0                                                    RDPARM
ISN C076            RCBCD = 0                                                   RDPARM
ISN C077            DISTN = 0                                                   RDPARM
ISN C078            WHTRN = 0                                                   RDPARM
ISN C079            ROWS = 0                                                    RDPARM
ISN C080            COLS = 0                                                    RDPARM
ISN C081            TAIL = 0                                                    RDPARM
ISN C082            ALPHA = 0.                                                  RDPARM
ISN C083            DPARM1 = 0.                                                 RDPARM
ISN C084            DPARM2 = 0.                                                 RDPARM.
ISN C085            BLOCK = BLANK                                               RDPARM
ISN C086            TREAT = BLANK                                               RDPARM
ISN C087            REPLI = BLANK                                               RDPARM
                C                                                               RDPARM
ISN C088            DO 12 I = 1, 40                                             RDPARM
ISN C089            LABEL(I) = BLANK                                            RDPARM
ISN C090            SSIZE(I) = 0                                                RDPARM
ISN C091            IPS(I) = 1                                                  RDPARM
ISN C092            FMT(I) = BLANK                                              RDPARM
ISN C093         12 FMT(I+40) = BLANK                                           RDPARM
                C                                                               RDPARM
ISN C094            DO 13 I = 1, 20                                             RDPARM
ISN C095            RLEV(I) = 0.                                                RDPARM
ISN C096            CLEV(I) = 0.                                                RDPARM
ISN C097            FMT(I+80) = BLANK                                           RDPARM
ISN C098         13 TESTS(I) = .FALSE.                                          RDPARM
                C                                                               RDPARM
ISN C099            DOALL = .FALSE.                                             RDPARM
ISN C100            RC  = 0                                                     RDPARM
                C                                                               RDPARM
ISN C101            NPARM = 0                                                   RDPARM
ISN C102            LINE = 0                                                    RDPARM
ISN C103            IERRS = 0                                                   RDPARM
                C                                                               RDPARM
                C   CHECK FOR THE POSSIBILITY OF A TITLE CARD BEING READ IN PLACE RDPARM
                C   OF THE END OF DATA CARD                                     RDPARM
                C                                                               RDPARM
```

```
ISN C104            IF (EOD .NE. 3) GO TO 16                                    RDPARM
ISN C106            DO 14 I = 1, 10                                             RDPARM
ISN C107         14 PARM(I) = BUFFER(I,BUFNO)                                   RDPARM
ISN C108            GO TO 11                                                    RDPARM
                 C                                                              RDPARM
                 C     READ A PARAMETER CARD                                    RDPARM
                 C                                                              RDPARM
ISN C109         16 EOD = 0                                                     RDPARM
ISN C110            READ (RDR,101,END=903) PARM                                 RDPARM
ISN C111            IF (LINE .EQ. 0) GO TO 11                                   RDPARM
ISN C113            WRITE (LST,104) PARM                                        RDPARM
ISN C114            CALL OFLOW(2)                                               RDPARM
ISN C115         11 IF (PARM(1) .EQ. ENDOF) GO TO 10                            RDPARM
                 C     CALL DECODE IF A CARD WAS READ                           RDPARM
ISN C117            CALL DECODE                                                 RDPARM
ISN C118            NPARM = NPARM + 1                                           RDPARM
ISN C119            PERR = 0                                                    RDPARM
                 C     BRANCH BASED ON IPARM, AS SET BY DECODE                  RDPARM
ISN C120            GO TO (26, 17, 16, 16, 18, 19, 20, 21, 22, 23, 24, 25, 901, 902), RDPARM
                   1    IPARM                                                   RDPARM
                 C                                                              RDPARM
                 C     UNKNOWN PARAMETER TYPE                                   RDPARM
                 C                                                              RDPARM
ISN C121       901 IF (LINE .NE. 0) GO TO 911                                   RDPARM
ISN C123            CALL HEAD                                                   RDPARM
ISN C124            WRITE (LST,103)                                             RDPARM
ISN C125            WRITE (LST,104) PARM                                        RDPARM
ISN C126            CALL OFLOW(5)                                               RDPARM
ISN C127       911 WRITE (LST,201)                                              RDPARM
ISN C128            CALL OFLOW(1)                                               RDPARM
ISN C129            RC  = 8                                                     RDPARM
ISN C130            IERRS = IERRS + 1                                           RDPARM
ISN C131            GO TO 16                                                    RDPARM
                 C                                                              RDPARM
                 C     END OF JOB OR END OF RUN CARD                            RDPARM
                 C                                                              RDPARM
ISN C132         25 RC = -1                                                     RDPARM
ISN C133            IF (NPARM .EQ. 1) GO TO 900                                 RDPARM
                 C                                                              RDPARM
                 C     END OF FILE ENCOUNTERED (POSSIBLY END OF JOB)            RDPARM
                 C                                                              RDPARM
ISN C135       903 EOD = 2                                                      RDPARM
ISN C136            IF ( NPARM .EQ. 0) GO TO 900                                RDPARM
```

```
ISN 0138          CALL OFLOW(0)                                                  RDPARM
            C     ERROR BECAUSE AT LEAST 1 PARAMETER HAD BEEN READ IN THIS SET   RDPARM
ISN 0139          WRITE (LST,203)                                                RDPARM
            C     END OF JOB WITH ERROR                                          RDPARM
ISN 0140          RC = -2                                                        RDPARM
            C                                                                    RDPARM
            C     END OF JOB REACHED - NORMAL OR ABNORMAL                        RDPARM
            C                                                                    RDPARM
ISN 0141      900 CALL OFLOW(0)                                                  RDPARM
ISN 0142          IF (RC .EQ. 0) RC = -1                                         RDPARM
ISN 0144          WRITE (LST,200)                                                RDPARM
ISN 0145          RETURN                                                         RDPARM
            C                                                                    RDPARM
            C     CODE 1 - TITLE STATEMENT - CAUSES IMMEDIATE PAGING             RDPARM
            C                                                                    RDPARM
ISN 0146       26 CALL HEAD                                                      RDPARM
ISN 0147          WRITE (LST,103)                                                RDPARM
ISN 0148          WRITE (LST,104) PARM                                           RDPARM
ISN 0149          CALL OFLOW(5)                                                  RDPARM
ISN 0150          GO TO 16                                                       RDPARM
            C                                                                    RDPARM
            C     CODE 2 - DATA STATEMENT                                        RDPARM
            C                                                                    RDPARM
ISN 0151       17 DO 1701 I = 1, 8                                               RDPARM
            C     CHECK THE DATA TYPE SPECIFIED                                  RDPARM
ISN 0152          IF (SUBPRM(1) .EQ. DTYPES(I)) GO TO 1702                       RDPARM
ISN 0154     1701 CONTINUE                                                       RDPARM
            C     ERROR IN FIRST PARAMETER                                       RDPARM
ISN 0155          PERR = 1                                                       RDPARM
            C                                                                    RDPARM
            C     PARAMETER ERROR MESSAGE ROUTINE                                RDPARM
            C                                                                    RDPARM
ISN 0156      902 IF (LINE .NE. 0) GO TO 912                                     RDPARM
ISN 0158          CALL HEAD                                                      RDPARM
ISN 0159          WRITE (LST,103)                                                RDPARM
ISN 0160          WRITE (LST,104) PARM                                           RDPARM
ISN 0161          CALL OFLOW(5)                                                  RDPARM
ISN 0162      912 WRITE (LST,202) PERR                                           RDPARM
ISN 0163          CALL OFLOW(1)                                                  RDPARM
ISN 0164          RC = 8                                                         RDPARM
ISN 0165          IERRS = IERRS + 1                                              RDPARM
ISN 0166          GO TO 16                                                       RDPARM
            C     SET THE DATA TYPE AND CHECK THE OTHER SPECIFICATIONS           RDPARM
```

```
ISN C167      1702 TYPE = I                                                    RDPARM
ISN 0168           PERR = 2                                                    RDPARM
ISN C169           IF (IPCLS(2) .GT. 0) GO TO 902                              RDPARM
ISN C171           IF (TYPE .LE. 3 .OR. TYPE .EC. 6) SMPLS = IVALUE(2)         RDPARM
ISN 0173           IF (TYPE .EQ. 4 .OR. TYPE .EQ. 5) VARITE = IVALUE(2)        RDPARM
ISN C175           IF (INUMB .LT. 3) GO TO 16                                  RDPARM
ISN C177           DO 1703 I = 3, INUMB                                        RDPARM
ISN C178           PERR = I                                                    RDPARM
ISN C179           IF (IPCLS(I) .GT. 0) GO TO 902                              RDPARM
ISN C181      1703 SSIZE(I-2) = IVALUE(I)                                      RDPARM
ISN C182           IF (TYPE .EQ. 7) GO TC 1705                                 RDPARM
ISN C184           IF (TYPE .NE. 8) GO TC 16                                   RDPARM
              C    BIBD DATA                                                   RDPARM
ISN C186           BLK = IVALUE(2)                                             RDPARM
ISN C187           TRT = IVALUE(3)                                             RDPARM
ISN C188           BLKT = IVALUE(4)                                            RDPARM
ISN C189           TRTB = IVALUE(5)                                            RDPARM
ISN C190           DO 1704 I = 2, 5                                            RDPARM
ISN C191           PERR = I                                                    RDPARM
ISN C192           IF (IVALUE(I) .LE. 0) GO TO 902                             RDPARM
ISN C194      1704 CONTINUE                                                    RDPARM
ISN C195           GO TO 16                                                    RDPARM
              C    RCB DATA                                                    RDPARM
ISN C196      1705 BLK = IVALUE(2)                                             RDPARM
ISN C197           TRT = IVALUE(3)                                             RDPARM
ISN C198           REP = IVALUE(4)                                             RDPARM
ISN C199           IF (REP .EQ. 0) REP = 1                                     RDPARM
ISN 0201           PERR = 2                                                    RDPARM
ISN C202           IF (BLK .EQ. 0) GO TO 902                                   RDPARM
ISN C204           PERR = 3                                                    RDPARM
ISN C205           IF (TRT .EQ. 0) GO TO 902                                   RDPARM
ISN 0207           GO TO 16                                                    RDPARM
              C                                                                RDPARM
              C    CODE 5 - CODING STATEMENT                                   RDPARM
              C                                                                RDPARM
ISN C208        18 PERR = 1                                                    RDPARM
ISN 0209           IF (IPCLS(1) .GT. 0) GO TO 902                              RDPARM
ISN C211           IPR = IVALUE(1)                                             RDPARM
ISN C212           IF (INUMB .LT. 2) GO TO 16                                  RDPARM
              C    CHECK WHAT TYPE OF PARAMETERS WERE CODED                    RDPARM
ISN C214           IF (IPCLS(2) .EQ. 1) GO TC 1802                             RDPARM
              C    NUMERICAL PARAMETERS                                        RDPARM
ISN C216           DO 1801 I = 2, INUMB                                        RDPARM
```

```
ISN 0217            PERR = I                                                    RDPARM
ISN 0218            IF (IPCLS(I) .GT. 0) GO TO 902                              RDPARM
ISN 0220            IF (IVALUE(I) .LE. 0) GO TO 902                             RDPARM
ISN 0222      1801  IPS(I-1) = IVALUE(I)                                        RDPARM
ISN 0223            GO TO 16                                                    RDPARM
                C   CHARACTER PARAMETERS - RCB TYPE DATA                        RDPARM
ISN 0224      1802  IF (SUBPRM(2) .NE. CNCNE) GO TO 1803                        RDPARM
ISN 0226            RCBCD = 9                                                   RDPARM
ISN 0227            GO TO 16                                                    RDPARM
ISN 0228      1803  RCBCD = 0                                                   RDPARM
ISN 0229            DO 1807 I = 2, INUMB                                        RDPARM
ISN 0230            IF (SUBPRM(I) .EQ. CBLK) GO TO 1806                         RDPARM
ISN 0232            IF (SUBPRM(I) .EQ. CTRT) GO TO 1805                         RDPARM
ISN 0234            IF (SUBPRM(I) .EQ. CREP) GO TO 1804                         RDPARM
ISN 0236            PERR = I                                                    RDPARM
ISN 0237            GO TO 902                                                   RDPARM
                C   THIS IS A VERY SNEAKY WAY OF CALCULATING WHAT WAS SPECIFIED  RDPARM
                C   BUT A DUMB USER COULD WIPE IT OUT ROYAL                     RDPARM
ISN 0238      1804  RCBCD = RCBCD + 2                                           RDPARM
ISN 0239      1805  RCBCD = RCBCD + 1                                           RDPARM
ISN 0240      1806  RCBCD = RCBCD + 1                                           RDPARM
ISN 0241      1807  CONTINUE                                                    RDPARM
ISN 0242            GO TO 16                                                    RDPARM
                C                                                               RDPARM
                C   CODE 6 - SOURCE STATEMENT                                   RDPARM
                C                                                               RDPARM
ISN 0243         1$ PERR = 1                                                    RDPARM
ISN 0244            IF (IPCLS(1) .GT. 0) GO TO 902                              RDPARM
ISN 0246            PERR = 3                                                    RDPARM
ISN 0247            IF (IPCLS(3) .GT. 0) GO TO 902                              RDPARM
ISN 0249            IPT = IVALUE(1)                                             RDPARM
ISN 0250            OUT = IVALUE(3)                                             RDPARM
ISN 0251            DFORM = -1                                                  RDPARM
ISN 0252            DOUT = -1                                                   RDPARM
                C   SET A ZERO INPUT UNIT TO THE STANDARD READER                RDPARM
ISN 0253            IF (IPT .EQ. 0) IPT = RDR                                   RDPARM
ISN 0255            PERR = 1                                                    RDPARM
                C   CHECK FOR POSSIBLE UNIT CONFLICTS                           RDPARM
ISN 0256            IF (IPT .EQ. PCH                                            RDPARM
                   1    .OR. IPT .EQ. LST                                       RDPARM
                   2    .OR. IPT .EQ. UT1                                       RDPARM
                   3    .OR. IPT .EQ. UT2                                       RDPARM
                   4    .OR. IPT .EQ. UT3                                       RDPARM
```

```
                5    .OR. IPT .EQ. UT4) GO TO 902                                  RDPARM
ISN C258         PERR = 3                                                          RDPARM
ISN C259         IF (OUT .EQ. IPT                                                  RDPARM
                1    .OR. OUT .EQ. RDR                                            RDPARM
                2    .OR. OUT .EQ. PCH                                            RDPARM
                4    .OR. OUT .EQ. UT1                                            RDPARM
                5    .OR. OUT .EQ. UT2                                            RDPARM
                6    .OR. OUT .EQ. UT3                                            RDPARM
                7    .OR. OUT .EQ. UT4) GO TO 902                                 RDPARM
          C      CHECK FORMS SPECIFICATION                                         RDPARM
ISN C261         PERR = 2                                                          RDPARM
ISN C262         IF (SUBPRM(2) .EQ. BLANK) DFORM = 0                               RDPARM
ISN C264         IF (SUBPRM(2) .EQ. INTRNL) DFORM = 1                              RDPARM
ISN C266         IF (SUBPRM(2) .EQ. EXTRN) DFORM = 2                               RDPARM
ISN 0268         IF (IPT .EQ. RDR .AND. DFORM .EQ. 2) DFORM = 0                    RDPARM
ISN C270         IF (IPT .EQ. RDR .AND. DFORM .EQ. 1) GO TO 902                    RDPARM
ISN C272         IF (DFORM .EQ. -1) GO TO 902                                      RDPARM
ISN C274         IF (IPT .NE. RDR .AND. DFORM .EQ. 0) DFORM = 1                    RDPARM
ISN C276         PERR = 4                                                          RDPARM
ISN C277         IF (SUBPRM(4) .EQ. BLANK) DOUT = 0                                RDPARM
ISN C279         IF (SUBPRM(4) .EQ. TRAW) DOUT = 0                                 RDPARM
ISN C281         IF (SUBPRM(4) .EQ. TTRAN .AND. SUBPRM(5) .EQ. TMED) DOUT = 1      RDPARM
ISN C283         IF (SUBPRM(4) .EQ. TSMPL) DOUT = 2                                RDPARM
ISN C285         IF (SUBPRM(4) .EQ. TRANK) DOUT = 3                                RDPARM
ISN C287         IF (DOUT .EQ. -1) GO TO 902                                       RDPARM
ISN 0289         GO TO 16                                                          RDPARM
          C                                                                        RDPARM
          C      CODE 7 - DISTRIB STATEMENT - TYPE OF DISTRIBUTION AND PARAMETERS  RDPARM
          C                                                                        RDPARM
ISN C290       2C PERR = 1                                                         RDPARM
ISN C291         I = -1                                                            RDPARM
ISN C292         IF (SUBPRM(1) .EQ. BLANK .OR. SUBPRM(1) .EQ. DMED                 RDPARM
                1    .OR. SUBPRM(1) .EQ. DICHCT) GO TO 2002                       RDPARM
ISN C294         DO 2001 I = 1, 4                                                  RDPARM
          C      CHECK VALIDITY OF TYPE OF DISTRIBUTION                            RDPARM
ISN C295         IF (SUBPRM(1) .EQ. DSTYPS(I)) GO TO 2002                          RDPARM
ISN C297       2001 CONTINUE                                                       RDPARM
ISN C298         GO TO 902                                                         RDPARM
ISN C299       2002 DISTN = I                                                      RDPARM
ISN C300         IF (INUMB .LT. 2) GO TO 16                                        RDPARM
ISN 0302         DPARM1 = VALUE(2)                                                 RDPARM
ISN C303         DPARM2 = VALUE(3)                                                 RDPARM
ISN C304         IF (IPCLS(3) .EQ. 1) PERR = 3                                     RDPARM
```

```
ISN 0306            IF (IPCLS(2) .EQ. 1) PERR = 2                                RDPARM
ISN 0308            IF (PERR .EQ. 1) GO TO 16                                    RDPARM
ISN 0310            GO TO 902                                                    RDPARM
                C                                                                RDPARM
                C       CODE 8 - CLASSES STATEMENT - ROWS OR COLUMNS AND VALUES   RDPARM
                C                                                                RDPARM
ISN 0311          21 PERR = 1                                                    RDPARM
ISN 0312            I = 0                                                        RDPARM
ISN 0313            IF (SUBPRM(1) .EQ. CROW .OR. SUBPRM(1) .EQ. CROWS) I = 1     RDPARM
ISN 0315            IF (SUBPRM(1) .EQ. CCOL .OR. SUBPRM(1) .EQ. CCOLS) I = 2     RDPARM
ISN 0317            IF (I .EQ. 0) GO TO 902                                      RDPARM
ISN 0319            PERR = 2                                                     RDPARM
ISN 0320            IF (IPCLS(2) .NE. 0) GO TO 902                               RDPARM
ISN 0322            DO 2101 J = 1, 20                                            RDPARM
ISN 0323         2101 TRC(J) = 0.                                                RDPARM
ISN 0324            ICNT = I                                                     RDPARM
ISN 0325            IF (INUMB .LT. 3) GO TO 2103                                 RDPARM
ISN 0327            VALUE(2) = VALUE(3) - 1.                                     RDPARM
ISN 0328            DO 2102 I = 3, INUMB                                         RDPARM
ISN 0329            PERR = I                                                     RDPARM
ISN 0330            IF (IABS(IPCLS(I)) .EQ. 1) GO TO 902                         RDPARM
ISN 0332            IF (VALUE(I-1) .GE. VALUE(I)) GO TO 902                      RDPARM
ISN 0334         2102 TRC(I-2) = VALUE(I)                                        RDPARM
                C       TRANSFER INFORMATION TO EITHER ROW DATA OR COLUMN DATA    RDPARM
ISN 0335         2103 IF (ICNT .EQ. 2) GO TO 2105                                RDPARM
ISN 0337            ROWS = IVALUE(2)                                             RDPARM
ISN 0338            DO 2104 I = 1, 20                                            RDPARM
ISN 0339         2104 RLEV(I) = TRC(I)                                           RDPARM
ISN 0340            GO TO 16                                                     RDPARM
ISN 0341         2105 COLS = IVALUE(2)                                           RDPARM
ISN 0342            DO 2106 I = 1, 20                                            RDPARM
ISN 0343         2106 CLEV(I) = TRC(I)                                           RDPARM
ISN 0344            GO TO 16                                                     RDPARM
                C                                                                RDPARM
                C       CODE 9 - TESTS STATEMENT - ALPHA, TEST TAIL, AND TESTS    RDPARM
                C                                                                RDPARM
ISN 0345          22 PERR = 1                                                    RDPARM
ISN 0346            ALPHA = VALUE(1)                                             RDPARM
ISN 0347            IF (ALPHA .LE. 0. .OR. ALPHA .GE. 1.) GO TO 902              RDPARM
ISN 0349            TAIL = 0                                                     RDPARM
ISN 0350            J = 2                                                        RDPARM
                C       CHECK TAIL AND SET IF PRESENT, IF OMITTED OR NOT MATCHING, RDPARM
                C       ASSUME THAT THE 2ND PARAMETER IS A TEST SPECIFICATION      RDPARM
```

```
ISN 0351          DO 2201 I = 1, 3                                          RDPARM
ISN 0352     2201 IF (SUBPRM(2) .EQ. DTAIL(I)) TAIL = I                     RDPARM
ISN 0354          IF (TAIL .EQ. 0) GO TO 2202                               RDPARM
ISN 0356          J = 3                                                     RDPARM
ISN 0357          TAIL = TAIL - 2                                           RDPARM
ISN 0358     2202 IF (INUMB .GE. J) GO TO 2204                              RDPARM
            C     NO TESTS SPECIFIED - ASSUME ALL                           RDPARM
ISN 0360     2203 DOALL = .TRUE.                                            RDPARM
ISN 0361          GO TO 16                                                  RDPARM
            C     AT LEAST 1 TEST WAS SPECIFIED                             RDPARM
ISN 0362     2204 IF (INUMB .EQ. J .AND. SUBPRM(J) .EQ. ALLC) GO TO 2203    RDPARM
            C     TEST SPECIFIED WAS NOT ALL                                RDPARM
ISN 0364          DO 2206 I = J, INUMB                                      RDPARM
ISN 0365          PERR = I                                                  RDPARM
ISN 0366          TVAL = SUBPRM(I)                                          RDPARM
ISN 0367          IF (IPCLS(I) .NE. 1) GO TO 902                            RDPARM
            C     SCAN ARRAY OF CURRENTLY DEFINED TESTS                     RDPARM
ISN 0369          DO 2205 K = 1, NTSTS                                      RDPARM
ISN 0370          IF (TVAL .EQ. TSTNAM(K)) GO TO 2206                       RDPARM
ISN 0372     2205 CONTINUE                                                  RDPARM
ISN 0373          GO TO 902                                                 RDPARM
ISN 0374     2206 TESTS( TSTNUM(K) ) = .TRUE.                               RDPARM
ISN 0375          GO TO 16                                                  RDPARM
            C                                                               RDPARM
            C     CODE 10 - TRANS STATEMENT                                 RDPARM
            C                                                               RDPARM
ISN 0376       23 PERR = 1                                                  RDPARM
ISN 0377          IF (INUMB .NE. 1) GO TO 902                               RDPARM
ISN 0379          IF (IPCLS(1) .NE. 1) GO TO 902                            RDPARM
ISN 0381          WHTRN = -2                                                RDPARM
ISN 0382          IF (SUBPRM(1) .EQ. CNONE) WHTRN = 0                       RDPARM
ISN 0384          IF (SUBPRM(1) .EQ. CRAW) WHTRN = -1                       RDPARM
ISN 0386          IF (SUBPRM(1) .EQ. CRANK) WHTRN = 1                       RDPARM
ISN 0388          IF (WHTRN .EQ. -2) GO TO 902                              RDPARM
ISN 0390          GO TO 16                                                  RDPARM
            C                                                               RDPARM
            C     CODE 11 - END OF PARAMETERS OR JUST PLAIN END             RDPARM
            C                                                               RDPARM
ISN 0391       24 CALL OFLOW(0)                                             RDPARM
ISN 0392          IF (IERRS .NE. 0) GO TO 2401                              RDPARM
ISN 0394          WRITE (LST,105)                                           RDPARM
ISN 0395          GO TO 2403                                                RDPARM
            C     PARAMETER SYNTAX ERROR(S) DETECTED                        RDPARM
```

```
   ISN C396       2401 IF (IERRS .NE. 1) GO TO 2402                                RDPARM
   ISN C398            WRITE (LST,204)                                             RDPARM
   ISN C399            GO TO 2403                                                  RDPARM
   ISN C400       2402 WRITE (LST,205) IERRS                                       RDPARM
                C                                                                  RDPARM
                C     RETURN TO CALLING PROGRAM AFTER CHECKING OVERFLOW            RDPARM
                C                                                                  RDPARM
   ISN C401       2403 CALL OFLOW(3)                                               RDPARM
   ISN C402            RETURN                                                      RDPARM
                C                                                                  RDPARM
   ISN C403            END                                                         RDPARM

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     402 ,PROGRAM SIZE =     5342

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE RHO                                                  RHO
              C                                                                    RHO
              C    CORRELATION USING SPEARMAN'S RHO                                RHO
              C                                                                    RHO
              C    THIS SUBROUTINE CALCULATES SPEARMAN'S RANK CORRELATION          RHO
              C    COEFFICIENT, COMMONLY REFERED TO AS SPEARMAN'S RHO.  THE        RHO
              C    VALUE OF RHO IS ALWAYS BETWEEN -1 AND +1 AND CAN BE             RHO
              C    CONSIDERED A MEASURE OF CORRELATION BETWEEN THE 2 HAVLES        RHO
              C    OF A BIVARIATE SAMPLE.  IT CAN BE USED TO TEST HYPOTHESES OF    RHO
              C    NO CORRELATION VERSUS SOME CORRELATION, POSITIVE CORRELATION,   RHO
              C    OR NEGATIVE CORRELATION.                                        RHO
              C                                                                    RHO
              C    THE CRITICAL REGION FOR THE HYPOTHESIS TESTING IS DETERMINED    RHO
              C    FROM A LARGE SAMPLE APPROXIMATION AND CANNOT BE CONSIDERED      RHO
              C    VALID FOR SAMPLES OF LESS THAN 30 OBSERVATIONS.                 RHO
              C                                                                    RHO
              C    FOR A BETTER EXPLANATION FOR SPEARMAN'S RHO, SEE CONOVER, W.J., RHO
              C    PRACTICAL NON-PARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 386-392. RHO
              C                                                                    RHO
              C                                                                    RHO
              C    COMMON BLOCKS                                                   RHO
              C                                                                    RHO
 ISN 0003          COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                    DECLARE
 ISN 0004          INTEGER*4  ROWS,COLS                                            DECLARE
 ISN 0005          COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                        DECLARE
 ISN 0006          INTEGER*4  DISTN,WHTRN                                          DECLARE
 ISN 0007          COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                     DECLARE
 ISN 0008          REAL*8  BUFFER                                                  DECLARE
 ISN 0009          INTEGER*4  EOD,CDCOL,BUFNO                                      DECLARE
 ISN 0010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD           DECLARE
 ISN 0011          REAL*8   FMT                                                    DECLARE
 ISN 0012          INTEGER*4  DFORM,DOUT,RCBCD                                     DECLARE
 ISN 0013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),     DECLARE
                 1    VALUE(40),IVALUE(40),RC                                      DECLARE
 ISN 0014          REAL*8   PARM,SUBPRM                                            DECLARE
 ISN 0015          INTEGER*4  RC                                                   DECLARE
 ISN 0016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                     DECLARE
 ISN 0017          INTEGER*4  TAIL                                                 DECLARE
 ISN 0018          LOGICAL*4  TESTS,DOALL                                          DECLARE
 ISN 0019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),         DECLARE
                 1    DATE,TIME,PAGE,LINE                                          DECLARE
```

```
ISN C020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                    DECLARE
ISN C021          INTEGER*4  PAGE                                                     DECLARE
ISN C022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)  DECLARE
ISN C023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE      DECLARE
ISN C024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C025          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C026          COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                           DECLARE
ISN C027          INTEGER*4  ITEMS,LOCAT,RDNO                                        DECLARE
ISN C028          COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                       DECLARE
ISN C029          INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                             DECLARE
ISN C030          EQUIVALENCE (WORK(1),IWRK(1))                                      DECLARE
ISN C031          REAL*8  BLANK /' '/                                                DECLARE
ISN C032          EQUIVALENCE (BLANK,IBLANK)                                         DECLARE
                C                                                                    RHO
                C     OTHER DATA ITEMS                                               RHO
                C                                                                    RHO
ISN C033          REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                         RHO
ISN C034          REAL*8  XLBL /'X'/, YLBL /'Y'/                                     RHO
ISN C035          REAL*8  PCOR /'POSITIVE'/, NCOR /'NEGATIVE'/                       RHO
ISN C036          REAL*8  DX                                                         RHO
ISN C037          REAL*4 RECORD(6)                                                   RHO
ISN C038          INTEGER*4  CURSMP,CURVAR,CURCNT,XTIES,YTIES,XPLC,YPLC              RHO
ISN C039          EQUIVALENCE (RECORD(1),CURSMP,XRANK), (RECORD(2),CURVAR,YRANK),    RHO
                 1     (RECORD(3),CURCNT,XVALUE), (RECORD(4),YVALUE),                RHO
                 2     (RECORD(6),OBS)                                               RHO
                C                                                                    RHO
                C     FORMAT STATEMENTS                                              RHO
ISN C040        231 FORMAT ('-', T12, 'IKMS31I  UNEQUAL SAMPLE/VARIATE SIZES AFTER TRARHO
                   1NSFORMATION')                                                    RHO
ISN C041        234 FORMAT ('-', T12, 'IKMS34I  INVALID SAMPLE SIZE(S)')           RHO
ISN C042        236 FORMAT ('-', T12, 'IKMS36I  TOO MANY TIED VALUES FOR THIS PROGRAM RHO
                   1TO HANDLE')                                                      RHO
ISN C043        29C FORMAT ('-', T12, 'IKMS90I  ***** BECAUSE OF SAMPLE SIZES, ALPHA HRHO
                   1AT MAY NOT BE CORRECT.  VERIFY WITH TABLES. *****')              RHO
ISN C044        301 FORMAT ('-', T33, 'TEST OF CORRELATION USING SPEARMAN''S RANK CORRRHO
                   1ELATION COEFFICIENT, RHO')                                       RHO
ISN C045        302 FORMAT ('0', T12, 'HO: ', A8, ' AND ', A8,                       RHO
                  1     ' ARE MUTUALLY INDEPENDENT'/ '0', T12, 'VERSUS')             RHO
ISN C046        303 FORMAT ('0', T12, 'H1: ', A8, ' AND ', A8,                       RHO
                  1     ' ARE CORRELATED WITH EACH OTHER IN SOME MANNER')            RHO
ISN C047        304 FORMAT ('0', T12, 'H1: ', A8, ' IS ', A8,                        RHO
                  1     'LY CORRELATED WITH ', A8)                                   RHO
ISN C048        305 FORMAT ('0', T12, 'THE SAMPLE CONSISTS OF', I6,                  RHO
```

```
                 1     ' OBSERVATIONS')                                          RHO
ISN 0049       306 FORMAT ('0', T12, 'THE CORRELATION COEFFICIENT (RHO) IS', F7.3)  RHO
ISN 0050       307 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')       RHO
ISN 0051       308 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                          RHO
ISN 0052       309 FORMAT ('-', T50, 'END OF CORRELATION WITH SPEARMAN''S RHO')     RHO
ISN 0053       310 FORMAT ('0', T12, A8, ' WAS TIED A TOTAL OF', I5,                RHO
                 1     ' TIMES IN', I3, ' PLACES')                                 RHO
            C                                                                     RHO
            C     START BY PRINTING HYPOTHESES AND FIXING THE LABELS              RHO
            C                                                                     RHO
ISN 0054           WRITE (LST,301)                                                RHO
            C                                                                     RHO
ISN 0055           IF (LABEL(1) .EQ. BLANK) LABEL(1) = XLBL                       RHO
ISN 0057           IF (LABEL(2) .EQ. BLANK) LABEL(2) = YLBL                       RHO
            C                                                                     RHO
ISN 0059           ITL = 1                                                        RHO
ISN 0060           IF (TAIL) 11, 13, 12                                           RHO
ISN 0061        11 ITL = 2                                                        RHO
ISN 0062           GO TO 13                                                       RHO
ISN 0063        12 ITL = 3                                                        RHO
            C                                                                     RHO
ISN 0064        13 WRITE (LST,302) LABEL(1), LABEL(2)                             RHO
ISN 0065           GO TO (14, 15, 16), ITL                                        RHO
            C                                                                     RHO
ISN 0066        14 WRITE (LST,303) LABEL(1), LABEL(2)                             RHO
ISN 0067           GO TO 17                                                       RHO
ISN 0068        15 WRITE (LST,304) LABEL(1), NCOR, LABEL(2)                       RHO
ISN 0069           GO TO 17                                                       RHO
ISN 0070        16 WRITE (LST,304) LABEL(1), PCOR, LABEL(2)                       RHO
            C                                                                     RHO
ISN 0071        17 CALL OFLOW(9)                                                  RHO
            C                                                                     RHO
            C     KILL ANY POSSIBLE INDICATION OF PREVIOUSLY SORTED DATA          RHO
            C                                                                     RHO
ISN 0072           SUBS(1) = 0                                                    RHO
ISN 0073           SUBS(2) = 0                                                    RHO
            C                                                                     RHO
ISN 0074           REWIND UT1                                                     RHO
ISN 0075           REWIND UT2                                                     RHO
            C                                                                     RHO
            C     PASS 1 - PICK UP FIRST VARIATE                                  RHO
            C                                                                     RHO
ISN 0076        21 READ (UT1,END=23) RECORD                                       RHO
```

```
ISN C077          IF (CURVAR .NE. 1) GO TO 21                                    RHO
ISN C079          WRITE (UT2) RECORD                                             RHO
ISN C080          GO TO 21                                                       RHO
                C                                                                RHO
                C     SECOND PASS - COMBINE THE VARIATES TOGETHER                RHO
                C                                                                RHO
ISN C081       23 ENDFILE UT2                                                    RHO
ISN C082          REWIND UT1                                                     RHO
ISN C083          REWIND UT2                                                     RHO
ISN C084          REWIND UT4                                                     RHO
                C                                                                RHO
ISN C085       24 READ (UT1,END=26) RECORD                                       RHO
ISN C086          IF (CURVAR .EQ. 1) GO TO 24                                    RHO
                C                                                                RHO
ISN C088          X(2) = OBS                                                     RHO
ISN C089          READ(UT2,END=1231) RECORD                                      RHO
ISN C090          X(1) = OBS                                                     RHO
ISN C091          ITEMS = 2                                                      RHO
ISN C092          LOCAT(4) = 0                                                   RHO
ISN C093          IF (WHTRN .EQ. 1) CALL TRANS                                   RHO
ISN C095          IF (ITEMS .NE. 2) GO TO 24                                     RHO
ISN C097          IF (OUT .NE. 0 .AND. COUT .EQ. 2) CALL WRDATA                  RHO
                C                                                                RHO
                C     WRITE OUT THE COMBINED OBSERVATION                         RHO
                C                                                                RHO
ISN C099          XRANK = 0.                                                     RHO
ISN C100          YRANK = 0.                                                     RHO
ISN C101          XVALUE = X(1)                                                  RHO
ISN C102          YVALUE = X(2)                                                  RHO
ISN C103          OBS = XVALUE                                                   RHO
ISN C104          WRITE (UT4) RECORD                                             RHO
ISN C105          GO TO 24                                                       RHO
                C                                                                RHO
                C     END OF SECOND PASS - OBSERVATIONS ARE COMBINED             RHO
                C                                                                RHO
ISN C106       26 IF (OUT .NE. 0 .AND. COUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT RHO
ISN C108          OUT = 0                                                        RHO
ISN C109          ENDFILE UT4                                                    RHO
                C                                                                RHO
                C     SORT THE DATA ON THE X VALUE                               RHO
                C                                                                RHO
ISN C110          CALL RANKER(UT2,UT4,UT2,UT3,J)                                 RHO
ISN C111          IF (RC .NE. 0) GO TO 92                                        RHO
```

```
ISN C113            RANK = 0.                                                    RHO
ISN C114            T = 0.                                                       RHO
ISN C115            XTIES = 0                                                    RHO
ISN C116            XPLC = 0                                                     RHO
ISN C117            REWIND UT2                                                   RHO
ISN C118            REWIND UT4                                                   RHO
ISN C119            READ (UT2,END=1234) RECORD                                   RHO
                C                                                                RHO
                C     SAVE THE OBSERVATION                                       RHO
                C                                                                RHO
ISN C120         31 DO 32 I = 1, 6                                               RHO
ISN C121         32 DATA(I) = RECORD(I)                                          RHO
ISN C122            RANK = RANK + 1.                                             RHO
                C                                                                RHO
                C     READ AN OBSERVATION AND CHECK FOR A TIE                    RHO
                C                                                                RHO
ISN C123            READ (UT2,END=33) RECORD                                     RHO
ISN C124            IF (DATA(6) .EQ. OBS) GO TO 36                               RHO
                C                                                                RHO
                C     WRITE OUT UNTIED DATA FOR SORTING ON Y                     RHO
                C                                                                RHO
ISN C126            DATA(1) = RANK                                               RHO
ISN C127            DATA(6) = DATA(4)                                            RHO
ISN C128            WRITE (UT4) (DATA(I), I = 1, 6)                              RHO
ISN C129            GO TO 31                                                     RHO
                C                                                                RHO
                C     END OF FILE - PROCESS LAST OBSERVATION                     RHO
                C                                                                RHO
ISN C130         33 XRANK = RANK                                                 RHO
ISN C131            OBS = YVALUE                                                 RHO
ISN C132            WRITE (UT4) RECORD                                           RHO
ISN C133            GO TO 51                                                     RHO
                C                                                                RHO
                C     PROCESS TIED OBSERVATIONS                                  RHO
                C     OBSERVATIONS MUST BE SAVED, SINCE THEY MUST BE WRITTEN     RHO
                C     OUT WITH THE TIED RANK VALUE AND THE Y VALUE FOR SORTING.  RHO
                C     IF MORE THAN (WKSZE/6) TIED VALUES, THIS TEST ABORTS.      RHO
                C     CURRENTLY THAT VALUE IS 833, SO THE DATA MUST BE PRETTY BAD RHO
                C     FOR SUCH AN OCCURANCE TO HAPPEN.                           RHO
                C                                                                RHO
ISN C134         36 XPLC = XPLC + 1                                              RHO
ISN C135            LIM = (WKSZE / 6) * 6                                        RHO
ISN C136            IPTR = 0                                                     RHO
```

```
ISN C137            DO 37 I = 1, 6                                          RHO
ISN C138         37 WORK(I) = DATA(I)                                       RHO
                 C                                                          RHO
                 C     SAVE THE NEXT TIED OBSERVATION                       RHO
                 C                                                          RHO
ISN C139         38 IPTR = IPTR + 6                                         RHO
ISN C140            IF (IPTR .EQ. LIM) GO TO 1236                           RHO
ISN C142            DO 39 I = 1, 6                                          RHO
ISN C143         39 WORK(IPTR+I) = RECORD(I)                                RHO
                 C                                                          RHO
                 C     READ THE NEXT OBSERVATION AND CHECK FOR ANOTHER TIE  RHO
                 C                                                          RHO
ISN C144            READ (UT2,END=43) RECORD                                RHO
ISN C145            IF (DATA(6) .EQ. OBS) GO TO 38                          RHO
                 C                                                          RHO
                 C     NEXT VALUE IS NOT A TIE                              RHO
                 C                                                          RHO
ISN C147            J = (IPTR + 6) / 6                                      RHO
ISN C148            XTIES = XTIES + J                                       RHO
ISN C149            Z = 0.                                                  RHO
                 C                                                          RHO
ISN C150            DO 40 I = 1, J                                          RHO
ISN C151            Z = Z + RANK                                            RHO
ISN C152         40 RANK = RANK + 1.                                        RHO
                 C                                                          RHO
                 C     CALCULATE TIE VALUE AND WRITE OUT OBSERVATIONS FOR Y SORTING  RHO
                 C                                                          RHO
ISN C153            Z = Z / J                                               RHO
ISN C154            JPTR = 0                                                RHO
                 C                                                          RHO
ISN C155         41 WORK(JPTR+1) = Z                                        RHO
ISN C156            WORK(JPTR+6) = WORK(JPTR+4)                             RHO
ISN C157            WRITE (UT4) (WORK(JPTR+I), I = 1, 6)                    RHO
ISN C158            JPTR = JPTR + 6                                         RHO
ISN C159            IF (JPTR .LE. IPTR) GO TO 41                            RHO
ISN C161            RANK = RANK - 1.                                        RHO
ISN C162            GO TO 31                                                RHO
                 C                                                          RHO
                 C     END OF FILE WHILE SEARCHING FOR ANOTHER TIE          RHO
                 C                                                          RHO
ISN C163         43 J = (IPTR + 6) / 6                                      RHO
ISN C164            XTIES = XTIES + J                                       RHO
ISN C165            Z = 0.                                                  RHO
```

```
                C                                                                RHO
ISN C166            DO 44 I = 1, J                                               RHO
ISN C167            Z = Z + RANK                                                 RHO
ISN C168         44 RANK = RANK + 1.                                             RHO
                C                                                                RHO
                C     CALCULATE RANK AVERAGE AND WRITE OUT FOR Y SORTING          RHO
                C                                                                RHO
ISN C169            Z = Z / J                                                    RHO
ISN C170            JPTR = 0                                                     RHO
                C                                                                RHO
ISN C171         45 WORK(JPTR+1) = Z                                             RHO
ISN C172            WORK(JPTR+6) = WORK(JPTR+4)                                  RHO
ISN C173            WRITE (UT4) (WORK(JPTR+I), I = 1, 6)                         RHO
ISN C174            JPTR = JPTR + 6                                              RHO
ISN C175            IF (JPTR .LE. IPTR) GO TO 45                                 RHO
                C                                                                RHO
                C     END OF THE X RANKING PASS                                  RHO
                C                                                                RHO
ISN C177         51 ENDFILE UT4                                                  RHO
ISN C178            CALL RANKER (UT2,UT4,LT2,UT3,J)                              RHO
ISN C179            IF (RC .NE. 0) GO TO 92                                      RHO
                C                                                                RHO
ISN C181            REWIND UT2                                                   RHO
ISN C182            RANK = 0.                                                    RHO
ISN C183            YPLC = 0                                                     RHO
ISN C184            YTIES = 0                                                    RHO
ISN C185            SSIZE(1) = 0                                                 RHO
                C                                                                RHO
ISN C186            READ (UT2,END=1234) RECORD                                   RHO
ISN C187            SSIZE(1) = SSIZE(1) + 1                                      RHO
                C                                                                RHO
                C     SAVE THE OBSERVATION AND READ THE NEXT ONE                 RHO
                C                                                                RHO
ISN C188         52 DO 53 I = 1, 6                                               RHO
ISN 0189         53 DATA(I) = RECORD(I)                                          RHO
ISN 0190            RANK = RANK + 1.                                             RHO
                C                                                                RHO
ISN C191            READ (UT2,END=54) RECORD                                     RHO
ISN 0192            SSIZE(1) = SSIZE(1) + 1                                      RHO
ISN C193            IF (DATA(6) .EQ. OBS) GO TO 56                               RHO
                C                                                                RHO
                C     CALCULATE RANK DIFFERENCE ON A NON-TIED VALUE              RHO
                C                                                                RHO
```

```
ISN 0195            T = T + (DATA(1) - RANK) ** 2                                  RHO
ISN 0196            GO TO 52                                                       RHO
                C                                                                  RHO
                C     END OF FILE - CALCULATE LAST CONTRIBUTION TO TEST STATISTIC   RHO
                C                                                                  RHO
ISN 0197         54 T = T + (DATA(1) - RANK) ** 2                                  RHO
ISN 0198            GO TO 71                                                       RHO
                C                                                                  RHO
                C     Y VALUES TIED                                                RHO
                C                                                                  RHO
ISN 0199         56 YPLC = YPLC + 1                                                RHO
ISN 0200            LIM = (WKSZE / 6) * 6                                          RHO
ISN 0201            IPTR = 0                                                       RHO
ISN 0202            DO 57 I = 1, 6                                                 RHO
ISN 0203         57 WORK(I) = DATA(I)                                              RHO
                C                                                                  RHO
                C     SAVE OBSERVATION IN WORK AND READ NEXT OBSERVATION           RHO
                C                                                                  RHO
ISN 0204         58 IPTR = IPTR + 6                                                RHO
ISN 0205            IF (IPTR .EQ. LIM) GO TO 1236                                  RHO
ISN 0207            DO 59 I = 1, 6                                                 RHO
ISN 0208         59 WORK(IPTR+I) = RECORD(I)                                       RHO
                C                                                                  RHO
ISN 0209            READ (UT2,END=63) RECORD                                       RHO
ISN 0210            SSIZE(1) = SSIZE(1) + 1                                        RHO
ISN 0211            IF (DATA(6) .EQ. OBS) GO TO 58                                 RHO
                C                                                                  RHO
                C     ALL OF TIED VALUES HAVE BEEN READ                            RHO
                C                                                                  RHO
ISN 0213            J = (IPTR + 6) / 6                                             RHO
ISN 0214            YTIES = YTIES + J                                              RHO
ISN 0215            Z = 0.                                                         RHO
                C                                                                  RHO
ISN 0216            DO 60 I = 1, J                                                 RHO
ISN 0217            Z = Z + RANK                                                   RHO
ISN 0218         60 RANK = RANK + 1.                                               RHO
                C                                                                  RHO
                C     CALCULATE TIED CONTRIBUTION TO TEST STATISTIC                RHO
                C                                                                  RHO
ISN 0219            Z = Z / J                                                      RHO
ISN 0220            JPTR = 0                                                       RHO
ISN 0221         61 T = T + (WORK(JPTR+1) - Z) ** 2                                RHO
ISN 0222            JPTR = JPTR + 6                                                RHO
```

```
ISN 0223            IF (JPTR .LE. IPTR) GO TO 61                                RHO
ISN 0225            RANK = RANK - 1.                                            RHO
ISN 0226            GO TO 52                                                    RHO
                C                                                               RHO
                C     END OF FILE SEARCHING FOR MORE TIES                       RHO
                C                                                               RHO
ISN 0227         63 J = (IPTR + 6) / 6                                          RHO
ISN 0228            YTIES = YTIES + J                                           RHO
ISN 0229            Z = 0.                                                      RHO
                C                                                               RHO
ISN 0230            DO 64 I = 1, J                                              RHO
ISN 0231            Z = Z + RANK                                                RHO
ISN 0232         64 RANK = RANK + 1.                                            RHO
                C                                                               RHO
                C     CALCULATE FINAL CONTRIBUTION TO TEST STATISTIC            RHO
                C                                                               RHO
ISN 0233            Z = Z / J                                                   RHO
ISN 0234            JPTR = 0                                                    RHO
                C                                                               RHO
ISN 0235         65 T = T + (WORK(JPTR+1) - Z) ** 2                             RHO
ISN 0236            JPTR = JPTR + 6                                             RHO
ISN 0237            IF (JPTR .LE. IPTR) GO TO 65                                RHO
                C                                                               RHO
                C     END OF CALCULATING T                                      RHO
                C                                                               RHO
ISN 0239         71 WRITE (LST,305) SSIZE(1)                                    RHO
ISN 0240            CALL OFLOW(2)                                               RHO
                C                                                               RHO
ISN 0241            IF (XTIES .EQ. 0) GO TO 72                                  RHO
ISN 0243            WRITE (LST,310) LABEL(1),XTIES,XPLC                         RHO
ISN 0244            CALL OFLOW(2)                                               RHO
                C                                                               RHO
ISN 0245         72 IF (YTIES .EQ. 0) GO TO 73                                  RHO
ISN 0247            WRITE (LST,310) LABEL(2),YTIES,YPLC                         RHO
ISN 0248            CALL OFLOW(2)                                               RHO
                C                                                               RHO
                C     CALCULATE THE VALUE OF RHO                                RHO
                C                                                               RHO
ISN 0249         73 DN = SSIZE(1) * (SSIZE(1) * SSIZE(1) - 1)                   RHO
ISN 0250            R = 1. - 6. * T / DN                                        RHO
ISN 0251            WRITE (LST,306) R                                           RHO
ISN 0252            CALL OFLOW(2)                                               RHO
                C                                                               RHO
```

```
ISN C253            DX = SSIZE(1) - 1                                    RHO
ISN C254            ADX = DSQRT(DX)                                      RHO
                C                                                        RHO
ISN C255            GO TO (76, 81, 86), ITL                              RHO
                C                                                        RHO
                C     2-TAILED CORRELATION TEST                          RHO
                C                                                        RHO
ISN C256         76 ALFHAT = ANDTR(ADX*R)                                RHO
ISN C257            ALFHAT = 2. * AMIN1(ALFHAT, 1.-ALFHAT)              RHO
ISN C258            IF (ALFHAT .GT. 1.) ALFHAT = 1.                     RHO
ISN C260            GO TO 91                                            RHO
                C                                                        RHO
                C     NEGATIVE CORRELATION                               RHO
                C                                                        RHO
ISN C261         81 ALFHAT = ANDTR(ADX*R)                                RHO
ISN C262            GO TO 91                                            RHO
                C                                                        RHO
                C     POSITIVE CORRELATION                               RHO
                C                                                        RHO
ISN C263         86 ALFHAT = ANDTR(-ADX*R)                               RHO
                C                                                        RHO
                C     DETERMINE CONCLUSION TO BE REACHED                 RHO
                C                                                        RHO
ISN C264         91 I = 1                                                RHO
ISN C265            IF (ALPHA .GT. ALFHAT) I = 2                        RHO
ISN C267            WRITE (LST,307) ALPHA, ACCREJ(I)                    RHO
ISN C268            WRITE (LST,308) ALFHAT                              RHO
ISN C269            CALL OFLOW(5)                                       RHO
                C                                                        RHO
                C     CHECK FOR POOR SAMPLE SIZE                         RHO
                C                                                        RHO
ISN C270            IF (SSIZE(1) .GT. 30) GO TO 92                      RHO
ISN C272            WRITE (LST,290)                                     RHO
ISN C273            CALL OFLOW(3)                                       RHO
                C                                                        RHO
                C     END OF THE CORRELATION WITH RHO                    RHO
                C                                                        RHO
ISN C274         92 WRITE (LST,309)                                      RHO
ISN C275            CALL OFLOW(3)                                       RHO
ISN C276            RETURN                                              RHO
                C                                                        RHO
                C     ERROR IS THE DATA                                  RHO
                C                                                        RHO
```

```
   ISN C277       1231 WRITE (LST,231)                                          RHO
   ISN C278            GO TO 99                                                 RHO
                C                                                               RHO
   ISN C279       1234 WRITE (LST,234)                                          RHO
   ISN C280            GO TO 99                                                 RHO
                C                                                               RHO
   ISN C281       1236 WRITE (LST,236)                                          RHO
                C                                                               RHO
   ISN C282         99 CALL OFLOW(3)                                           RHO
   ISN C283            RC = 12                                                  RHO
   ISN C284            GO TO 92                                                 RHO
                C                                                               RHO
   ISN C285            END                                                      RHO

*OPTIONS IN EFFECT*       NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*      SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*      SOURCE STATEMENTS =      284 ,PROGRAM SIZE =      5002

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE SETIO                                                  SETIO
              C                                                                       SETIO
              C    THIS SUBROUTINE IS USED ONLY ONCE AT THE VERY START OF THE         SETIO
              C    PROGRAM TO INITIALIZE THE FORTRAN I/O UNITS THAT WILL BE USED      SETIO
              C    BY THE PROGRAM.  HENCE, THIS ROUTINE CAN BE MODIFIED FOR           SETIO
              C    INSTALLATIONS WHERE THE I/O ASSIGNMENTS ARE DIFFERENT FROM         SETIO
              C    THOSE CURRENTLY SPECIFIED, AND THE MAIN PROGRAM CAN REMAIN         SETIO
              C    INTACT WITH NO CHANGES NECESSARY.                                  SETIO
              C                                                                       SETIO
              C    THE UNIT NAMES ARE CARRIED IN THE COMMON BLOCK /UNITS/,            SETIO
              C    AND HAVE THE FOLLOWING MEANINGS:                                   SETIO
              C                                                                       SETIO
              C         RDR IS THE STANDARD FORTAN INPUT UNIT FOR CARDS.              SETIO
              C         PCH IS THE STANDARD FORTRAN OUTPUT UNIT FOR PUNCHED OUTPUT    SETIO
              C         LST IS THE STANDARD FORTRAN OUTPUT UNIT FOR PRINTED LISTINGS  SETIO
              C         UT1 IS THE FORTRAN UNIT FOR A SEQUENTIAL WORK FILE            SETIO
              C         UT2 IS THE FORTRAN UNIT NUMBER FOR ANOTHER WORK FILE          SETIO
              C         UT3 IS THE FORTRAN UNIT NUMBER FOR ANOTHER WORK FILE          SETIO
              C         UT4 IS THE FORTRAN UNIT NUMBER FOR ANOTHER WORK FILE          SETIO
              C         IPT AND OUT ARE SET VIA DATA PARAMETERS AND ANY SETTING       SETIO
              C         OF THEM IN THIS ROUTINE IS IGNORED.                           SETIO
              C                                                                       SETIO
              C    NOTE THAT SINCE THE UNIT NAMES ARE IN COMMON, THEY CANNOT          SETIO
              C    BE INITIAILIZED BY A DATA STATEMENT, BARRING A BLOCK DATA          SETIO
              C    SUBROUTINE, WHICH IS NOT RECCOMMENDED.                             SETIO
              C                                                                       SETIO
              C    DEFINE THE COMMON BLOCK /UNITS/                                    SETIO
              C                                                                       SETIO
 ISN 0003          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                SETIO
 ISN 0004          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                SETIO
              C                                                                       SETIO
              C    INITIALIZE THE UNIT NUMBERS                                        SETIO
              C                                                                       SETIO
 ISN 0005          RDR = 1                                                           SETIO
 ISN 0006          PCH = 2                                                           SETIO
 ISN 0007          LST = 3                                                           SETIO
 ISN 0008          UT1 = 11                                                          SETIO
 ISN 0009          UT2 = 12                                                          SETIO
 ISN 0010          UT3 = 13                                                          SETIO
 ISN 0011          UT4 = 14                                                          SETIO
              C                                                                       SETIO
```

```
                C     THESE UNIT NUMBERS CORRESPOND TO VALID UNIT NUMBERS IN THE        SETIO
                C     KANSAS STATE UNIVERSITY COMPUTING CENTER, JULY, 1970.             SETIO
                C                                                                       SETIO
   ISN C012           RETURN                                                            SETIO
   ISN C013           END                                                               SETIO

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      12 ,PROGRAM SIZE =      260

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE SGNTST                                                SGNTST
              C                                                                     SGNTST
              C    THE SIGN TEST                                                    SGNTST
              C                                                                     SGNTST
              C    THE SIGN TEST CAN BE USED TO COMPARE TWO RANDOM SAMPLES OF       SGNTST
              C    THE SAME SIZE OR THE TWO ELEMENTS OF A BIVARIATE SAMPLE.         SGNTST
              C    THE SIGN TEST TESTS THE PROBABILITY OF THE FIRST ELEMENT         SGNTST
              C    OR VARIATE BEING LARGER THAN ITS CORRESPONDING SECOND ELEMENT.   SGNTST
              C    HENCE, THE TEST CAN BE USED TO TEST THE EXPECTED VALUES          SGNTST
              C    OF THE TWO SAMPLES OR VARIATES.                                  SGNTST
              C                                                                     SGNTST
              C    THE CRITICAL REGION IS COMPUTED USING THE BINOMIAL DISTRIBUTION  SGNTST
              C    WITH PROBABILITY OF .5.                                          SGNTST
              C                                                                     SGNTST
              C    FOR A COMPLETE DESCRIPTION OF THE SIGN TEST, SEE CONOVER, W.J.,  SGNTST
              C    PRACTICAL NON-PARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 197-205  SGNTST
              C                                                                     SGNTST
              C                                                                     SGNTST
              C    COMMON BLOCKS                                                    SGNTST
              C                                                                     SGNTST
 ISN 0003          COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                     DECLARE
 ISN 0004          INTEGER*4  ROWS,COLS                                             DECLARE
 ISN 0005          COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                         DECLARE
 ISN 0006          INTEGER*4  DISTN,WHTRN                                           DECLARE
 ISN 0007          COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                      DECLARE
 ISN 0008          REAL*8  BUFFER                                                   DECLARE
 ISN 0009          INTEGER*4  EOD,CDCOL,BUFNO                                       DECLARE
 ISN 0010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD            DECLARE
 ISN 0011          REAL*8   FMT                                                     DECLARE
 ISN 0012          INTEGER*4  DFORM,DOUT,RCBCD                                      DECLARE
 ISN 0013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),      DECLARE
                  1    VALUE(40),IVALUE(40),RC                                      DECLARE
 ISN 0014          REAL*8   PARM,SUBPRM                                             DECLARE
 ISN 0015          INTEGER*4  RC                                                    DECLARE
 ISN 0016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                      DECLARE
 ISN 0017          INTEGER*4  TAIL                                                  DECLARE
 ISN 0018          LOGICAL*4  TESTS,DOALL                                           DECLARE
 ISN 0019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),          DECLARE
                  1    DATE,TIME,PAGE,LINE                                          DECLARE
 ISN 0020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                 DECLARE
 ISN 0021          INTEGER*4  PAGE                                                  DECLARE
```

```
ISN C022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
ISN C023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE     DECLARE
ISN C024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C025          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C026          COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                          DECLARE
ISN C027          INTEGER*4  ITEMS,LOCAT,RDNO                                       DECLARE
ISN C028          COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                      DECLARE
ISN C029          INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                            DECLARE
ISN C030          EQUIVALENCE (WORK(1),IWRK(1))                                     DECLARE
ISN C031          REAL*8  BLANK /' '/                                               DECLARE
ISN C032          EQUIVALENCE (BLANK,IBLANK)                                        DECLARE
                  C                                                                 SGNTST
                  C     OTHER DATA NEEDED IN THIS PROGRAM                           SGNTST
                  C                                                                 SGNTST
ISN C033          REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                        SGNTST
ISN C034          REAL*4  H0 /'H0'/, H1 /'H1'/                                      SGNTST
ISN C035          INTEGER*4  HYPLBL(2,3)  /'.EQ.', '.NE.', '.LE.', '.GT.',          SGNTST
                 1    '.GE.', '.LT.'/                                             SGNTST
ISN C036          REAL*8  XLBL /'X'/, YLBL /'Y'/                                    SGNTST
ISN C037          INTEGER*4  CURSMP,CURVAR,CURCNT,T,TIES                            SGNTST
                  C                                                                 SGNTST
                  C     FORMAT STATEMENTS                                           SGNTST
                  C                                                                 SGNTST
ISN C038        231 FORMAT ('-', T12, 'IKMS31I  UNEQUAL SAMPLE/VARIATE SIZES AFTER TRASGNTST
                   1NSFORMATION')                                                   SGNTST
ISN C039        301 FORMAT ('-', T62, 'THE SIGN TEST')                              SGNTST
ISN C040        302 FORMAT ('0', T12, A2, ': E(', A8, ') ', A4, ' E(', A8, ')')     SGNTST
ISN C041        303 FORMAT ('0', T12, 'THE SAMPLE SIZE IS',I6)                      SGNTST
ISN C042        304 FORMAT ('0', T12, 'THERE WERE', I5,                             SGNTST
                 1    ' TIES, RESULTING IN A TEST OF', I6, ' OBSERVATIONS')        SGNTST
ISN C043        305 FORMAT ('0', T12, 'VERSUS')                                     SGNTST
ISN C044        306 FORMAT ('0', T12, 'THE TEST STATISTIC IS', I6)                  SGNTST
ISN C045        307 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', H0 IS ', A8, '.')      SGNTST
ISN C046        308 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                          SGNTST
ISN 0047        309 FORMAT ('-', T58, 'END OF THE SIGN TEST')                       SGNTST
                  C                                                                 SGNTST
                  C     PRINT THE TEST TITLE                                        SGNTST
                  C                                                                 SGNTST
ISN C048          WKSZE = WKSZE                                                     SGNTST
ISN C049          WRITE (LST,301)                                                   SGNTST
                  C                                                                 SGNTST
                  C     SET A POINTER FOR WHICH TAIL IS TO BE TESTED                SGNTST
                  C                                                                 SGNTST
```

```
ISN C050            ITL = 1                                                  SGNTST
ISN C051            IF (TAIL) 11, 13, 12                                     SGNTST
ISN C052         11 ITL = 2                                                  SGNTST
ISN C053            GO TO 13                                                 SGNTST
ISN C054         12 ITL = 3                                                  SGNTST
                 C                                                          SGNTST
                 C     REPLACE ANY MISSING LABELS BY APPROPRIATE SUBSTITUTES SGNTST
                 C                                                          SGNTST
ISN C055         13 IF (LABEL(1) .EQ. BLANK) LABEL(1) = XLBL                SGNTST
ISN C057            IF (LABEL(2) .EQ. BLANK) LABEL(2) = YLBL                 SGNTST
                 C                                                          SGNTST
                 C     LIST WHAT THE NULL AND ALTERNATE HYPOTHSES ARE       SGNTST
                 C                                                          SGNTST
ISN C059            WRITE (LST,302) H0,LABEL(1),HYPLBL(1,ITL),LABEL(2)       SGNTST
ISN C060            WRITE (LST,305)                                          SGNTST
ISN C061            WRITE (LST,302) H1,LABEL(1),HYPLBL(2,ITL),LABEL(2)       SGNTST
ISN C062            CALL OFLOW(9)                                            SGNTST
                 C                                                          SGNTST
                 C     FIRST PASS OF DATA                                   SGNTST
                 C     READ THE SAMPLES AND WRITE SAMPLE 1 (VARIATE 1) ON UT2 SGNTST
                 C                                                          SGNTST
ISN C063            REWIND UT1                                               SGNTST
ISN C064            REWIND UT2                                               SGNTST
                 C                                                          SGNTST
ISN C065         14 READ (UT1,END=21) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)    SGNTST
ISN C066            IF (TYPE .EQ. 6) GO TO 16                                SGNTST
                 C                                                          SGNTST
                 C     THE SAMPLE IS BIVARIATE                              SGNTST
                 C                                                          SGNTST
ISN C068            IF (CURVAR .NE. 1) GO TO 14                              SGNTST
                 C                                                          SGNTST
                 C     WRITE SAMPLE 1 OR VARIATE 1 ONTO UT2                 SGNTST
                 C                                                          SGNTST
ISN C070         15 WRITE (UT2) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)         SGNTST
ISN C071            GO TO 14                                                 SGNTST
                 C                                                          SGNTST
                 C     THE TEST IS ON 2 MATCHED SAMPLES                     SGNTST
                 C                                                          SGNTST
ISN C072         16 IF (CURSMP .NE. 1) GO TO 14                              SGNTST
ISN C074            ITEMS = 1                                                SGNTST
ISN C075            LOCAT(4) = 0                                             SGNTST
                 C     CHECK FOR TRANSFORMATION BY SAMPLE                   SGNTST
ISN C076            IF (WHTRN .EQ. 1) CALL TRANS                             SGNTST
```

```
ISN C078            IF (ITEMS .NE. 1) GO TO 14                                  SGNTST
ISN C080            IF (OUT .EQ. 0 .OR. DOUT .NE. 2) GO TO 15                   SGNTST
                C   WRITE OUT OBSERVATIONS BY SAMPLE                            SGNTST
ISN C082            CALL WRDATA                                                 SGNTST
ISN C083            GO TO 15                                                    SGNTST
                C                                                               SGNTST
                C   END OF PASS 1 - INITIATE PASS 2 TO COMPARE SAMPLES           SGNTST
                C                                                               SGNTST
ISN C084         21 ENDFILE UT2                                                 SGNTST
ISN C085            REWIND UT1                                                  SGNTST
ISN C086            REWIND UT2                                                  SGNTST
                C                                                               SGNTST
ISN C087            T = 0                                                       SGNTST
ISN C088            TIES = 0                                                    SGNTST
ISN C089            NSZ = 0                                                     SGNTST
ISN C090         22 READ (UT1,END=41) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(2)       SGNTST
ISN C091            IF (TYPE .EQ. 6) GO TO 31                                   SGNTST
                C                                                               SGNTST
                C   BIVARIATE SAMPLE                                            SGNTST
                C                                                               SGNTST
ISN C093            IF (CURVAR .EQ. 1) GO TO 22                                 SGNTST
ISN C095            READ (UT2,END=99) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)       SGNTST
ISN C096            ITEMS = 2                                                   SGNTST
ISN C097            LOCAT(4) = 0                                                SGNTST
                C                                                               SGNTST
                C   CHECK FOR TRANSFORMATIONS OF ORGANIZED DATA                  SGNTST
                C                                                               SGNTST
ISN C098            IF (WHTRN .EQ. 1) CALL TRANS                                SGNTST
ISN C100            IF (ITEMS .NE. 2) GO TO 22                                  SGNTST
ISN C102            IF (OUT .EQ. 0 .OR. DOUT .NE. 2) GO TO 51                   SGNTST
                C   WRITE OUT THE DATA FOR THE USER                             SGNTST
ISN C104            CALL WRDATA                                                 SGNTST
ISN 0105            GO TO 51                                                    SGNTST
                C                                                               SGNTST
                C   2 MATCHED SAMPLE CASE                                       SGNTST
                C                                                               SGNTST
ISN C106         31 IF (CURSMP .EQ. 1) GO TO 22                                 SGNTST
ISN C108            ITEMS = 1                                                   SGNTST
ISN C109            X(1) = X(2)                                                 SGNTST
ISN 0110            LOCAT(4) = 0                                                SGNTST
                C   PERFORM TRANSFORMATION IF DESIRED                           SGNTST
ISN C111            IF (WHTRN .EQ. 1) CALL TRANS                                SGNTST
ISN C113            IF (ITEMS .NE. 1) GO TO 22                                  SGNTST
```

```
ISN C115            IF (OUT .EQ. 0 .OR. DOUT .NE. 2) GO TO 32                    SGNTST
                C   WRITE DATA TO SUIT THE USER                                  SGNTST
ISN C117            CALL WRDATA                                                  SGNTST
                C                                                                SGNTST
                C   MOVE THE SECOND VALUE BACK WHERE IT BELONGS AND GET THE FIRST  SGNTST
                C                                                                SGNTST
ISN C118         32 X(2) = X(1)                                                  SGNTST
ISN C119            READ (UT2,END=99) CURSMP,CURVAR,CURCNT,RDNO,IENT,X(1)        SGNTST
                C                                                                SGNTST
                C   CLASSIFY THIS OBSERVATION                                    SGNTST
                C                                                                SGNTST
ISN C120         51 NSZ = NSZ + 1                                                SGNTST
                C                                                                SGNTST
ISN C121            IF (X(1) - X(2)) 52, 53, 54                                  SGNTST
                C   FIRST SAMPLE SMALLER - COUNT IN TEST STATISTIC                SGNTST
ISN C122         52 T = T + 1                                                    SGNTST
ISN C123            GO TO 22                                                     SGNTST
                C   VALUES ARE EQUAL COUNT AS TIES THAT ARE DISCARDED           SGNTST
ISN C124         53 TIES = TIES + 1                                              SGNTST
ISN C125            GO TO 22                                                     SGNTST
                C   FIRST SAMPLE LARGER - DO NOTHING                            SGNTST
ISN C126         54 GO TO 22                                                     SGNTST
                C                                                                SGNTST
                C   END OF SECOND PASS AND TEST STATISTIC COMPUTATION            SGNTST
                C                                                                SGNTST
ISN C127         41 IF (OUT .NE. 0 .AND. DOUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT  SGNTST
ISN C129            OUT = 0                                                      SGNTST
ISN C130            WRITE (LST,303) NSZ                                          SGNTST
ISN C131            CALL OFLOW(2)                                                SGNTST
                C   CHECK THE NUMBER OF TIES                                     SGNTST
ISN C132            IF (TIES .EQ. 0) GO TO 42                                    SGNTST
ISN C134            NSZ = NSZ - TIES                                             SGNTST
ISN C135            WRITE (LST,304) TIES, NSZ                                    SGNTST
ISN C136            CALL OFLOW(2)                                                SGNTST
                C   PRINT THE VALUE OF THE TEST STATISTIC                        SGNTST
ISN C137         42 WRITE (LST,306) T                                            SGNTST
                C                                                                SGNTST
ISN C138            GO TO (61, 71, 81), ITL                                      SGNTST
                C                                                                SGNTST
                C   TWO TAIL TEST - E(1) = E(2)                                  SGNTST
                C                                                                SGNTST
ISN C139         61 T = MINO(T,NSZ-T)                                            SGNTST
ISN C140            ALPHAT = BINOM(NSZ,.5,T)                                     SGNTST
```

```
ISN C141            ALFHAT = 2. * AMIN1(ALFHAT, 1.-ALFHAT)                  SGNTST
ISN 0142            IF (ALFHAT .GT. 1.) ALFHAT = 1.                         SGNTST
ISN 0144            GO TO 91                                                SGNTST
                C                                                           SGNTST
                C     ONE TAIL TEST - E(1) <= E(2)                          SGNTST
                C                                                           SGNTST
ISN C145         71 ALFHAT = BINOM(NSZ,.5,T)                                SGNTST
ISN C146            GO TO 91                                                SGNTST
                C                                                           SGNTST
                C     ONE TAIL TEST - E(1) >= E(2)                          SGNTST
                C                                                           SGNTST
ISN C147         81 ALFHAT = BINOM(NSZ,.5,NSZ-T)                            SGNTST
                C                                                           SGNTST
                C     ACCEPT OR REJECT BASED ON USER'S ALPHA SPECIFICATION  SGNTST
                C                                                           SGNTST
ISN 0148         91 I = 1                                                   SGNTST
ISN C149            IF (ALPHA .GE. ALFHAT) I = 2                            SGNTST
ISN C151            WRITE (LST,307) ALPHA, ACCREJ(I)                        SGNTST
ISN 0152            WRITE (LST,308) ALFHAT                                  SGNTST
ISN C153            CALL OFLOW(5)                                           SGNTST
                C                                                           SGNTST
                C     END OF TEST - RETURN TO CALLING PROGRAM               SGNTST
                C                                                           SGNTST
ISN C154         98 WRITE (LST,309)                                         SGNTST
ISN C155            CALL OFLOW(3)                                           SGNTST
ISN C156            RETURN                                                  SGNTST
                C                                                           SGNTST
                C     ERROR IN READING FIRST SAMPLE SECOND TIME - PREMATURE EOF SGNTST
                C                                                           SGNTST
ISN C157         99 WRITE (LST,231)                                         SGNTST
ISN 0158            CALL OFLOW(3)                                           SGNTST
ISN C159            RC = 12                                                 SGNTST
ISN 0160            GO TO 98                                                SGNTST
                C                                                           SGNTST
ISN C161            END                                                     SGNTST

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     160 ,PROGRAM SIZE =     2498

*STATISTICS*  NO  DIAGNOSTICS GENERATED
```

****** END OF COMPILATION ******

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN C002            SUBROUTINE TAU                                                  TAU
                    C                                                               TAU
                    C     CORRELATION USING KENDALL'S TAU                           TAU
                    C                                                               TAU
                    C     THIS SUBROUTINE CALCULATES KENDALL'S RANK CORRELATION      TAU
                    C     COEFFICIENT, COMMONLY REFERRED TO AS KENDALL'S TAU.  THE   TAU
                    C     VALUE OF TAU IS ALWAYS BETWEEN -1 AND +1 AND CAN BE       TAU
                    C     CONSIDERED A MEASURE OF CORRELATION BETWEEN THE TWO HALVES TAU
                    C     OF A BIVARIATE SAMPLE.  IT CAN BE USED TO TEST HYPOTHESES  TAU
                    C     OF NO CORRELATION VERSUS SOME, POSITIVE, OR NEGATIVE       TAU
                    C     CORRELATION.  THE CRITICAL REGION IS DETERMINED FROM A     TAU
                    C     LARGE SAMPLE APPROXIMATION AND CANNOT BE CONSIDERED VALID  TAU
                    C     FOR SAMPLES SMALLER THAN 40 OBSERVATIONS.                  TAU
                    C                                                               TAU
                    C     FOR DETIALS ON KENDALL'S TAU, SEE CONOVER, W.J., PRACTICAL TAU
                    C     NONPARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 393-399.      TAU
                    C                                                               TAU
                    C                                                               TAU
                    C     COMMON BLOCKS                                             TAU
                    C                                                               TAU
ISN C003            COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                    DECLARE
ISN C004            INTEGER*4  ROWS,COLS                                            DECLARE
ISN C005            COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                        DECLARE
ISN C006            INTEGER*4  DISTN,WHTRN                                          DECLARE
ISN C007            COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                     DECLARE
ISN C008            REAL*8  BUFFER                                                  DECLARE
ISN C009            INTEGER*4  EOD,CDCOL,BUFNO                                      DECLARE
ISN C010            COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD           DECLARE
ISN C011            REAL*8   FMT                                                    DECLARE
ISN C012            INTEGER*4  DFORM,DOUT,RCBCD                                     DECLARE
ISN C013            COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),     DECLARE
                   1    VALUE(40),IVALUE(40),RC                                     DECLARE
ISN C014            REAL*8   PARM,SUBPRM                                            DECLARE
ISN C015            INTEGER*4  RC                                                   DECLARE
ISN C016            COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                     DECLARE
ISN C017            INTEGER*4  TAIL                                                 DECLARE
ISN C018            LOGICAL*4  TESTS,DOALL                                          DECLARE
ISN C019            COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),         DECLARE
                   1    DATE,TIME,PAGE,LINE                                         DECLARE
ISN C020            REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                DECLARE
ISN C021            INTEGER*4  PAGE                                                 DECLARE
```

```
ISN C022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
ISN C023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE     DECLARE
ISN C024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C025          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT               DECLARE
ISN C026          COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                          DECLARE
ISN C027          INTEGER*4  ITEMS,LOCAT,RDNO                                       DECLARE
ISN C028          COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                      DECLARE
ISN C029          INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                            DECLARE
ISN C030          EQUIVALENCE (WORK(1),IWRK(1))                                     DECLARE
ISN C031          REAL*8  BLANK /' '/                                               DECLARE
ISN C032          EQUIVALENCE (BLANK,IBLANK)                                        DECLARE
                  C                                                                 TAU
                  C     OTHER DATA USED                                             TAU
ISN C033          REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                        TAU
ISN C034          REAL*8  PCOR /'POSITIVE'/, NCOR /'NEGATIVE'/                      TAU
ISN 0035          REAL*8  DX                                                        TAU
ISN C036          INTEGER*4  CURSMP,CURVAR,CURCNT                                   TAU
ISN C037          REAL*4  RECORD(6)                                                 TAU
ISN C038          EQUIVALENCE (RECORD(1),CURSMP), (RECORD(2),CURVAR),               TAU
                 1     (RECORD(3),CURCNT), (RECORD(4),XOBS), (RECORD(5),YOBS),     TAU
                 2     (RECORD(6),OBS)                                             TAU
ISN C039          REAL*8  XLBL /'X'/, YLBL /'Y'/                                    TAU
                  C                                                                 TAU
                  C     FORMAT STATEMENTS                                           TAU
ISN C040        231 FORMAT ('-', T12, 'IKMS31I  UNEQUAL SAMPLE/VARIATE SIZES AFTER TRATAU
                   1NSFORMATION')                                                   TAU
ISN C041        234 FORMAT ('-', T12, 'IKMS34I  INVALID SAMPLE SIZE(S)')            TAU
ISN C042        290 FORMAT ('-', T12, 'IKMS90I  ***** BECAUSE OF SAMPLE SIZES, ALPHA HTAU
                   1AT MAY NOT BE CORRECT.  VERIFY WITH TABLES. *****')             TAU
ISN C043        301 FORMAT ('-', T33, 'TEST OF CORRELATION USING KENDALL''S RANK CORRETAU
                   1LATION COEFFICIENT, TAU')                                       TAU
ISN C044        302 FORMAT ('0', T12, 'H0: ', A8, ' AND ', A8,                      TAU
                   1    ' ARE MUTUALLY INDEPENDENT'/ '0', T12, 'VERSUS')         TAU
ISN C045        303 FORMAT ('0', T12, 'H1: ', A8, ' AND ', A8,                      TAU
                   1    ' ARE CORRELATED WITH EACH OTHER IN SOME MANNER')        TAU
ISN 0046        304 FORMAT ('0', T12, 'H1: ', A8, ' IS ', A8,                       TAU
                   1    'LY CORRELATED WITH ', A8)                               TAU
ISN C047        305 FORMAT ('0', T12, 'THE SAMPLE CONSISTS OF', I6, ' OBSERVATIONS') TAU
ISN 0048        306 FORMAT ('0', T12, 'THE NUMBER OF CONCORDANT PAIRS IS',          TAU
                   1    I5/ '0', T12,                                            TAU
                   2    'THE NUMBER OF DISCORDANT PAIRS IS', I5/ '0', T12,       TAU
                   3    'THE TEST STATISTIC IS', I5/ '0', T12,                   TAU
                   4    'THE CORRELATION COEFFICIENT (TAU) IS', F7.3)            TAU
```

```
ISN C049        307 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')      TAU
ISN C050        308 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                          TAU
ISN C051        309 FORMAT ('-', T50, 'END OF CORRELATION WITH KENDALL''S TAU')     TAU
                C                                                                   TAU
                C     WRITE OUT HYPOTHESES AFTER CORRECTING LABELS                  TAU
                C                                                                   TAU
ISN C052            WRITE (LST,301)                                                 TAU
ISN C053            IF (LABEL(1) .EQ. BLANK) LABEL(1) = XLBL                        TAU
ISN C055            IF (LABEL(2) .EQ. BLANK) LABEL(2) = YLBL                        TAU
ISN C057            ITL = 1                                                         TAU
ISN C058            IF (TAIL) 11, 13, 12                                            TAU
ISN C059         11 ITL = 2                                                         TAU
ISN C060            GO TO 13                                                        TAU
ISN C061         12 ITL = 3                                                         TAU
                C                                                                   TAU
ISN C062         13 WRITE (LST,302) LABEL(1),LABEL(2)                               TAU
ISN C063            GO TO (14, 15, 16), ITL                                         TAU
                C                                                                   TAU
ISN C064         14 WRITE (LST,303) LABEL(1),LABEL(2)                               TAU
ISN C065            GO TO 17                                                        TAU
ISN C066         15 WRITE (LST,304) LABEL(1),NCOR,LABEL(2)                          TAU
ISN C067            GO TO 17                                                        TAU
ISN C068         16 WRITE (LST,304) LABEL(1),PCOR,LABEL(2)                          TAU
                C                                                                   TAU
ISN C069         17 CALL OFLOW(9)                                                   TAU
ISN C070            SUBS(1) = 0                                                     TAU
ISN C071            SUBS(2) = 0                                                     TAU
ISN C072            REWIND UT1                                                      TAU
ISN C073            REWIND UT2                                                      TAU
                C                                                                   TAU
                C     PASS 1 OF DATA - EXTRACT VARIATE 1                            TAU
                C                                                                   TAU
ISN C074         21 READ (UT1,END=23) RECORD                                        TAU
ISN C075            IF (CURVAR .NE. 1) GO TO 21                                     TAU
ISN C077            WRITE (UT2) RECORD                                              TAU
ISN C078            GO TO 21                                                        TAU
                C                                                                   TAU
                C     SECOND PASS - PAIR UP THE VARIATES                            TAU
                C                                                                   TAU
ISN C079         23 ENDFILE UT2                                                     TAU
ISN C080            REWIND UT1                                                      TAU
ISN C081            REWIND UT2                                                      TAU
ISN 0082            REWIND UT4                                                      TAU
```

```
ISN C083            ISZ = 0                                                          TAU
                C                                                                    TAU
ISN C084           24 READ (UT1,END=26) RECORD                                       TAU
ISN C085            IF (CURVAR .EQ. 1) GO TO 24                                      TAU
                C                                                                    TAU
ISN C087            X(2) = OBS                                                       TAU
ISN C088            READ (UT2,END=1231) RECORD                                       TAU
ISN C089            X(1) = OBS                                                       TAU
ISN C090            ITEMS = 2                                                        TAU
ISN C091            LOCAT(4) = 0                                                     TAU
ISN C092            IF (WHTRN .EQ. 1) CALL TRANS                                     TAU
ISN C094            IF (ITEMS .NE. 2) GO TO 24                                       TAU
ISN C096            IF (OUT .NE. 0 .AND. COUT .EQ. 2) CALL WRDATA                    TAU
                C                                                                    TAU
                C     WRITE OUT THE COMBINED OBSERVATION                             TAU
                C                                                                    TAU
ISN C098            XOBS = X(1)                                                      TAU
ISN C099            YOBS = X(2)                                                      TAU
ISN C100            OBS = XOBS                                                       TAU
ISN C101            ISZ = ISZ + 1                                                    TAU
ISN C102            WRITE (UT4) RECORD                                               TAU
ISN C103            GO TO 24                                                         TAU
                C                                                                    TAU
                C     END OF PASS 2                                                  TAU
                C                                                                    TAU
ISN C104           26 IF (OUT .NE. 0 .AND. COUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT TAU
ISN C106            OUT = 0                                                          TAU
ISN C107            ENDFILE UT4                                                      TAU
                C                                                                    TAU
                C     SORT THE DATA ON THE X OBSERVATIONS                            TAU
                C                                                                    TAU
ISN C108            CALL RANKER (UT2,UT4,LT2,UT3,J)                                  TAU
ISN C109            IF (RC .NE. 0) GO TO 92                                          TAU
                C                                                                    TAU
                C     INITIALIZE FOR COUNT                                           TAU
                C                                                                    TAU
ISN C111            NC = 0                                                           TAU
ISN C112            ND = 0                                                           TAU
ISN C113            LIM = WKSZE / 2                                                  TAU
ISN C114            ISKP = 0                                                         TAU
ISN C115            GO TO 33                                                         TAU
                C                                                                    TAU
                C     BYPASS PREVIOUSLY PROCESSED RECORDS                            TAU
```

```
                C                                                                TAU
ISN C116           31 REWIND UT2                                                 TAU
ISN C117              DO 32 I = 1, ISKP                                          TAU
ISN C118           32 READ (UT2,END=1234) RECORD                                 TAU
                C                                                                TAU
                C     READ RECORDS AND FILL WORK AREA                            TAU
                C                                                                TAU
ISN C119           33 IRD = 0                                                    TAU
ISN C120              INDX = 0                                                   TAU
ISN C121              IEND = 0                                                   TAU
                C                                                                TAU
ISN C122           34 READ (UT2,END=35) RECCRD                                   TAU
ISN C123              IRD = IRD + 1                                              TAU
ISN C124              WORK(INDX+1) = XOBS                                        TAU
ISN C125              WORK(INDX+2) = YOBS                                        TAU
ISN C126              INDX = INDX + 2                                            TAU
ISN C127              IF (IRD .LT. LIM) GO TO 34                                 TAU
ISN C129              GO TO 36                                                   TAU
                C                                                                TAU
                C     END OF FILE - SET FLAG FOR LAST TIME                       TAU
                C                                                                TAU
ISN C130           35 IEND = 1                                                   TAU
ISN C131              IF (IRD .EQ. 1) GO TO 51                                   TAU
                C                                                                TAU
                C     PROCESS CONTENTS FOR WORK AREA                             TAU
                C                                                                TAU
ISN C133           36 ISKP = ISKP + IRD                                          TAU
ISN C134              IRD = IRD * 2                                              TAU
                C                                                                TAU
                C     COMPARE WHAT'S IN CORE TO WHAT'S IN CORE                   TAU
                C                                                                TAU
ISN C135              IRD1 = IRD - 2                                             TAU
ISN C136              DO 41 I1 = 2, IRD1, 2                                      TAU
ISN C137              IX = I1 + 2                                                TAU
ISN C138              DO 41 I2 = IX, IRD, 2                                      TAU
                C                                                                TAU
ISN C139              IF (WORK(I1-1) - WORK(I2-1)) 37, 41, 38                    TAU
ISN C140           37 IF (WORK(I1) - WORK(I2)) 39, 41, 40                        TAU
ISN C141           38 IF (WORK(I1) - WORK(I2)) 40, 41, 39                        TAU
                C     CONCORDANT PAIR                                            TAU
ISN C142           39 NC = NC + 1                                                TAU
ISN C143              GO TO 41                                                   TAU
                C     DISCORDANT PAIR                                            TAU
```

```
ISN 0144          40 ND = ND + 1                                                TAU
                C                                                               TAU
ISN 0145          41 CONTINUE                                                   TAU
                C                                                               TAU
                C     NOW COMPARE IN CORE TO WHAT'S LEFT ON DISK                TAU
                C                                                               TAU
ISN 0146             IF (IEND .EQ. 1) GO TO 51                                  TAU
ISN 0148          43 READ (UT2,END=49) RECORD                                   TAU
                C                                                               TAU
ISN 0149             DO 48 I1 = 2, IRD, 2                                       TAU
                C                                                               TAU
ISN 0150             IF (WORK(I1-1) - XOBS)  44, 48, 45                         TAU
ISN 0151          44 IF (WORK(I1) - YOBS) 46, 48, 47                            TAU
ISN 0152          45 IF (WORK(I1) - YOBS) 47, 48, 46                            TAU
                C     CONCORDANT PAIR                                           TAU
ISN 0153          46 NC = NC + 1                                                TAU
ISN 0154             GO TO 48                                                   TAU
                C     DISCORDANT PAIR                                           TAU
ISN 0155          47 ND = ND + 1                                                TAU
                C                                                               TAU
ISN 0156          48 CONTINUE                                                   TAU
ISN 0157             GO TO 43                                                   TAU
                C                                                               TAU
                C     END OF MATCHING CORE TO INTERMEDIATE STORAGE              TAU
                C                                                               TAU
ISN 0158          49 GO TO 31                                                   TAU
                C                                                               TAU
                C     END OF CONCORDANT/DISCORDANT COUNTING                     TAU
                C                                                               TAU
ISN 0159          51 T = NC - ND                                                TAU
ISN 0160             IT = T                                                     TAU
ISN 0161             D = ISZ * (ISZ - 1) / 2                                    TAU
ISN 0162             TAUR = T / D                                               TAU
                C                                                               TAU
ISN 0163             WRITE (LST,306) NC,ND,IT,TAUR                              TAU
ISN 0164             CALL OFLOW(8)                                              TAU
ISN 0165             DX = ISZ * (ISZ - 1) * (2 * ISZ + 5)                       TAU
ISN 0166             DX = DX / 18.DO                                            TAU
ISN 0167             ADX = T / DX                                               TAU
                C                                                               TAU
ISN 0168             GO TO (61, 71, 81), ITL                                    TAU
                C                                                               TAU
                C     2 TAIL CORRELATION TEST                                   TAU
```

```
                C                                                                TAU
ISN C169           61 ALFHAT = ANDTR(ADX)                                        TAU
ISN C170              ALFHAT = 2. * AMIN1(ALFHAT, 1.-ALFHAT)                     TAU
ISN C171              IF (ALFHAT .GT. 1.) ALFHAT = 1.                            TAU
ISN C173              GO TO 91                                                   TAU
                C                                                                TAU
                C     NEGATIVE CORRELATION                                       TAU
                C                                                                TAU
ISN C174           71 ALFHAT = ANDTR(ADX)                                        TAU
ISN C175              GO TO 91                                                   TAU
                C                                                                TAU
                C     POSITIVE CORRELATION                                       TAU
                C                                                                TAU
ISN C176           81 ALFHAT = ANDTR(-ADX)                                       TAU
                C                                                                TAU
                C     DETERMINE THE SIGNIFICANCE OF THE CORRELATION              TAU
                C                                                                TAU
ISN C177           91 I = 1                                                      TAU
ISN C178              IF (ALPHA .GT. ALFHAT) I = 2                               TAU
ISN C180              WRITE (LST,307) ALPHA,ACCREJ(I)                            TAU
ISN C181              WRITE (LST,308) ALFHAT                                     TAU
ISN C182              CALL OFLOW(5)                                              TAU
                C                                                                TAU
                C     CHECK FOR SMALL SAMPLE                                     TAU
                C                                                                TAU
ISN C183              IF (ISZ .GT. 40) GO TO 92                                  TAU
ISN C185              WRITE (LST,290)                                            TAU
ISN C186              CALL OFLOW(3)                                              TAU
                C                                                                TAU
                C     END OF KENDALL'S CORRELATION                               TAU
                C                                                                TAU
ISN C187           92 WRITE (LST,309)                                            TAU
ISN C188              CALL OFLOW(3)                                              TAU
ISN C189              RETURN                                                     TAU
                C                                                                TAU
                C     ERRORS THAT MIGHT HAPPEN                                   TAU
                C                                                                TAU
ISN C190         1231 WRITE (LST,231)                                            TAU
ISN C191              GO TO 99                                                   TAU
                C                                                                TAU
ISN C192         1234 WRITE (LST,234)                                            TAU
                C                                                                TAU
ISN C193           99 CALL OFLOW(3)                                              TAU
```

```
   ISN 0194            RC = 12                                                  TAU
   ISN 0195            GO TO 92                                                 TAU
                  C                                                             TAU
   ISN 0196            END                                                      TAU

*OPTIONS IN EFFECT*       NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*      SOURCE STATEMENTS =      195 ,PROGRAM SIZE =      3622

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
  ISN 0002            SUBROUTINE TRANS                                                    TRANS
                      C                                                                   TRANS
                      C     DATA TRANSFORMATION SUBROUTINE FOR CNPSTP                     TRANS
                      C                                                                   TRANS
                      C     DATA TO BE TRANSFORMED IS PASSED TO THIS SUBROUTINE IN        TRANS
                      C     THE COMMON BLOCK /USER/.                                      TRANS
                      C               ITEMS IS THE NUMBER OF DATA ITEMS IN THE X ARRAY.   TRANS
                      C               X CONTAINS THE DATA ITEM(S).                        TRANS
                      C               LOCAT(1),(2),(3) CONTAIN THE LOCATION SUBSCRIPTS    TRANS
                      C                    ASSOCIATED WITH THESE ITEMS.                   TRANS
                      C               LOCAT(4) CONTAINS THE NUMBER OF VALID SUBSCRIPTS     TRANS
                      C                    IN LOCAT(1),(2),(3).                           TRANS
                      C               RDNO CONTAINS THE NUMBER OF THE READ THAT INPUTTED  TRANS
                      C                    THE DATA IN THE X ARRAY.                       TRANS
                      C                                                                   TRANS
                      C     TO DELETE THE DATA PASSED TO THIS ROUTINE FROM ANY FURTHER    TRANS
                      C     CONSIDERATION, THIS SUBROUTINE MUST SET ITEMS TO ZERO.        TRANS
                      C     TO INCREASE OR DECREASE THE NUMBER OF ITEMS IN THE X ARRAY,   TRANS
                      C     THIS SUBROUTINE MUST CHANGE ITEMS CORRESPONDINGLY AND         TRANS
                      C     PLACE ALL X VALUES TO BE KEPT IN THE FIRST THROUGH 'ITEMS'-TH TRANS
                      C     POSITIONS OF X.                                               TRANS
                      C                                                                   TRANS
                      C                                                                   TRANS
                      C     COMMON BLOCKS                                                 TRANS
                      C                                                                   TRANS
  ISN 0003            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                            TRANS
  ISN 0004            INTEGER*4  ITEMS,LOCAT,RDNO                                         TRANS
                      C                                                                   TRANS
                      C     ENTER THE SUBROUTINE WITH A FLOURISH                          TRANS
                      C                                                                   TRANS
                      C                                                                   TRANS
                      C     LEAVE THE SAME WAY                                            TRANS
                      C                                                                   TRANS
  ISN 0005            RETURN                                                              TRANS
                      C                                                                   TRANS
  ISN 0006            END                                                                 TRANS

*OPTIONS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =       5 ,PROGRAM SIZE =      168
```

```
*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE TREND                                                TREND
              C                                                                     TREND
              C    TEST OF TREND USING SPREAMAN'S RHO                               TREND
              C                                                                     TREND
              C    THIS SUBROUTINE USES SPEARMAN'S RANK CORRELATION COEFFICIENT     TREND
              C    AS A MEASURE OF TREND ON A SINGLE RANDOM SAMPLE.  A SECOND       TREND
              C    VALUE IS PROVIDED FOR EACH OBSERVATION WHICH CONSISTS            TREND
              C    OF THE OBSERVATION NUMBER (1,2,3,...,N).  THIS FORMS A           TREND
              C    BIVARIATE POPULATION ON WHICH SPEARMAN'S RHO IS CALCULATED       TREND
              C    AS A MEASURE OF TREND.                                           TREND
              C                                                                     TREND
              C    FOR DETAILS ON USING SPEARMAN'S RHO AS A MEASURE OF TREND,       TREND
              C    SEE CONOVER, W.J., PRACTICAL NONPARAMETRIC STATISTICS,           TREND
              C    ROUGH DRAFT, PAGES 396-398                                       TREND
              C                                                                     TREND
              C                                                                     TREND
              C    COMMON BLOCKS                                                    TREND
              C                                                                     TREND
 ISN 0003          COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                     DECLARE
 ISN 0004          INTEGER*4  ROWS,COLS                                             DECLARE
 ISN 0005          COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                         DECLARE
 ISN 0006          INTEGER*4  DISTN,WHTRN                                           DECLARE
 ISN 0007          COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                      DECLARE
 ISN 0008          REAL*8  BUFFER                                                   DECLARE
 ISN 0009          INTEGER*4  EOD,CDCOL,BUFNO                                       DECLARE
 ISN 0010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD            DECLARE
 ISN 0011          REAL*8   FMT                                                     DECLARE
 ISN 0012          INTEGER*4  DFORM,DOUT,RCBCD                                      DECLARE
 ISN 0013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),      DECLARE
                  1    VALUE(40),IVALUE(40),RC                                      DECLARE
 ISN 0014          REAL*8   PARM,SUBPRM                                             DECLARE
 ISN 0015          INTEGER*4  RC                                                    DECLARE
 ISN 0016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                      DECLARE
 ISN 0017          INTEGER*4  TAIL                                                  DECLARE
 ISN 0018          LOGICAL*4  TESTS,DOALL                                           DECLARE
 ISN 0019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),          DECLARE
                  1    DATE,TIME,PAGE,LINE                                          DECLARE
 ISN 0020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                 DECLARE
 ISN 0021          INTEGER*4  PAGE                                                  DECLARE
 ISN 0022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40) DECLARE
 ISN 0023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE    DECLARE
```

```
ISN C024            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C025            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                DECLARE
ISN C026            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                           DECLARE
ISN C027            INTEGER*4  ITEMS,LOCAT,RDNO                                        DECLARE
ISN C028            COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                       DECLARE
ISN C029            INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                             DECLARE
ISN C030            EQUIVALENCE (WORK(1),IWRK(1))                                      DECLARE
ISN C031            REAL*8  BLANK /' '/                                                DECLARE
ISN C032            EQUIVALENCE (BLANK,IBLANK)                                         DECLARE
                  C                                                                    TREND
                  C     OTHER DATA NEEDED                                              TREND
                  C                                                                    TREND
ISN C033            REAL*8 ACCREJ(2) /'ACCEPTED', 'REJECTED'/                          TREND
ISN C034            REAL*8  HYPLBL(3) /'DEFINITE', 'NEGATIVE', 'POSITIVE'/             TREND
ISN C035            REAL*8  DX                                                         TREND
ISN C036            REAL*4  RECORD(6)                                                  TREND
ISN C037            INTEGER*4  CURSMP,CURVAR,CURCNT,XTIES,XPLC                         TREND
ISN C038            EQUIVALENCE (RECORD(1),CURSMP,XRANK), (RECORD(2),CURVAR,YRANK),    TREND
                   1     (RECORD(3),CURCNT,XVALUE), (RECORD(4),YVALUE), (RECORD(6),OBS)TREND
                  C                                                                    TREND
                  C     FORMAT STATEMENTS                                              TREND
ISN C039          234 FORMAT ('-', T12, 'IKMS341  INVALID SAMPLE SIZE(S)')             TREND
ISN C040          236 FORMAT ('-', T12, 'IKMS361  TOO MANY TIED VALUES FOR THIS PROGRAM TREND
                     1TO HANDLE')                                                      TREND
ISN C041          290 FORMAT ('-', T12, 'IKMS901  ***** BECAUSE OF SAMPLE SIZES, ALPHA HTREND
                     1AT MAY NOT BE CORRECT.  VERIFY WITH TABLES. *****')              TREND
ISN C042          301 FORMAT ('-', T38, 'TEST OF TREND USING SPEARMAN''S RANK CORRELATIOTREND
                     1N COEFFICIENT')                                                  TREND
ISN C043          302 FORMAT ('0', T12, 'H0: THERE IS NO SIGNIFICANT TREND IN THE DATA'/TREND
                     1     '0', T12, 'VERSUS'/ '0', T12,                               TREND
                     2     'H1: THERE IS A ', A8, ' TREND IN THE OBSERVED DATA')       TREND
ISN C044          303 FORMAT ('-', T12, 'THE SAMPLE IS OF SIZE', I6)                   TREND
ISN C045          305 FORMAT ('0', T12, 'THE CORRELATION COEFFICIENT (RHO) IS', F7.3)  TREND
ISN C046          306 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')       TREND
ISN C047          307 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                           TREND
ISN C048          308 FORMAT ('-', T56, 'END OF THE TEST OF TREND')                    TREND
ISN C049          309 FORMAT ('0', T12, 'THERE WAS A TOTAL OF', I5, ' TIES IN',        TREND
                     1     I3, ' PLACES')                                            TREND
                  C                                                                    TREND
                  C     START BY PRINTING THE TITLE AND HYPOTHESES                     TREND
                  C                                                                    TREND
ISN C050            WRITE (LST,301)                                                    TREND
                  C                                                                    TREND
```

```
ISN C051            ITL = 1                                                    TREND
ISN C052            IF (TAIL) 11, 13, 12                                       TREND
ISN C053         11 ITL = 1                                                    TREND
ISN C054            GO TO 13                                                   TREND
ISN C055         12 ITL = 2                                                    TREND
                 C                                                             TREND
ISN C056         13 WRITE (LST,302) HYPLBL(ITL)                                TREND
ISN C057            CALL OFLOW(9)                                              TREND
                 C                                                             TREND
ISN C058            SUBS(1) = 0                                                TREND
ISN C059            SUBS(2) = 0                                                TREND
ISN C060            REWIND UT1                                                 TREND
ISN 0061            REWIND UT2                                                 TREND
ISN C062            ISZ = 0                                                    TREND
ISN C063            RANK = 0.                                                  TREND
                 C                                                             TREND
                 C     READ THE RECORDS AND COUNT THE AND TRANSFORM THEM AND ETC.  TREND
                 C                                                             TREND
ISN C064         14 READ (UT1,END=16) RECORD                                   TREND
ISN C065            ITEMS = 1                                                  TREND
ISN C066            LOCAT(4) = 0                                               TREND
ISN C067            X(1) = OBS                                                 TREND
ISN C068            IF (WHTRN .EQ. 1) CALL TRANS                               TREND
ISN C070            IF (ITEMS .LT. 1) GO TO 14                                 TREND
ISN C072            IF (OUT .NE. 0 .AND. COUT .EQ. 2) CALL WRDATA              TREND
ISN C074            DO 15 I = 1, ITEMS                                         TREND
ISN C075            OBS = X(I)                                                 TREND
                 C     ASSIGN RANKS OF THE Y'S                                 TREND
ISN C076            RANK = RANK + 1.                                           TREND
ISN C077            YRANK = RANK                                               TREND
ISN C078            YVALUE = RANK                                              TREND
ISN C079            WRITE (UT2) RECORD                                         TREND
ISN C080         15 ISZ = ISZ + 1                                              TREND
ISN C081            GO TO 14                                                   TREND
                 C                                                             TREND
                 C     END OF FIRST PASS OF THE DATA                           TREND
                 C                                                             TREND
ISN C082         16 ENDFILE UT2                                                TREND
ISN C083            IF (OUT .NE. 0 .AND. COUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT  TREND
ISN C085            OUT = 0                                                    TREND
ISN C086            WRITE (LST,303) ISZ                                        TREND
ISN C087            CALL OFLOW(3)                                              TREND
ISN C088            IF (ISZ .LE. 3) GO TO 1234                                 TREND
```

```
ISN C090            CALL RANKER (UT4,UT2,LT4,UT3,J)                    TREND
ISN C091            IF (RC .NE. 0) GO TO 92                            TREND
                C                                                      TREND
                C   ASSIGN RANKS TO THE X'S                            TREND
                C                                                      TREND
ISN C093            RANK = 0.                                          TREND
ISN C094            T = 0.                                             TREND
ISN C095            XTIES = 0                                          TREND
ISN C096            XPLC = 0                                           TREND
ISN C097            REWIND UT4                                         TREND
ISN C098            READ (UT4,END=1234) RECORD                         TREND
                C                                                      TREND
                C   SAVE THE CURRENT OBSERVATION AND GET THE NEXT ONE   TREND
                C                                                      TREND
ISN C099         21 DO 22 I = 1, 6                                     TREND
ISN C100         22 DATA(I) = RECORD(I)                                TREND
ISN C101            RANK = RANK + 1.                                   TREND
                C                                                      TREND
ISN C102            READ (UT4,END=23) RECORD                           TREND
ISN 0103            IF (DATA(6) .EQ. OBS) GO TO 26                     TREND
                C                                                      TREND
                C   NON-TIED OBSERVATION - WRITE OUT FOR Y SORT        TREND
                C                                                      TREND
ISN C105            T = T + (RANK - DATA(2)) ** 2                      TREND
ISN C106            GO TO 21                                           TREND
                C                                                      TREND
                C   END OF FILE - RANK LAST OBSERVATION                 TREND
                C                                                      TREND
ISN 0107         23 T = T + (RANK - DATA(2)) ** 2                      TREND
ISN C108            GO TO 61                                           TREND
                C                                                      TREND
                C   PROCESS TIED OBSERVATIONS                         TREND
                C                                                      TREND
ISN 0109         26 XPLC = XPLC + 1                                    TREND
ISN C110            LIM = (WKSZE / 6) * 6                              TREND
ISN 0111            IPTR = 0                                           TREND
ISN 0112            DO 27 I = 1, 6                                     TREND
ISN 0113         27 WORK(I) = DATA(I)                                  TREND
                C                                                      TREND
                C   SAVE THE NEXT TIED VALUE INFC                      TREND
                C                                                      TREND
ISN C114         28 IPTR = IPTR + 6                                    TREND
ISN C115            IF (IPTR .EQ. LIM) GO TO 1236                      TREND
```

```
ISN C117            DO 29 I = 1, 6                                          TREND
ISN C118         29 WORK(IPTR+I) = RECORD(I)                                TREND
                 C                                                          TREND
                 C     LOOK FOR NEXT VALUE                                  TREND
                 C                                                          TREND
ISN 0119            READ (UT4,END=33) RECORD                                TREND
ISN 0120            IF (DATA(6) .EQ. OBS) GO TO 28                          TREND
                 C                                                          TREND
                 C     RANK THIS SET OF TIES                                TREND
                 C                                                          TREND
ISN C122            J = (IPTR + 6) / 6                                      TREND
ISN C123            XTIES = XTIES + J                                       TREND
ISN 0124            Z = 0.                                                  TREND
                 C                                                          TREND
ISN C125            DO 30 I = 1, J                                          TREND
ISN C126            Z = Z + RANK                                            TREND
ISN C127         30 RANK = RANK + 1.                                        TREND
ISN C128            Z = Z / J                                               TREND
ISN C129            JPTR = 0                                                TREND
                 C                                                          TREND
ISN C130         31 T = T + (Z - WORK(JPTR+2)) ** 2                         TREND
ISN C131            JPTR = JPTR + 6                                         TREND
ISN C132            IF (JPTR .LE. IPTR) GO TO 31                            TREND
ISN C134            RANK = RANK - 1.                                        TREND
ISN 0135            GO TO 21                                                TREND
                 C                                                          TREND
                 C     END OF FILE ON A TIE                                 TREND
                 C                                                          TREND
ISN C136         33 J = (IPTR + 6) / 6                                      TREND
ISN C137            XTIES = XTIES + J                                       TREND
ISN C138            Z = 0.                                                  TREND
ISN C139            DO 34 I = 1, J                                          TREND
ISN 0140            Z = Z + RANK                                            TREND
ISN C141         34 RANK = RANK + 1.                                        TREND
ISN C142            Z = Z / J                                               TREND
ISN 0143            JPTR = 0                                                TREND
                 C                                                          TREND
ISN 0144         35 T = T + (Z - WORK(JPTR+2)) ** 2                         TREND
ISN C145            JPTR = JPTR + 6                                         TREND
ISN 0146            IF (JPTR .LE. IPTR) GO TO 35                            TREND
                 C                                                          TREND
                 C     END OF T CALCULATION                                 TREND
                 C                                                          TREND
```

```
ISN C148          61 IF (XTIES .EQ. 0) GO TO 63                                  TREND
ISN C150             WRITE (LST,309) XTIES,XPLC                                  TREND
ISN C151             CALL OFLOW(2)                                               TREND
                  C                                                              TREND
                  C     NOW CALCULATE RHO                                        TREND
                  C                                                              TREND
ISN C152          63 DN = ISZ * (ISZ * ISZ - 1)                                  TREND
ISN C153             R = 1. - 6. * T / DN                                        TREND
ISN C154             WRITE (LST,305) R                                           TREND
ISN C155             CALL OFLOW(2)                                               TREND
ISN C156             DX = ISZ - 1                                                TREND
ISN C157             ADX = DSQRT(DX)                                             TREND
ISN C158             GO TO (66, 71, 76), ITL                                     TREND
                  C                                                              TREND
                  C     2 TAILED TREND TEST                                      TREND
                  C                                                              TREND
ISN C159          66 ALFHAT = ANDTR(ADX*R)                                       TREND
ISN C160             ALFHAT = 2. * AMIN1(ALFHAT, 1.-ALFHAT)                      TREND
ISN C161             IF (ALFHAT .GT. 1.) ALFHAT = 1.                             TREND
ISN C163             GO TO 91                                                    TREND
                  C                                                              TREND
                  C     NEGATIVE TREND TEST SPECIFIED                            TREND
                  C                                                              TREND
ISN C164          71 ALFHAT = ANDTR(ADX*R)                                       TREND
ISN C165             GO TO 91                                                    TREND
                  C                                                              TREND
                  C     POSITIVE TREND TEST                                      TREND
                  C                                                              TREND
ISN C166          76 ALFHAT = ANDTR(-ADX*R)                                      TREND
                  C                                                              TREND
                  C     REACH CONCLUSION ABOUT TREND OF DATA                     TREND
                  C                                                              TREND
ISN C167          91 I = 1                                                       TREND
ISN C168             IF (ALPHA .GT. ALFHAT) I = 2                                TREND
ISN C170             WRITE (LST,306) ALPHA,ACCREJ(I)                             TREND
ISN C171             WRITE (LST,307) ALFHAT                                      TREND
ISN C172             CALL OFLOW(5)                                               TREND
                  C                                                              TREND
                  C     CHECK IF SAMPLE SIZE MIGHT BE TOO SMALL                  TREND
                  C                                                              TREND
ISN C173             IF (ISZ .GE. 30) GO TO 92                                   TREND
ISN C175             WRITE (LST,290)                                             TREND
ISN C176             CALL OFLOW(3)                                               TREND
```

```
                C                                                                TREND
                C     END OF THE TREND TEST                                      TREND
                C                                                                TREND
   ISN C177        92 WRITE (LST,308)                                            TREND
   ISN C178           CALL OFLOW(3)                                              TREND
   ISN C179           RETURN                                                     TREND
                C                                                                TREND
                C     POSSIBLE ERROR CONDITIONS                                  TREND
                C                                                                TREND
   ISN 0180      1234 WRITE (LST,234)                                            TREND
   ISN C181           GO TO 99                                                   TREND
                C                                                                TREND
   ISN 0182      1236 WRITE (LST,236)                                            TREND
                C                                                                TREND
   ISN C183        99 CALL OFLOW(3)                                             TREND
   ISN 0184           RC = 12                                                    TREND
   ISN 0185           GO TO 92                                                   TREND
                C                                                                TREND
   ISN C186           END                                                        TREND

*OPTICNS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =     185 ,PROGRAM SIZE =     3230

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE WILCOX                                                WILCOX
              C                                                                     WILCOX
              C    THE WILCOXON SIGNED RANKS TEST                                   WILCOX
              C                                                                     WILCOX
              C    THE WILCOXON SIGNED RANKS TEST IS USED TO TEST A BIVARIATE       WILCOX
              C    SAMPLE OR 2 MATCHED SAMPLES AS TO IF THE FIRST VARIATE           WILCOX
              C    IS EQUAL TO, GREATER THAN, OR LESS THAN THE SECOND VARIATE,      WILCOX
              C    HENCE TESTING WHETHER OR NOT THEIR DISTRIBUTIONS ARE THE         WILCOX
              C    SAME OR IF ONE TENDS TO BE GREATER THAN THE OTHER.               WILCOX
              C    ALTERNATELY, THIS TEST CAN BE USED TO TEST A SINGLE              WILCOX
              C    RANDOM SAMPLE AGAINST A SINGLE SPECIFIED VALUE.  THIS            WILCOX
              C    THEN BECOMES A TEST OF THE MEDIAN OF THE SAMPLE AGAINST          WILCOX
              C    THE VALUE SPECIFIED.                                             WILCOX
              C                                                                     WILCOX
              C    THE CRITICAL REGION IS DETERMINED USING THE LARGE SAMPLE         WILCOX
              C    APPROXIMATION, RATHER THAN BY USING AN EXACT TABLE.  THIS        WILCOX
              C    MEANS THAT FOR SMALL SAMPLES (SIZE LESS THAN 20), THE            WILCOX
              C    CRITICAL REGION GIVEN IS UNRELIABLE AND TABLES SHOULD BE         WILCOX
              C    CONSULTED USING THE TEST STATISTIC CALCULATED, WHICH IS          WILCOX
              C    VALID.                                                           WILCOX
              C                                                                     WILCOX
              C    FOR DETAILS ON THE WILCOXON SIGNED RANK TEST, SEE CONOVER, W.J., WILCOX
              C    PRACTICAL NON-PARAMETRIC STATISTICS, ROUGH DRAFT, PAGES 325-338. WILCOX
              C                                                                     WILCOX
              C                                                                     WILCOX
              C    COMMON BLOCKS                                                    WILCOX
              C                                                                     WILCOX
 ISN 0003          COMMON /CONTBL/  ROWS,COLS,RLEV(20),CLEV(20)                     DECLARE
 ISN 0004          INTEGER*4  ROWS,COLS                                             DECLARE
 ISN 0005          COMMON /DIST/  DISTN,DPARM1,DPARM2,WHTRN                         DECLARE
 ISN 0006          INTEGER*4  DISTN,WHTRN                                           DECLARE
 ISN 0007          COMMON /FLAG/  EOD,CDCOL,BUFFER(10,2),BUFNO                      DECLARE
 ISN 0008          REAL*8  BUFFER                                                   DECLARE
 ISN 0009          INTEGER*4  EOD,CDCOL,BUFNO                                       DECLARE
 ISN 0010          COMMON /FORM/   FMT(100),IPR,IPS(40),DFORM,DOUT,RCBCD            DECLARE
 ISN 0011          REAL*8   FMT                                                     DECLARE
 ISN 0012          INTEGER*4  DFORM,DOUT,RCBCD                                      DECLARE
 ISN 0013          COMMON /INPARM/  PARM(10),IPARM,INUMB,SUBPRM(40),IPCLS(40),      DECLARE
                  1    VALUE(40),IVALUE(40),RC                                      DECLARE
 ISN 0014          REAL*8   PARM,SUBPRM                                             DECLARE
 ISN 0015          INTEGER*4  RC                                                    DECLARE
```

```
ISN C016          COMMON /HYPTST/  ALPHA,TAIL,TESTS(20),DOALL                          DECLARE
ISN C017          INTEGER*4  TAIL                                                     DECLARE
ISN C018          LOGICAL*4  TESTS,DOALL                                              DECLARE
ISN C019          COMMON /PRETTY/  TITLE(10),BLOCK,TREAT,REPLI,LABEL(40),             DECLARE
                 1     DATE,TIME,PAGE,LINE                                          DECLARE
ISN C020          REAL*8   TITLE,BLOCK,TREAT,REPLI,LABEL,DATE,TIME                    DECLARE
ISN C021          INTEGER*4  PAGE                                                     DECLARE
ISN C022          COMMON /TYPES/  TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE(40)   DECLARE
ISN C023          INTEGER*4       TYPE,SMPLS,VARITE,BLK,TRT,REP,BLKT,TRTB,SSIZE       DECLARE
ISN C024          COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                 DECLARE
ISN C025          INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                 DECLARE
ISN C026          COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                            DECLARE
ISN C027          INTEGER*4  ITEMS,LOCAT,RDNO                                         DECLARE
ISN C028          COMMON /WORKA/  DATA(100),SUBS(4),WORK(5000)                        DECLARE
ISN C029          INTEGER*4  SUBS,IWRK(5000),WKSZE/5000/                              DECLARE
ISN C030          EQUIVALENCE (WORK(1),IWRK(1))                                       DECLARE
ISN C031          REAL*8  BLANK /' '/                                                 DECLARE
ISN C032          EQUIVALENCE (BLANK,IBLANK)                                          DECLARE
                C                                                                     WILCOX
                C     OTHER DATA AND CONSTANTS                                        WILCOX
                C                                                                     WILCOX
ISN C033          REAL*8  ACCREJ(2) /'ACCEPTED', 'REJECTED'/                          WILCOX
ISN C034          INTEGER*4  HYPLBL(2,3) /'.EQ.', '.NE.', '.LE.', '.GT.',             WILCOX
                 1     '.GE.', '.LT.'/                                              WILCOX
ISN C035          REAL*8  XLBL /'X'/, YLBL /'Y'/                                      WILCOX
ISN C036          INTEGER*2  H0 /'H0'/, H1 /'H1'/                                     WILCOX
ISN C037          REAL*4  RECORD(6)                                                   WILCOX
ISN C038          INTEGER*4  CURSMP, CURVAR, CURCNT, TIES, DROP, PLCS                 WILCOX
ISN C039          EQUIVALENCE (RECORD(1),CURSMP), (RECORD(2),CURVAR),                 WILCOX
                 1     (RECORD(3),CURCNT), (RECORD(5),DI), (RECORD(6),ADI)           WILCOX
ISN C040          REAL*8  T1, T2, T3                                                  WILCOX
                C                                                                     WILCOX
                C     FORMAT STATEMENTS                                               WILCOX
ISN C041        231 FORMAT ('-', T12, 'IKMS31I  UNEQUAL SAMPLE/VARIATE SIZES AFTER TRAWILCOX
                   1NSFORMATION')                                                     WILCOX
ISN C042        235 FORMAT ('-', T12, 'IKMS35I  NO MEDIAN VALUE SPECIFIED')         WILCOX
ISN C043        29C FORMAT ('-', T12, 'IKMS90I  ***** BECAUSE OF SAMPLE SIZES, ALPHA HWILCOX
                   1AT MAY NOT BE CORRECT.  VERIFY WITH TABLES. *****')             WILCOX
ISN C044        3C1 FORMAT ('-', T53, 'THE WILCOXON SIGNED RANKS TEST')             WILCOX
ISN C045        3C2 FORMAT ('0', T12, A2, ': E(', A8, ') ', A4, ' E(', A8, ')')     WILCOX
ISN C046        3C3 FORMAT ('0', T12, 'VERSUS')                                     WILCOX
ISN C047        3C4 FORMAT ('0', T12, A2, ': E(', A8, ') ', A4, G12.5)              WILCOX
ISN C048        3C5 FORMAT ('0', T12, 'THE SAMPLE SIZE IS', I6)                     WILCOX
```

```
ISN C049        306 FORMAT ('0', T12, 'THERE WERE', I5,                                  WILCOX
                   1    ' EQUAL VALUES, RESULTING IN A TEST OF', I6, ' OBSERVATIONS')  WILCOX
ISN C050        307 FORMAT ('0', T12, 'THERE WAS A TOTAL OF', I5,                        WILCOX
                   1    ' TIED DIFFERENCES IN', I3, ' PLACES')                         WILCOX
ISN C051        308 FORMAT ('0', T12, 'THE TEST STATISTIC IS ', G11.5)                   WILCOX
ISN C052        309 FORMAT ('-', T12, 'AT ALPHA =', F6.3, ', HO IS ', A8, '.')          WILCOX
ISN C053        310 FORMAT ('0', T12, 'ALPHA HAT =', F6.3)                               WILCOX
ISN C054        311 FORMAT ('-', T50, 'END OF THE WILCOXON SIGNED RANK TEST')            WILCOX
                C                                                                        WILCOX
                C     START BY FIXING THE LABELS AND PRINTING THE HYPOTHESES             WILCOX
                C                                                                        WILCOX
ISN C055              WKSZE = WKSZE                                                      WILCOX
                C                                                                        WILCOX
ISN C056              WRITE (LST,301)                                                    WILCOX
                C                                                                        WILCOX
ISN C057              ITL = 1                                                            WILCOX
ISN C058              IF (TAIL) 11, 13, 12                                               WILCOX
ISN C059           11 ITL = 2                                                            WILCOX
ISN C060              GO TO 13                                                           WILCOX
ISN C061           12 ITL = 3                                                            WILCOX
                C                                                                        WILCOX
ISN C062           13 IF (LABEL(1) .EQ. BLANK) LABEL(1) = XLBL                           WILCOX
ISN C064              IF (LABEL(2) .EQ. BLANK) LABEL(2) = YLBL                           WILCOX
ISN C066              IF (TYPE .EQ. 1) GO TO 14                                          WILCOX
                C                                                                        WILCOX
ISN 0068              WRITE (LST,302) HO, LABEL(1), HYPLBL(1,ITL), LABEL(2)              WILCOX
ISN C069              WRITE (LST,303)                                                    WILCOX
ISN C070              WRITE (LST,302) H1, LABEL(1), HYPLBL(2,ITL), LABEL(2)              WILCOX
ISN C071              GO TO 15                                                           WILCOX
                C                                                                        WILCOX
ISN C072           14 IF (DISTN .NE. -1) GO TO 1235                                      WILCOX
ISN C074              WRITE (LST,304) HO, LABEL(1), HYPLBL(1,ITL), DPARM1                WILCOX
ISN CC75              WRITE (LST,303)                                                    WILCOX
ISN C076              WRITE (LST,304) H1, LABEL(1), HYPLBL(2,ITL), DPARM1                WILCOX
                C                                                                        WILCOX
ISN C077           15 CALL OFLOW(9)                                                      WILCOX
ISN 0078              IF (TYPE .EQ. 1) GO TO 41                                          WILCOX
                C                                                                        WILCOX
                C     BIVARIATE OR 2 MATCHED SAMPLE DATA                                 WILCOX
                C                                                                        WILCOX
ISN CC80              REWIND UT1                                                         WILCOX
ISN C081              REWIND UT2                                                         WILCOX
                C                                                                        WILCOX
```

```
ISN C082          16 READ (UT1,END=21) RECCRD                                  WILCOX
ISN C083             IF (TYPE .EQ. 6) GO TC 18                                 WILCOX
                C                                                              WILCOX
                C    BIVARIATE SAMPLE                                          WILCOX
                C                                                              WILCOX
ISN 0085             IF (CURVAR .NE. 1) GO TO 16                               WILCOX
                C                                                              WILCOX
                C    WRITE OUT THE FIRST VARIATE CR SAMPLE                     WILCOX
                C                                                              WILCOX
ISN C087          17 WRITE (UT2) RECORD                                        WILCOX
ISN C088             GO TO 16                                                  WILCOX
                C                                                              WILCOX
                C    DATA IS 2 MATCHED SAMPLES                                 WILCOX
                C                                                              WILCOX
ISN 0089          18 IF (CURSMP .NE. 1) GO TO 16                               WILCOX
ISN C091             ITEMS = 1                                                 WILCOX
ISN C092             LOCAT(4) = 0                                              WILCOX
ISN C093             X(1) = RECORD(6)                                          WILCOX
ISN C094             IF (WHTRN .EQ. 1) CALL TRANS                              WILCOX
ISN CC96             IF (ITEMS .NE. 1) GO TO 14                                WILCOX
ISN C098             RECORD(6) = X(1)                                          WILCOX
ISN C099             IF (OUT .NE. 0 .AND. COUT .EQ. 2) CALL WRDATA             WILCOX
ISN 0101             GO TO 17                                                  WILCOX
                C                                                              WILCOX
                C    END OF PASS 1 OF DATA                                     WILCOX
                C                                                              WILCOX
ISN C102          21 ENDFILE UT2                                               WILCOX
ISN 0103             REWIND UT1                                                WILCOX
ISN C104             REWIND UT2                                                WILCOX
ISN 0105             REWIND UT4                                                WILCOX
ISN C106             DROP = 0                                                  WILCOX
ISN C107             NSZ = 0                                                   WILCOX
                C                                                              WILCOX
                C    SECOND PASS OF MATCHING THE VARAITES OR SAMPLES           WILCOX
                C                                                              WILCOX
ISN C108          22 READ (UT1,END=51) RECCRD                                  WILCOX
ISN C109             IF (TYPE .EQ. 6) GO TC 26                                 WILCOX
ISN 0111             IF (CURVAR .EQ. 1) GO TO 22                               WILCOX
ISN C113             X(2) = RECORD(6)                                          WILCOX
ISN C114             READ (UT2,END=99) RECCRD                                  WILCOX
ISN 0115             X(1) = RECORD(6)                                          WILCOX
ISN C116             ITEMS = 2                                                 WILCOX
ISN 0117             LOCAT(4) = 0                                              WILCOX
```

```
ISN C118            IF (WHTRN .EQ. 1) CALL TRANS                                WILCOX
ISN C120            IF (ITEMS .NE. 2) GO TO 22                                  WILCOX
ISN C122            IF (OUT .NE. 0 .OR. DOUT .EQ. 2) CALL WRDATA                WILCOX
ISN C124            GO TO 31                                                    WILCOX
                C                                                               WILCOX
                C     2 MATCHED SAMPLES                                         WILCOX
                C                                                               WILCOX
ISN C125         26 IF (CURSMP .EQ. 1) GO TO 22                                 WILCOX
ISN C127            ITEMS = 1                                                   WILCOX
ISN C128            X(1) = RECORD(6)                                            WILCOX
ISN C129            LOCAT(4) = 0                                                WILCOX
ISN C130            IF (WHTRN .EQ. 1) CALL TRANS                                WILCOX
ISN C132            IF (ITEMS .NE. 1) GO TO 22                                  WILCOX
ISN C134            IF (OUT .NE. 0 .AND. DOUT .EQ. 2) CALL WRDATA               WILCOX
                C                                                               WILCOX
ISN C136         27 X(2) = X(1)                                                 WILCOX
ISN C137            READ (UT2,END=99) RECORD                                    WILCOX
ISN C138            X(1) = RECORD(6)                                            WILCOX
                C                                                               WILCOX
                C     PROCESS THE VARIATES OR MATCHED SAMPLES                   WILCOX
                C                                                               WILCOX
ISN C139         31 NSZ = NSZ + 1                                               WILCOX
ISN C140            DI = X(1) - X(2)                                            WILCOX
ISN C141            ADI = ABS(DI)                                               WILCOX
ISN C142            IF (DI .EQ. 0.) GO TO 32                                    WILCOX
ISN C144            WRITE (UT4) RECORD                                          WILCOX
ISN C145            GO TO 22                                                    WILCOX
                C                                                               WILCOX
                C     ZERO DIFFERENCE - DROP THE OBSERVATION                    WILCOX
                C                                                               WILCOX
ISN C146         32 DROP = DROP + 1                                             WILCOX
ISN C147            GO TO 22                                                    WILCOX
                C                                                               WILCOX
                C     PROCESS 1 RANDOM SAMPLE AGAINST A "MEDIAN" VALUE          WILCOX
                C                                                               WILCOX
ISN C148         41 REWIND UT1                                                  WILCOX
ISN C149            REWIND UT4                                                  WILCOX
ISN C150            NSZ = 0                                                     WILCOX
ISN C151            DROP = 0                                                    WILCOX
                C                                                               WILCOX
ISN C152         42 READ (UT1,END=51) RECORD                                    WILCOX
ISN C153            X(1) = RECORD(6)                                            WILCOX
ISN C154            ITEMS = 1                                                   WILCOX
```

```
ISN C155            LOCAT(4) = 0                                                    WILCOX
ISN C156            IF (WHTRN .EQ. 1) CALL TRANS                                    WILCOX
ISN C158            IF (ITEMS .LT. 1) GO TO 42                                      WILCOX
ISN C160            IF (OUT .NE. 0 .AND. COUT .EQ. 2) CALL WRDATA                   WILCOX
                C                                                                   WILCOX
ISN C162         43 DO 45 I = 1, ITEMS                                              WILCOX
ISN C163            NSZ = NSZ + 1                                                   WILCOX
ISN C164            DI = X(I) - DPARM1                                              WILCOX
ISN C165            ADI = ABS(DI)                                                   WILCOX
ISN C166            IF (DI .EQ. 0.) GO TO 44                                        WILCOX
ISN C168            WRITE (UT4) RECORD                                              WILCOX
ISN 0169            GO TO 45                                                        WILCOX
ISN C170         44 DROP = DROP + 1                                                 WILCOX
ISN C171         45 CONTINUE                                                        WILCOX
ISN C172            GO TO 42                                                        WILCOX
                C                                                                   WILCOX.
                C     END OF BOTH PASSES - THE DI'S ARE ON UT4                      WILCOX
                C                                                                   WILCOX
ISN C173         51 ENDFILE UT4                                                     WILCOX
ISN C174            IF (OUT .NE. 0 .AND. COUT .EQ. 2 .AND. OUT .NE. LST) ENDFILE OUT WILCOX
ISN C176            OUT = 0                                                         WILCOX
                C                                                                   WILCOX
                C     SORT THE DATA DOWN ON THE ABSOLUTE DIFFERENCES                WILCOX
                C                                                                   WILCOX
ISN C177            CALL RANKER(UT2,UT4,UT2,UT3,J)                                  WILCOX
ISN C178            IF (RC .NE. 0) GO TO 92                                         WILCOX
ISN C180            SUBS(1) = 0                                                     WILCOX
ISN 0181            SUBS(2) = 0                                                     WILCOX
ISN 0182            WRITE (LST,305) NSZ                                             WILCOX
                C                                                                   WILCOX
                C     CHECK THE NUMBER OF DROPPED OBSERVATIONS                      WILCOX
                C                                                                   WILCOX
ISN 0183            IF (DROP .EQ. 0) GO TO 52                                       WILCOX
ISN C185            NSZ = NSZ - DROP                                                WILCOX
ISN C186            WRITE (LST,306) DROP,NSZ                                        WILCOX
                C                                                                   WILCOX
                C     PROCESS SORTED OBSERVATIONS AND RANK THEM                     WILCOX
                C                                                                   WILCOX
ISN 0187         52 REWIND UT2                                                      WILCOX
ISN 0188            T = 0.                                                          WILCOX
ISN 0189            RANK = 0.                                                       WILCOX
ISN 0190            TIES = 0                                                        WILCOX
ISN 0191            PLCS = 0                                                        WILCOX
```

```
ISN 0192            READ (UT2,END=99) RECORD                                WILCOX
                C                                                           WILCOX
                C     SAVE THE OBSERVATION VALUES                           WILCOX
                C                                                           WILCOX
ISN 0193         61 D = RECORD(5)                                           WILCOX
ISN 0194            ABSD = RECORD(6)                                        WILCOX
ISN 0195            RANK = RANK + 1.                                        WILCOX
                C                                                           WILCOX
                C     READ THE NEXT OBSERVATION                             WILCOX
                C                                                           WILCOX
ISN 0196            READ (UT2,END=62) RECORD                                WILCOX
ISN 0197            IF (ABSD .EQ. RECORD(6)) GO TO 63                       WILCOX
                C                                                           WILCOX
                C     PROCESS A NON-TIED OBSERVATION                        WILCOX
                C                                                           WILCOX
ISN 0199            IF (D .GT. 0.) T = T + RANK                             WILCOX
ISN 0201            GO TO 61                                                WILCOX
                C                                                           WILCOX
                C     END OF FILE                                           WILCOX
                C                                                           WILCOX
ISN 0202         62 IF (D .GT. 0) T = T + RANK                              WILCOX
ISN 0204            GO TO 71                                                WILCOX
                C                                                           WILCOX
                C     AT LEAST 2 OBSERVATIONS ARE TIED                      WILCOX
                C                                                           WILCOX
ISN 0205         63 PLCS = PLCS + 1                                         WILCOX
ISN 0206            I = 2                                                   WILCOX
ISN 0207            K = 0                                                   WILCOX
ISN 0208            IF (D .GT. 0.) K = 1                                    WILCOX
ISN 0210         64 IF (RECORD(5) .GT. 0.) K = K + 1                        WILCOX
                C                                                           WILCOX
                C     READ TO SEE IF THE NEXT ONE IS TIED                   WILCOX
                C                                                           WILCOX
ISN 0212            READ (UT2,END=67) RECORD                                WILCOX
ISN 0213            IF (RECORD(6) .NE. ABSD) GO TO 65                       WILCOX
ISN 0215            I = I + 1                                               WILCOX
ISN 0216            GO TO 64                                                WILCOX
                C                                                           WILCOX
                C     CALCULATE AVERAGE OF RANKS                            WILCOX
                C                                                           WILCOX
ISN 0217         65 Z = 0.                                                  WILCOX
ISN 0218            DO 66 L = 1, I                                          WILCOX
ISN 0219            Z = Z + RANK                                            WILCOX
```

```
ISN 0220          66 RANK = RANK + 1.                                    WILCOX
                C                                                        WILCOX
ISN 0221             TIES = TIES + I                                     WILCOX
ISN 0222             Z = Z / I                                           WILCOX
ISN 0223             T = T + K * Z                                       WILCOX
ISN 0224             RANK = RANK - 1.                                    WILCOX
ISN 0225             GO TO 61                                            WILCOX
                C                                                        WILCOX
                C    END OF FILE WHILE PROCESSING TIES                   WILCOX
                C                                                        WILCOX
ISN 0226          67 Z = 0.                                              WILCOX
ISN 0227             DO 68 L = 1, I                                      WILCOX
ISN 0228             Z = Z + RANK                                        WILCOX
ISN 0229          68 RANK = RANK + 1.                                    WILCOX
                C                                                        WILCOX
ISN 0230             TIES = TIES + I                                     WILCOX
ISN 0231             Z = Z / I                                           WILCOX
ISN 0232             T = T + K * Z                                       WILCOX
ISN 0233             GO TO 71                                            WILCOX
                C                                                        WILCOX
                C    END OF FILE NOT WHILE PROCESSING TIES               WILCOX
                C                                                        WILCOX
ISN 0234          69 IF (D .GT. 0) T = T + RANK                          WILCOX
                C                                                        WILCOX
                C    PROCESSING OF THE RANKS IS THROUGH                  WILCOX
                C                                                        WILCOX
ISN 0236          71 IF (TIES .EQ. 0) GO TC 72                           WILCOX
ISN 0238             WRITE (LST,307) TIES, PLCS                          WILCOX
ISN 0239             CALL OFLOW(2)                                       WILCOX
                C                                                        WILCOX
ISN 0240          72 WRITE (LST,308) T                                   WILCOX
ISN 0241             CALL OFLOW(2)                                       WILCOX
                C                                                        WILCOX
                C    DETERMINE CRITICAL REGION                           WILCOX
                C                                                        WILCOX
ISN 0242          73 T1 = NSZ * (NSZ - 1)                                WILCOX
ISN 0243             T1 = T1 / 4.                                        WILCOX
ISN 0244             T2 = T1 * (2 * NSZ + 1) / 6.                        WILCOX
ISN 0245             T2 = DSQRT(T2)                                      WILCOX
ISN 0246             T3 = T                                              WILCOX
                C                                                        WILCOX
ISN 0247             XP = (T3 - T1) / T2                                 WILCOX
ISN 0248             ALFHAT = ANDTR(XP)                                  WILCOX
```

```
                C                                                                WILCOX
ISN 0249              GO TO (76, 81, 86), ITL                                    WILCOX
                C                                                                WILCOX
                C     2-TAILED TEST                                              WILCOX
                C                                                                WILCOX
ISN 0250           76 ALFHAT = 2. * AMIN1(ALFHAT, 1.-ALFHAT)                     WILCOX
ISN 0251              IF (ALFHAT .GT. 1.) ALFHAT = 1.                            WILCOX
ISN 0253              GO TO 91                                                   WILCOX
                C                                                                WILCOX
                C     LOWER TAIL TEST                                            WILCOX
                C                                                                WILCOX
ISN 0254           81 GO TO 91                                                   WILCOX
                C                                                                WILCOX
                C     UPPER TAIL TEST                                            WILCOX
                C                                                                WILCOX
ISN 0255           86 ALFHAT = 1. - ALFHAT                                       WILCOX
                C                                                                WILCOX
                C     DETERMINE RESULT OF TEST                                   WILCOX
                C                                                                WILCOX
ISN 0256           91 I = 1                                                      WILCOX
ISN 0257              IF (ALPHA .GT. ALFHAT) I = 2                               WILCOX
                C                                                                WILCOX
ISN 0259              WRITE (LST,309) ALPHA, ACCREJ(I)                           WILCOX
ISN 0260              WRITE (LST,310) ALFHAT                                     WILCOX
ISN 0261              CALL OFLOW(5)                                              WILCOX
                C                                                                WILCOX
                C     CHECK FOR TOO SMALL OF A SAMPLE SIZE                       WILCOX
                C                                                                WILCOX
ISN 0262              IF (NSZ .GT. 20) GO TC 92                                  WILCOX
ISN 0264              WRITE (LST,290)                                            WILCOX
ISN 0265              CALL OFLOW(3)                                              WILCOX
                C                                                                WILCOX
                C     END OF THE WILCOXON SIGNED RANK TEST                       WILCOX
                C                                                                WILCOX
ISN 0266           92 WRITE (LST,311)                                            WILCOX
ISN 0267              CALL OFLOW(3)                                              WILCOX
ISN 0268              RETURN                                                     WILCOX
                C                                                                WILCOX
                C     ERRORS THAT CAN OCCUR                                      WILCOX
                C                                                                WILCOX
ISN 0269           99 WRITE (LST,231)                                            WILCOX
ISN 0270              GO TO 98                                                   WILCOX
                C                                                                WILCOX
```

```
   ISN C271      1235 WRITE (LST,235)                                          WILCOX
                C                                                             WILCOX
   ISN C272        98 CALL OFLOW(3)                                            WILCOX
   ISN C273           RC = 12                                                  WILCOX
   ISN C274           GO TO 92                                                 WILCOX
                C                                                             WILCOX
   ISN C275           END                                                      WILCOX

*OPTIONS IN EFFECT*       NAME=  MAIN,OPT=01,LINECNT=45

*OPTIONS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*      SOURCE STATEMENTS =      274 ,PROGRAM SIZE =      4198

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
  ISN C002            SUBROUTINE WRDATA                                                    WRDATA
                C                                                                          WRDATA
                C                                                                          WRDATA
                C     THIS SUBROUTINE WRITES THE DATA OUT ON INTERMEDIATE STORAGE          WRDATA
                C     IF THE USER HAS SO REQUESTED.  THE DATA IS WRITTEN IN SINGLE         WRDATA
                C     PRECISION INTERNAL FORM (UNFORMATTED) WITH THE LOCATION              WRDATA
                C     PARAMETERS (IF ANY) FIRST, IMMEDIATELY FOLLOWED BY THE DATA          WRDATA
                C     ASSOCIATED WITH THAT LOCATION.  THE NUMBER OF ITEMS ACTUALLY         WRDATA
                C     WRITTEN IS DETERMINED BY THE NUMBER OF ITEMS FIELD, WHICH            WRDATA
                C     IS SET BY THE INPUT RCUTINE (RDDATA) OR BY THE USER IN               WRDATA
                C     THE TRANSFORMATION SUBROUTINE.                                       WRDATA
                C     THE DATA IS WRITTEN ON THE UNIT NUMBER IN THE FIELD 'OUT'            WRDATA
                C     IN THE COMMON BLOCK /UNITS/.                                         WRDATA
                C     HOWEVER, IF THE UNIT 'OUT' IS THE SAME AS THE UNIT 'LST',            WRDATA
                C     THE DATA IS PRINTED IN FORMAT 7G16.8, RATHER THAN BE WRITTEN         WRDATA
                C     OUT IN INTERNAL FORMAT.  THIS IS HOW THE USER LISTS HIS DATA.        WRDATA
                C                                                                          WRDATA
                C                                                                          WRDATA
                C     COMMON BLOCKS USED                                                   WRDATA
                C                                                                          WRDATA
  ISN C003            COMMON /UNITS/  RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                WRDATA
  ISN C004            INTEGER*4       RDR,PCH,LST,UT1,UT2,UT3,UT4,IPT,OUT                WRDATA
  ISN C005            COMMON /USER/  ITEMS,X(40),LOCAT(4),RDNO                           WRDATA
  ISN C006            INTEGER*4  ITEMS,LOCAT,RDNO                                        WRDATA
                C                                                                          WRDATA
                C     FORMAT STATEMENTS FOR LISTING THE DATA ON THE PRINTER                WRDATA
                C                                                                          WRDATA
  ISN C007        401 FORMAT (' ', T12, 7G16.8)                                          WRDATA
                C                                                                          WRDATA
                C     CHECK IF DATA IS TO BE LISTED                                        WRDATA
                C                                                                          WRDATA
  ISN C008            IF (OUT .EQ. LST) GO TO 11                                         WRDATA
                C                                                                          WRDATA
                C     CHECK THE NUMBER OF LOCATION SUBSCRIPTS TO BE WRITTEN                WRDATA
                C                                                                          WRDATA
  ISN C010            LIM = LOCAT(4)                                                     WRDATA
  ISN C011            IF (LIM .EQ. 0) GO TO 1                                            WRDATA
                C                                                                          WRDATA
                C     WRITE OUT DATA WITH LOCATION INFORMATION                             WRDATA
                C                                                                          WRDATA
  ISN C013            WRITE (OUT) (LOCAT(I), I = 1, LIM), (X(I), I = 1, ITEMS)           WRDATA
```

```
ISN C014              GO TO 2                                                    WRDATA
                C                                                                WRDATA
                C     WRITE OUT THE DATA ONLY AS NO LOCATION INFO WAS SPECIFIED   WRDATA
                C                                                                WRDATA
ISN C015             1 WRITE (OUT) (X(I), I = 1, ITEMS)                          WRDATA
                C                                                                WRDATA
                C     RETURN TO CALLING PROGRAM                                  WRDATA
                C                                                                WRDATA
ISN C016             2 RETURN                                                    WRDATA
                C                                                                WRDATA
                C     DATA IS TO BE LISTED                                       WRDATA
                C                                                                WRDATA
ISN C017            11 LIM = LOCAT(4)                                            WRDATA
ISN C018               IF (LIM .NE. 0) GO TO 12                                  WRDATA
ISN C020               WRITE (OUT,401) (X(I), I = 1, ITEMS)                      WRDATA
ISN C021               LIM = (ITEMS + 6) / 7                                     WRDATA.
ISN C022               CALL OFLOW(LIM)                                           WRDATA
ISN C023               GO TO 2                                                   WRDATA
                C                                                                WRDATA
ISN 0024            12 WRITE (OUT,401) (LOCAT(I), I = 1, LIM), (X(I), I = 1, ITEMS)  WRDATA
ISN CC25               LIM = (ITEMS + LIM + 6) / 7                               WRDATA
ISN C026               CALL OFLOW(LIM)                                           WRDATA
ISN C027               GO TO 2                                                   WRDATA
                C                                                                WRDATA
ISN C028               END                                                       WRDATA

*OPTICNS IN EFFECT*      NAME=  MAIN,OPT=01,LINECNT=45

*OPTICNS IN EFFECT*     SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      27 ,PRCGRAM SIZE =      676

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******

*STATISTICS*  NO  DIAGNOSTICS THIS STEP
```

APPENDIX II

ASSEMBLY LANGUAGE PROGRAM LISTINGS

```
SYMBOL   TYPE ID  ADDR   LENGTH LD ID
CNTFMT    SD  01 000000 000000
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

                                  2 ************************************************************************ CNTFMT
                                  3 *                                                                      * CNTFMT
                                  4 *              THIS SUBROUTINE IS USED TO COUNT THE BALANCE OF         * CNTFMT
                                  5 *        PARENTHESES IN A USER SUPPLIED FORMAT STATEMENT SO THAT       * CNTFMT
                                  6 *        THE CALLING PROGRAM WILL KNOW WHETHER OR NOT THE FORMAT IS    * CNTFMT
                                  7 *        COMPLETE.                                                     * CNTFMT
                                  8 *              THIS SUBROUTINE IS ENTERED USING A STANDARD 'CALL'      * CNTFMT
                                  9 *        STATEMENT WITH 2 PARAMETERS.                                  * CNTFMT
                                 10 *              THE FIRST IS THE ADDRESS OF THE FIRST BYTE OF THE       * CNTFMT
                                 11 *              FORMAT STATEMENT TO BE EXAMINED.  THE FORMAT STATEMENT  * CNTFMT
                                 12 *              AREA MUST BE 800 BYTES LONG.  THIS CAN BE DEFINED IN    * CNTFMT
                                 13 *              FORTRAN BY REAL*8 FMT(100), AND WILL HOLD A FORMAT      * CNTFMT
                                 14 *              SPECIFICATION  UP TO 10 CARDS LONG.                     * CNTFMT
                                 15 *              THE SECOND PARAMETER IS THE ADDRESS OF A FULL WORD      * CNTFMT
                                 16 *              INTEGER (INTEGER*4) WHICH WILL CONTAIN THE 'LEFT-RIGHT'* CNTFMT
                                 17 *              PARENTHESES COUNT.  IF THIS GOES NEGATIVE, THE FORMAT   * CNTFMT
                                 18 *              STATEMENT IS OBVIOUSLY IN ERROR AND THE SUBROUTINE      * CNTFMT
                                 19 *              TERMINATES, RETURNING THIS VALUE.                       * CNTFMT
                                 20 *                                                                      * CNTFMT
                                 21 *              THE FORMAT STATEMENT IS SCANNED UNTIL ANY 1 OF THE      * CNTFMT
                                 22 *        FOLLOWING CONDITIONS ARE DETECTED:                            * CNTFMT
                                 23 *              1)   THE COUNT GOES NEGATIVE.                           * CNTFMT
                                 24 *              2)   THE END OF THE 800 BYTES IS REACHED.               * CNTFMT
                                 25 *                                                                      * CNTFMT
                                 26 *              THE RETURNED COUNT CAN BE USED TO DETERMINED THE        * CNTFMT
                                 27 *        STATUS OF THE FORMAT STATEMENT, AS FOLLOWS:                   * CNTFMT
                                 28 *              1)   NEGATIVE - FORMAT IS IN ERROR.                     * CNTFMT
                                 29 *              2)   POSITIVE - FORMAT IS NOT COMPLETE.                 * CNTFMT
                                 30 *              3)   ZERO - PARENTHESES BALANCE, FORMAT COMPLETE.       * CNTFMT
                                 31 *                                                                      * CNTFMT
                                 32 *              THIS PROGRAM DOES NOT CHECK THE SYNTAX OF THE FORMAT    * CNTFMT
                                 33 *        SPECIFICATIONS, BEYOND DESIRING BALANCED PARENTHESES.         * CNTFMT
                                 34 *                                                                      * CNTFMT
                                 35 *              THIS ROUTINE IS SERIALLY REUSABLE, BUT CAN BE MADE      * CNTFMT
                                 36 *        RE-ENTRANT BY REMOVAL OF THE 'SA=SAVEAREA' PARAMETER ON       * CNTFMT
                                 37 *        THE PRIME MACRO CALL IN THE FRONT OF THE PROGRAM.             * CNTFMT
                                 38 *                                                                      * CNTFMT
                                 39 ************************************************************************ CNTFMT
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

                                   41 CNTFMT   PRIME BR=R12,SA=SAVEAREA       ENTER THE SUBROUTINE LINKAGE    CNTFMT
000000                             42+CNTFMT   START 0
000000 47F0 F00C          0000C    43+         B     12(C,15) BRANCH AROUND ID
000004 C6                          44+         DC    AL1(6)
000005 C3D5E3C6D4E3                45+         DC    CL6'CNTFMT' IDENTIFIER
00000B C0
00000C 90EC D00C          0000C    46+         STM   14,12,12(13) SAVE REGISTERS
00001C 05C0                        47+         BALR  R12,0 CREATE BASE REGISTER
000012                             48+         USING *,R12
000012 1830                        49+         LR    3,0 SAVE -0-
000014 1821                        50+         LR    2,1 SAVE PARAMETER REGISTER
000016 4110 C066          00078    51+         LA    1,SAVEAREA GET ADDRESS OF SAVE AREA
00001A 501D 0008          00008    52+         ST    1,8(13) STORE FORWARD POINTER
00001E 50D1 0004          00004    53+         ST    13,4(1) STORE BACKWARD POINTER
000022 18D1                        54+         LR    13,1 SET UP REG FOR SAVE AREA
000024 1812                        55+         LR    1,2 RESTORE PARAMETER REGISTER
000026 1803                        56+         LR    0,3 RESTORE -0-
                                   57 *                                                                      CNTFMT
000028 9823 1000          00000    58          LM    R02,R03,0(R01)           LOAD THE PARAMETER ADDRESSES    CNTFMT
00002C 4122 0000          00000    59          LA    R02,0(R02)               ZERO THE HI ORDER BYTE OF THE   CNTFMT
000030 4140 0001          00001    60          LA    R04,1                    FORMAT ADDRESS                  CNTFMT
000034 4152 031F          0031F    61          LA    R05,FMTLENG-1(R02)       GET ADDRESS OF LAST BYTE OF FMT CNTFMT
000038 1766                        62          XR    R06,R06                  ZERO COUNT REGISTER             CNTFMT
                                   63 *                                                                      CNTFMT
00003A 954D 2000     00000         64 CKBLNK   CLI   0(R02),C'('              LOOK FOR A LEFTY                CNTFMT
00003E 4780 C052          00064    65          BE    CNTLEFT                  GO COUNT IT IF IT IS            CNTFMT
000042 955D 2000     00000         66          CLI   0(R02),C')'              NOW CHECK FOR A RIGHTY          CNTFMT
000046 4780 C058          0006A    67          BE    CNTRIGHT                 GO COUNT IT IF IT IS            CNTFMT
                                   68 *                                                                      CNTFMT
00004A 8724 C028          0003A    69 BXLE     BXLE  R02,R04,CKBLNK           LOOP THRU THE FORMAT STATEMENT  CNTFMT
                                   70 *                                                                      CNTFMT
00004E 5063 0000          00000    71 FINISH   ST    R06,0(R03)               STORE THE '('-')' COUNT         CNTFMT
                                   72          TERME RC=0                     RETURN TO CALLER                CNTFMT
000052 58DD 0004          00004    73+         L     13,4(13) RESTORE REGISTER -13-
000056 98EC D00C          0000C    74+         LM    14,12,12(13) RESTORE THE REGISTERS
00005A 92FF D00C     0000C         75+         MVI   12(13),X'FF' SET RETURN INDICATION
00005E 41F0 0000          00000    76+         LA    15,0(0,0) LOAD RETURN CODE
000062 07FE                        77+         BR    14 RETURN
                                   78 *                                                                      CNTFMT
000064 1A64                        79 CNTLEFT  AR    R06,R04                  COUNT UP FOR LEFT PARENTHESIS   CNTFMT
```

```
CFMT   SUBROUTINE -CNTFMT- COUNT FORMAT PARENTHESES                                                    PAGE    3

  LOC  CBJECT CODE     ADDR1 ADDR2  STMT   SCURCE STATEMENT                                                14 SEP 70

C00C66 47F0 C038            0004A    80          B     BXLE                GO TO CONTINUE LOOPING          CNTFMT
                                     81 *                                                                  CNTFMT
00006A 1B64                          82 CNTRIGHT SR    R06,R04             COUNT DOWN FOR A RIGHT ONE      CNTFMT
00006C 4740 C03C            0004E    83          BL    FINISH              QUIT IF IT GOES NEGATIVE        CNTFMT
000C7C 47F0 C038            0004A    84          B     BXLE                GO TO CONTINUE LOOP SCAN        CNTFMT
                                     85 *                                                                  CNTFMT
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000C74 C0000000
000C78 C000000000000000                87 SAVEAREA DC    9D'0'      SAVE AREA FOR OS                  CNTFMT
                                       88 *                                                           CNTFMT
000320                                 89 FMTLENG  EQU   800        BYTE LENGTH OF THE FORMAT STMT    CNTFMT
                                       90 *                                                           CNTFMT
                                       91          REGS  STD        REGISTER NAME EQUATES            CNTFMT
000000                                 92+R00      EQU   0
000001                                 93+R01      EQU   1
000002                                 94+R02      EQU   2
000003                                 95+R03      EQU   3
000004                                 96+R04      EQU   4
000005                                 97+R05      EQU   5
000006                                 98+R06      EQU   6
000007                                 99+R07      EQU   7
000008                                100+R08      EQU   8
000009                                101+R09      EQU   9
00000A                                102+R10      EQU   10
00000B                                103+R11      EQU   11
00000C                                104+R12      EQU   12
00000D                                105+R13      EQU   13
00000E                                106+R14      EQU   14
00000F                                107+R15      EQU   15
                                      108 *                                                          CNTFMT
                                      109          END                                               CNTFMT
```

SYMBOL	LEN	VALUE	DEFN	REFERENCES						
BXLE	4	000048	69	80	84					
CKBLNK	4	000038	64	69						
CNTFMT	1	000000	42							
CNTLEFT	2	000064	79	65						
CNTRIGHT	2	000068	82	67						
FINISH	4	000046	71	83						
FMTLENG	1	000320	89	61						
R00	1	000000	92							
R01	1	000001	93	58						
R02	1	000002	94	58	59	59	61	64	66	69
R03	1	000003	95	58	71					
R04	1	000004	96	60	69	79	82			
R05	1	000005	97	61						
R06	1	000006	98	62	62	71	79	82		
R07	1	000007	99							
R08	1	000008	100							
R09	1	000009	101							
R10	1	00000A	102							
R11	1	00000B	103							
R12	1	00000C	104	47	48					
R13	1	00000D	105							
R14	1	00000E	106							
R15	1	00000F	107							
SAVEAREA	8	000078	87	51						

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

```
SYMBOL    TYPE ID  ADDR   LENGTH LD ID

DECODE     SD  01 000000 000774
PRETTY     ER  02
INPARM     ER  03
IHBFFFIO   ER  04
```

```
DCDE   SUBROUTINE -DECODE-  ANALYZE PARAMETER CARDS                                                    PAGE    1

 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  14 SEP 70

                                   2 ************************************************************************ DECODE
                                   3 *                                                                      * DECODE
                                   4 *              THIS SUBROUTINE IS ENTERED USING A STANDARD 'CALL'      * DECODE
                                   5 *        STATEMENT WITH NO PARAMETERS PASSED.  ALL OF THE NECESSARY    * DECODE
                                   6 *        PARAMETERS ARE ASSUMED TO BE IN THE COMMON BLOCK NAMED        * DECODE
                                   7 *        /INPARM/.  THE RETURNED DATA IS PLACED IN THAT COMMON BLOCK   * DECODE
                                   8 *        AND ALSO THE BLOCK NAMED /PRETTY/.  BOTH OF THESE BLOCKS ARE  * DECODE
                                   9 *        DESCRIBED IN DUMMY SECTIONS AT THE END OF THIS PROGRAM.       * DECODE
                                  10 *                                                                      * DECODE
                                  11 *              THIS SUBROUTINE USES, AS AN EXTERNAL SUBROUTINE, THE    * DECODE
                                  12 *        FLOATING POINT CONVERSION ROUTINE, IHBFFFIO, WRITTEN BY       * DECODE
                                  13 *        KEITH MOE, APRIL, 1969.  IT USES THE MACROS WRITTEN AS PART   * DECODE
                                  14 *        OF THAT ROUTINE FOR LINKAGE TO IT.                            * DECODE
                                  15 *                                                                      * DECODE
                                  16 *              THIS ROUTINE ALSO USES SEVERAL MACROS WRITTEN BY        * DECODE
                                  17 *        KEITH MOE TO MAKE THE CODING EASIER AND LESS CLUTTERED.       * DECODE
                                  18 *        THESE MACROS ARE NEEDED FOR PROPER ASSEMBLY OF THIS ROUTINE.  * DECODE
                                  19 *                                                                      * DECODE
                                  20 *              THIS ROUTINE IS SERIALLY REUSABLE, BUT IS NOT           * DECODE
                                  21 *        RE-ENTRANT.  IT WOULD BE DIFFICULT TO MAKE IT SO.             * DECODE
                                  22 *                                                                      * DECODE
                                  23 *              AS AN EXTRA ADDED ATTRACTION, THIS ROUTINE WILL         * DECODE
                                  24 *        PROVIDE TODAY'S DATE AND TIME UPON ENTRY TO THIS ROUTINE      * DECODE
                                  25 *        WHENEVER THE 'DATE' FIELD (SEE THE COMMON BLOCK /PRETTY/)     * DECODE
                                  26 *        IS BLANK.                                                     * DECODE
                                  27 *                                                                      * DECODE
                                  28 ************************************************************************ DECODE
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

                                     30 DECODE   PRIME BR=R12,SA=SAVEAREA       ROUTINE ENTRY LINKAGE          DECODE
000000                               31+DECODE   START 0
000000 47F0 F00C            00000    32+         B     12(0,15) BRANCH AROUND ID
000004 06                            33+         DC    AL1(6)
000005 C4C5C3D6C4C5                  34+         DC    CL6'DECODE' IDENTIFIER
00000B 00
00000C 90EC D00C            0000C    35+         STM   14,12,12(13) SAVE REGISTERS
00001C 05C0                          36+         BALR  R12,0 CREATE BASE REGISTER
000012                               37+         USING *,R12
000012 1830                          38+         LR    3,0 SAVE -0-
000014 1821                          39+         LR    2,1 SAVE PARAMETER REGISTER
000016 411C C4D6            004E8    40+         LA    1,SAVEAREA GET ADDRESS OF SAVE AREA
00001A 501D 0008            00008    41+         ST    1,8(13) STORE FORWARD POINTER
00001E 50D1 0004            00004    42+         ST    13,4(1) STORE BACKWARD POINTER
000022 18D1                          43+         LR    13,1 SET UP REG FOR SAVE AREA
000024 1812                          44+         LR    1,2 RESTORE PARAMETER REGISTER
000026 1803                          45+         LR    0,3 RESTORE -0-
0004E8                               46          USING SAVEAREA,R13             USE SAVEAREA FOR EXTRA BASE    DECODE
                                     47 *                                                                       DECODE
000028 98AB D180            00668    48          LM    R10,R11,=V(PRETTY,INPARM)     LOAD THE COMMON ADDRESSES DECODE
000000                               49          USING PRTY,R10                 ESTABLISH ADDRESSIBILITY FOR   DECODE
000000                               50          USING PARMS,R11                THE COMMON BLOCKS USED         DECODE
                                     51 *                                                                       DECODE
00002C D507 A1A8 DCAC 001A8 00588    52          CLC   DATE,BLANKS              CHECK FOR A BLANK DATE FIELD   DECODE
000032 4770 00E0            000F2    53          BNE   GOTDATE                  SKIP DATE PICK UP IF THERE     DECODE
                                     54 *                                                                       DECODE
000036 1700                          55          XR    R00,R00                  CLEAR OUT TIME IN CASE MACHINE DECODE
                                     56          TIME  DEC                      DOESN'T HAVE TIMER             DECODE
000038 411C 0002            00002    57+         LA    1,2(0,0) LOAD 1 TO SPECIFY UNIT
00003C 0A0B                          58+         SVC   11 ISSUE TIME SVC
                                     59 *                                                                       DECODE
00003E 1200                          60          LTR   R00,R00                  CHECK FOR MISSING TIME         DECODE
000040 4780 0056            00068    61          BZ    NOTIME                   BRANCH IF NO TIME SUPPLIED     DECODE
                                     62 *                                                                       DECODE
000044 5000 D048            00530    63          ST    R00,DOUBLE               DOUBLE = HHMMSSTH........      DECODE
000048 F363 D050 D048 00538 00530    64          UNPK  WORK(7),DOUBLE(4)        WORK = FHFHFMFMFSFSHT          DECODE
                                     65 *                                                                       DECODE
00004E D201 A1B0 D050 001B0 00538    66          MVC   TIME(2),WORK             HH                             DECODE
000054 927A A1B2      001B2          67          MVI   TIME+2,C':'              :                              DECODE
000058 D201 A1B3 D052 001B3 0053A    68          MVC   TIME+3(2),WORK+2         MM                             DECODE
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

000C5E 927A A1B5       C01B5          69          MVI   TIME+5,C':'                :                                DECODE
CCCC62 D201 A1B6 DC54 001B6 0053C     70          MVC   TIME+6(2),WORK+4           SS                               DECODE
                                      71 *              YIELDS TIME = HH:MM:SS IN CHARACTER                        DECODE
                                      72 *                                                                         DECODE
CCCC68 5010 D04C             00534    73 NOTIME   ST    R01,DOUBLE+4               STORE JULIAN DECIMAL DATE        DECODE
CCCC6C 881C CCCC             0000C    74          SRL   R01,12                     SHIFT OUT DAYS                   DECODE
000C7C 501C DC48             C053C    75          ST    R01,CCUBLE                 STORE YEAR WITH BAD SIGN         DECODE
000C74 960F DC4B       C0533          76          OI    DOUBLE+3,X'0F'             FIX THAT BAD SIGN                DECODE
000C78 F311 A1AE DC4A 001AE 00532     77          UNPK  DATE+6(2),DOUBLE+2(2)      YY                               DECODE
000C7E 9261 A1AD       C01AD          78          MVI   DATE+5,C'/'                /                                DECODE
000C82 9261 A1AA       C01AA          79          MVI   DATE+2,C'/'                /                                DECODE
                                      80 *                                                                         DECODE
000C86 921C D109       C05F1          81          MVI   FEBDAY,28                  SET FEBRUARY TO 28 DAYS          DECODE
000C8A D707 DC5C DC5C C0538 C0538     82          XC    WORK(8),WORK               ZERO A DOUBLE WORD IN WORK       DECODE
000C9C D201 DC56 D04A C053E 00532     83          MVC   WORK+6(2),DOUBLE+2         MOVE IN YEAR TO MAKE PL8         DECODE
CCCC96 4F2C DC5C             C0538    84          CVB   R02,WORK                   CONVERT IT TO BINARY             DECODE
000C9A 5020 DC48             C0530    85          ST    R02,DOUBLE                 STORE IT FOR LEAP YEAR TEST      DECODE
CCCC9E 9103 DC4B       C0533          86          TM    DOUBLE+3,B'0CCC0011'       TEST FOR MULTIPLE OF 4           DECODE
CCCCA2 477C C098             000AA    87          BNZ   *+8                        BRANCH IF NOT                    DECODE
CCCCA6 921D D109       C05F1          88          MVI   FEBDAY,29                  SET IT TO 29 DAYS                DECODE
                                      89 *                                                                         DECODE
CCCCAA D705 DC48 DC48 C0530 00530     90          XC    DOUBLE(6),DOUBLE           CLEAR ALL BUT DAYS               DECODE
CCCCBC 4F1C DC48             0053C    91          CVB   R01,DOUBLE                 CONVERT DAY NUMBER TO BINARY     DECODE
                                      92 *                                                                         DECODE
CCCCB4 412C C001             00001    93          LA    R02,1                      SET MONTH NUMBER TO 1            DECODE
CCCCB8 413C D1C8             005F0    94          LA    R03,MONTBL                 SET POINTER TO DAYS PER MONTH    DECODE
000CBC 1744                           95          XR    R04,R04                    CLEAR A REGISTER                 DECODE
CCCCBE 4343 CC00             00000    96 MONLP    IC    R04,0(R03)                 INSERT NUMBER OF DAYS THIS MON   DECODE
0CCCC2 1B14                           97          SR    R01,R04                    SUBTRACT FROM DAYS THIS YEAR     DECODE
0CCCC4 47CC CCC2             C00D4    98          BNH   GOTMON                     BRANCH WHEN DAYS ZERO OR NEG     DECODE
0CCCC8 4122 C001             00001    99          LA    R02,1(R02)                 ADD 1 TO THE MONTH NUMBER        DECODE
CCCCCC 4133 CCC1             00001   100          LA    R03,1(R03)                 ADD 1 TO THE POINTER ADDRESS     DECODE
000CDC 47FC CCAC             C00BE   101          B     MONLP                      GO TRY NEXT MONTH                DECODE
                                     102 *                                                                         DECODE
000CD4 1A14                          103 GOTMON   AR    R01,R04                    THIS WILL LEAVE THE DAY OF       DECODE
000CD6 4E1C DC48             00530   104          CVD   R01,DOUBLE                 THE MONTH IN REGISTER 1          DECODE
000CDA 960F DC4F       C0537         105          OI    DOUBLE+7,X'0F'             FIX SIGN FOR DAY                 DECODE
CCCCDE F311 A1AB DC4E C01AB 00536    106          UNPK  DATE+3(2),DOUBLE+6(2)      DD                               DECODE
000CE4 4E2C DC48             0053C   107          CVD   R02,DOUBLE                 NOW GET THE MONTH NUMBER         DECODE
000CE8 960F DC4F       C0537         108          OI    DOUBLE+7,X'0F'             FIX THE SIGN                     DECODE
```

```
000CEC F311 A1A8 DC4E C01A8 C0536   109          UNPK  DATE(2),DOUBLE+6(2)      MM      DECODE
                                    110 *              YIELDS A DATE FIELD OF MM/DD/YY  DECODE
                                    111 *                                               DECODE
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

0000F2 D207 D0F0 D0A0 005D8 00588   113 GOTDATE  MVC   CDTYP,BLANKS              MAIN PROCESSING - BLANK TYPE     DECODE
0000F8 D703 D0F8 D0F8 005E0 005F0   114           XC    TYPNUMB,TYPNUMB           ZERO TYPE NUMBER                 DECODE
0000FE 4110 C008            00008   115           LA    R01,8                     SET LIMIT OF TYPE LENGTH         DECODE
000102 4120 B000            00000   116           LA    R02,PARM                  LOAD ADDRESS OF PARM CARD        DECODE
000106 4130 D0F0            005D8   117           LA    R03,CDTYP                 LOAD ADDRESS OF TYPE WORK AREA   DECODE
                                    118 *                                                                        DECODE
00010A D200 3000 2000 00000 00000   119 PMOVE    MVC   0(1,R03),0(R02)           MOVE 1 BYTE OF TYPE WORD         DECODE
000110 4610 C106            00118   120           BCT   R01,*+8                   COUNT LENGTH DOWN 1              DECODE
000114 47F0 C12A            0013C   121           B     LOOKPARM                  BRANCH WHEN COUNT GOES TO ZERO   DECODE
000118 4122 0001            00001   122           LA    R02,1(R02)                ADD 1 TO PARM ADDRESS            DECODE
00011C 4133 0001            00001   123           LA    R03,1(R03)                ADD 1 TO TYPE ADDRESS            DECODE
000120 957E 2000       00000        124           CLI   0(R02),C'='               CHECK FOR AN EQUAL SIGN          DECODE
000124 4780 C12A            0013C   125           BE    LOOKPARM                                                   DECODE
000128 956B 2000       00000        126           CLI   0(R02),C','               CHECK FOR A COMMA                DECODE
00012C 4780 C12A            0013C   127           BE    LOOKPARM                                                   DECODE
000130 954D 2000       00000        128           CLI   0(R02),C'('               LOOK FOR A LEFT PARENTHESIS      DECODE
000134 4780 C12A            0013C   129           BE    LOOKPARM                                                   DECODE
000138 47F0 C0F8            0010A   130           B     PMOVE                     GO TO MOVE NEXT CHARACTER        DECODE
                                    131 *                                                                        DECODE
00013C 5020 D0FC            005E4   132 LOOKPARM ST    R02,PARMPTR               SAVE ADDRESS OF DELIMITER        DECODE
000140 4130 D19C            00684   133           LA    R03,TYPTBL                GET ADDRESS OF TYPE TABLE        DECODE
000144 4140 000C            0000C   134           LA    R04,L'TYPTBL              LOAD LENGTH OF A TABLE ENTRY     DECODE
000148 4150 D274            0075C   135           LA    R05,TYPLAST               LOAD ADDRESS OF LAST ENTRY       DECODE
                                    136 *                                                                        DECODE
00014C D507 D0F0 3000 005D8 00000   137 SCANPARM CLC   CDTYP,0(R03)              LOOK FOR MATCH ON CARD TYPE      DECODE
000152 4780 C148            0015A   138           BE    GOTPARM                   BRANCH WHEN FOUND                DECODE
000156 8734 C13A            0014C   139           BXLE  R03,R04,SCANPARM          LOOP THRU TABLE                  DECODE
                                    140 *               IF THE ENTRY IS NOT FOUND TO MATCH THE TYPE ON THE       DECODE
                                    141 *               PARAMETER CARD, THE TABLE POINTER (R03) WILL BE POINTING DECODE
                                    142 *               AT AN ERROR ENTRY IN THE TABLE.                          DECODE
                                    143 *                                                                        DECODE
00015A D200 D0F8 3008 005E3 00008   144 GOTPARM  MVC   TYPNUMB+3(1),8(R03)       SET THE TYPE NUMBER FOUND        DECODE
000160 D203 B050 D0F8 00050 005E0   145           MVC   IPARM,TYPNUMB             SET THE NUMBER IN IPARM          DECODE
000166 D703 B054 B054 00054 00054   146           XC    INUMB,INUMB               ZERO THE PARAMETER COUNT         DECODE
00016C 9240 B058       00058        147           MVI   SUBPRM,C' '               BLANK OUT CHARACTER PARAMETERS   DECODE
000170 D29E B059 B058 00059 00058   148           MVC   SUBPRM+1(159),SUBPRM                                       DECODE
000176 D29F B0F8 B058 000F8 00058   149           MVC   SUBPRM+160(160),SUBPRM                                     DECODE
00017C 92FF B198       00198        150           MVI   IPCLS,X'FF'               SET ALL CLASS PARAMETERS TO -1   DECODE
000180 D29E B199 B198 00199 00198   151           MVC   IPCLS+1(159),IPCLS                                         DECODE
000186 D79F B238 B238 00238 00238   152           XC    VALUE(160),VALUE          ZERO REAL NUMBERS                DECODE
```

```
DCDE    SUBROUTINE -DECODE-  ANALYZE PARAMETER CARDS                                                    PAGE    6

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                14 SEP 70

000180 D79F B2D8 B2D8 002D8 002D8   153          XC     IVALUE(160),IVALUE       ZERO INTEGER VALUES          DECODE
                                    154 *                                                                    DECODE
000192 5813 C0C8            00C08   155          L      R01,8(R03)               LOAD THE ROUTINE ADDRESS     DECODE
000196 07F1                         156          BR     R01                      GO TO PROCESSING ROUTINE     DECODE
                                    157 *                                                                    DECODE
```

```
  LCC  CBJECT CODE     ADDR1 ADDR2  STMT   SCURCE STATEMENT                                                    14 SEP 70

0001S8 D24F ACCO DCAC CCOCC 00588   159 TTLE      MVC   TITLE(80),BLANKS          BLANK OUT EXISTING TITLE        DECODE
00019E 5810 DCFC            C05E4   160           L     R01,PARMPTR               LCAC ADDRESS OF DELIMITER       DECODE
CC01A2 4111 0C01            00001   161           LA    R01,1(R01)                BUMP ADDRESS PAST IT            DECODE
0001A6 D24F D050 DCAC C0538 00588   162           MVC   WCRK,BLANKS               BLANK OUT WORK AREA             DECODE
0001AC 4120 0048            CC048   163           LA    R02,72                    SET A SCAN LIMIT IN CASE OF     DECODE
                                    164 *                                         BLANK TITLE, WHICH MIGHT BE OK  DECODE
C001BC 954C 1CCO      CCOCO         165 LKLFTLE   CLI   0(R01),C' '               LOOK FOR LEFTMOST NON BLANK     DECODE
CC01E4 4770 C1CA            C01DC   166           BNE   TTLFRST                   BRNACH IF FOUND                 DECODE
CC01E8 4111 C001            C0001   167           LA    R01,1(R01)                INCREMENT SCAN ADDRESS          DECODE
0001BC 462C C19E            001B0   168           BCT   R02,LKLFTLE               LOOP TO CHECK AGAIN             DECODE
                                    169 *                                         FALL THRU ON BLANK TITLE SUPPLI DECODE
                                    170 *                                                                         DECODE
                                    171 *               TITLE PROCESSING DONE AT THIS POINT                       DECODE
                                    172 *                                                                         DECODE
0001C0 9201 B057      CC057         173 THRLTTL   MVI   INUMB+3,1                 SET NUMBER OF PARAMETERS TO 1   DECODE
C001C4 C203 B198 D188 C0198 00670   174           MVC   IPCLS,=F'1'               AND PARAMETER CLASS TO CHAR     DECODE
                                    175 *                                                                         DECODE
                                    176 *         EXIT BACK TO USER                                               DECODE
                                    177 *                                                                         DECODE
                                    178 FINISH    TERME RC=0                      RETURN TO CALLER (FORTRAN)      DECODE
0001CA 58DD CC04            CC004   179+FINISH    L     13,4(13) RESTCRE REGISTER -13-
0001CE 98EC DCOC            0000C   180+          LM    14,12,12(13) RESTORE THE REGISTERS
0001D2 92FF DCOC      C000C         181+          MVI   12(13),X'FF' SET RETURN INDICATION
0001D6 41FC CCCO            CCCCO   182+          LA    15,0(0,0) LOAD RETURN CODE
0001DA C7FE                         183+          BR    14 RETURN
                                    184 *                                                                         DECODE
0001DC 412C B04F            0004F   185 TTLFRST   LA    R02,PARM+79               NOW FIND LAST CHARACTER         DECODE
C001EC 9540 2000      CCCCC         186           CLI   C(RC2),C' '               IN THE TITLE                    DECODE
CC01E4 4770 C1DA            001EC   187           BNE   TTLAST                    BRANCH WHEN FOUND               DECODE
0001E8 462C C1CE            001E0   188           BCT   R02,TTLFRST+4             LOOP UNTIL FOUND                DECODE
                                    189 *               FALL THRU IS IMPOSSIBLE SINCE A NON BLANK CHARACTER       DECODE
                                    190 *               WAS FOUND PREVIOUS TO BRANCHING TO TTLFRST                DECODE
                                    191 *                                                                         DECODE
0001EC 1B21                         192 TTLAST    SR    R02,R01                   THIS YIELDS TITLE LENGTH - 1    DECODE
CC01EE 413C C050            00050   193           LA    R03,80                    LCAD TITLE MAX LENGTH           DECODE
C001F2 1B32                         194           SR    R03,R02                   GET WHAT'S UNUSED               DECODE
0001F4 883C CC01            C0001   195           SRL   R03,1                     DIVIDE IT BY 2                  DECODE
0001F8 4133 A000            00000   196           LA    R03,TITLE(R03)            GET THE CENTERED ADDRESS        DECODE
0C01FC 422C C1EF            00201   197           STC   R02,MVCTTLE+1             STORE THE MVC LENGTH            DECODE
0002C0 C200 3C00 1CCC C00C0 00000   198 MVCTTLE   MVC   C(0,R03),0(R01)           MOVE THE TITLE, CENTERED NICELY DECODE
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT			
002C6	47FC C1AE		001C0	199		B	THRUTTL	GO TO CLEAN UP THE THING
				200 *				

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT			
								14 SEP 70
0002CA	5820 L0FC		005E4	202	BLKTRTRP	L	R02,PARMPTR	LOAD POINTER AT DELIMITER	DECODE
0002CE	957E 2000		0000	203		CLI	0(R02),C'='	IS IT AN EQUAL SIGN	DECODE
0002D2	4780 C20C		0021E	204		BE	CKEQ	BRANCH IF IT IS	DECODE
				205	*				DECODE
				206	*	PARAMETER SYSTAX ERROR - TERMINATE PROCESSING			DECODE
				207	*				DECODE
0002D6	920E B053		00053	208	SYNTAX	MVI	IPARM+3,14	SET SYNTAX ERROR TYPE CODE	DECODE
0002DA	47FC C1B8		001CA	209		B	FINISH	GO TO RETURN TO CALLER	DECODE
				210	*				DECODE
0002DE	4122 0001		00001	211	CKEQ	LA	R02,1(R02)	ADD 1 TO GO PAST =	DECODE
0002E2	4130 0008		00008	212		LA	R03,8	SET LABEL LENGTH LIMIT	DECODE
0002E6	D207 D050 DCAC	00538	00588	213		MVC	WORK(8),BLANKS	BLANK OUT WORK AREA	DECODE
0002EC	4140 D050		00548	214		LA	R04,WORK	GET START OF WORK AREA	DECODE
0002F0	D200 D058 B000	00540	00000	215		MVC	WORK+8(1),PARM	SAVE B,T, OR R FROM CARD	DECODE
				216	*				DECODE
0002F6	D200 4000 2000	00000	00000	217	MVCBTR	MVC	0(1,R04),0(R02)	MOVE 1 CHARACTER FROM LABEL	DECODE
0002FC	4122 0001		00001	218		LA	R02,1(R02)	ADD 1 TO PARM POINTER	DECODE
000300	4144 0001		00001	219		LA	R04,1(R04)	ADD 1 TO WORK POINTER	DECODE
000304	956B 2000		00000	220		CLI	0(R02),C','	CHECK FOR A COMMA	DECODE
000308	4780 C26E		00280	221		BE	ANOTBTR	BRANCH IF A COMMA	DECODE
00030C	4630 C224		00236	222		BCT	R03,MVCBTR	OTHERWISE COUNT DOWN 8	DECODE
				223	*				DECODE
000310	4120 A050		00050	224	ENDBTR	LA	R02,BLOCK	LOAD ADDRESS OF BLOCK LABEL	DECODE
000314	95C2 D058	00540		225		CLI	WORK+8,C'B'	CHECK FOR BLOCK SPEC	DECODE
000318	478C C25A		00260	226		BE	*+20	BRANCH IF IT A BLOCK KEYWORD	DECODE
00031C	4120 A058		00058	227		LA	R02,TREAT	NOT BLOCK, LOAD TREAT LABEL	DECODE
000320	95E3 D05F	00540		228		CLI	WORK+8,C'T'	IS IT A TREAT KEYWORD	DECODE
000324	478C C25A		00260	229		BE	*+8	YES, BRANCH	DECODE
000328	4120 A060		00060	230		LA	R02,REPLI	NO, THEN ITS GOTTA BE A REP	DECODE
				231	*				DECODE
00032C	D207 2000 DC5C	00000	00538	232		MVC	0(8,R02),WORK	MOVE LABEL WHERE IT BELONGS	DECODE
000332	D50C BCC0 DC58	00000	00540	233		CLC	PARM(1),WORK+8	CHECK FOR ADDITIONAL PARMS	DECODE
000338	477C C0E0		000F2	234		BNE	GOTCATE	GO PROCESS THEM IF THERE	DECODE
				235	*				DECODE
00033C	47F0 C1AE		001C0	236	THRUBTR	B	THRUTTL	NOW THRU WITH BLOCK,TREAT, REP	DECODE
				237	*				DECODE
000340	D240 BC00 2001	00000	00001	238	ANOTBTR	MVC	PARM(65),1(R02)	A SECOND PARAMETER WAS FOUND,	DECODE
000346	47FC C23E		00250	239		B	ENDBTR	SHIFT IT DOWN FOR DECODING	DECODE
				240	*			AFTER FINISHING WITH THIS ONE	DECODE
				241	*				DECODE

```
  LCC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

0002EA 4120 C001             00001  243 LBLS     LA    R02,1                  SET THE DEFAULT SUBSCRIPT        DECODE
0002EE 957F B005       C0005        244          CLI   PARM+5,C'='            CHECK FOR EQUAL SIGN             DECODE
000292 478C C20A             002EC  245          BE    SUBSET                 BRANCH IF =, NO SUBSCRIPT        DECODE
000296 954D B005       C0005        246          CLI   PARM+5,C'('            IT NOW MUST BE A LEFT PAREN      DECODE
00029A 477C C204             00216  247          BNE   SYNTAX                 SYNTAX ERROR IF NOT              DECODE
00029E 4130 B006             00006  248          LA    R03,PARM+6             SET START POINTER FOR            DECODE
0002A2 1843                         249          LR    R04,R03                SUBSCRIPT SCAN                   DECODE
                                    250 *                                                                       DECODE
0002A4 91F0 4000       C0000        251 LBLLP    TM    0(R04),X'F0'           TEST FOR A NUMERIC CHARACTER     DECODE
0002A8 47E0 C2A2             002B4  252          BNO   ENDLBLP                BRANCH WHEN NOT NUMERIC          DECODE
0002AC 4144 C001             00001  253          LA    R04,1(R04)             ADD 1 TO POINTER                 DECODE
0002B0 47F0 C292             002A4  254          B     LBLLP                  CONTINUE TO LOOP UNTIL           DECODE
                                    255 *                                                                       DECODE
0002B4 955D 4000       C0000        256 ENDLBLP  CLI   0(R04),C')'            IT BETTER BE A RIGHT PAREN       DECODE
0002B8 477C C204             00216  257          BNE   SYNTAX                 AN ERROR IF IT ISN'T             DECODE
0002BC 1934                         258          CR    R03,R04                CHECK FOR CODING AS ()           DECODE
0002BE 4780 C204             00216  259          BE    SYNTAX                 ERROR IF NO SUBSCRIPT            DECODE
0002C2 1B43                         260          SR    R04,R03                CALCULATE SUBSCRIPT LENGTH       DECODE
0002C4 5940 D18C             00674  261          C     R04,=F'2'              CHECK AGAINST MAX LENGTH         DECODE
0002C8 4720 C204             00216  262          BH    SYNTAX                 ERROR IF LONGER THAN 2           DECODE
0002CC 4144 C06F             0006F  263          LA    R04,X'6F'(R04)         BUILD LENGTH FOR PACK INST       DECODE
                                    264 *              IF YOU CAN'T SEE WHY THIS WORKS, TRY IT ON PAPER         DECODE
0002D0 424C C2C3             002D5  265          STC   R04,LBLPK+1            STORE THIS LENGTH                DECODE
0002D4 F207 D048 B006 00530  00006  266 LBLPK    PACK  DOUBLE(0),PARM+6       PACK THE SUBSCRIPT               DECODE
0002DA 4F20 D048             00530  267          CVB   R02,DOUBLE             CONVERT IT TO BINARY             DECODE
0002DE 1222                         268          LTR   R02,R02                TEST FOR A ZERO SUBSCRIPT        DECODE
0002E0 478C C204             00216  269          BZ    SYNTAX                 THAT'S A NO-NO                   DECODE
0002E4 592C D190             00678  270          C     R02,=F'40'             CHECK AGAINST MAXIMUM            DECODE
0002E8 472C C204             00216  271          BH    SYNTAX                 THAT'S ALSO AN ERROR             DECODE
                                    272 *                                                                       DECODE
0002EC 502C D100             005E8  273 SUBSET   ST    R02,SUBSCRPT           STORE DEFAULT OR SUPPLIED SUB    DECODE
0002F0 458C C33C             0034E  274          BAL   R08,BLDSUBS            GO TO DIRTY WOR ROUTINE          DECODE
0002F4 918C D114       C05FC        275          TM    SUBSW8,X'80'           WERE ANY PARAMETERS LONGER       DECODE
0002F8 471C C204             00216  276          BC    SYNTAX                 THAN 8 CHARACTERS.  IF SO,       DECODE
                                    277 *                                    CALL IT A SYNTAX ERROR.          DECODE
                                    278 *                                                                       DECODE
0002FC 5820 D100             005EB  279          L     R02,SUBSCRPT           LOAD THE SUBSCRIPT               DECODE
000300 8920 0003             00003  280          SLL   R02,3                  MULTIPLY BY 8                    DECODE
000304 4122 A060             00060  281          LA    R02,LABEL-8(R02)       ADD LABEL BASE ADDRESS           DECODE
000308 4130 B058             00058  282          LA    R03,SUBPRM             LOAD PARAMETER POINTER           DECODE
```

```
DCDE   DECODE LABEL SPECIFICATIONS                                                                    PAGE   11

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                14 SEP 70

000300 5840 B054             00054   283          L     R04,INUMB          LOAD NUMBER OF LABELS          DECODE
000310 1244                          284          LTR   R04,R04            CHECK FOR NO LABELS            DECODE
000312 4780 C204             00216   285          BZ    SYNTAX             ERROR IF ZERO                  DECODE
000316 1854                          286          LR    R05,R04            MOVE NUMBER FOR TOO MANY CHECK DECODE
000318 5A50 D100             005E8   287          A     R05,SUBSCRPT       ADD STARTING SUBSCRIPT         DECODE
00031C 5950 D194             0067C   288          C     R05,=F'41'         CHECK AGAINST MAX              DECODE
000320 4720 C204             00216   289          BH    SYNTAX             ERROR IF TOO MANY LABELS       DECODE
                                     290 *                                                                 DECODE
000324 D207 2000 3000 00000  00000   291 MVLBLLP  MVC   0(8,R02),0(R03)    MOVE 1 LABEL                   DECODE
00032A 4122 0008             00008   292          LA    R02,8(R02)         ADVANCE LABEL POINTER          DECODE
00032E 4133 0008             00008   293          LA    R03,8(R03)         ADVANCE SUBPRM POINTER         DECODE
000332 4640 C312             00324   294          BCT   R04,MVLBLLP        COUNT DOWN UNTIL ALL MOVED     DECODE
                                     295 *                                                                 DECODE
000336 47F0 C188             001CA   296 THRLLBLS B     FINISH             LABEL PROCESSING IS THROUGH    DECODE
                                     297 *                                                                 DECODE
```

```
DCDE   DECODE ALL OTHER SPECIFICATION STATEMENTS                                                                  PAGE   12

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                        14 SEP 70

00033A 4580 C33C            0034E   299 GENERAL  BAL   R08,BLDSUBS          GO BUILD PARAMETER LISTS              DECODE
00033E 47F0 C1B8            001CA   300          B     FINISH                                                     DECODE
                                    301 *                                                                          DECODE
00033A                              302 DATA     EQU   GENERAL              DATA STATEMENT                        DECODE
00033A                              303 CODING   EQU   GENERAL              CODING STATEMENT                      DECODE
00033A                              304 SRCE     EQU   GENERAL              SOURCE STATEMENT                      DECODE
00033A                              305 DIST     EQU   GENERAL              DISTRIB STATEMENT                     DECODE
00033A                              306 CLSS     EQU   GENERAL              CLASSES STATEMENT                     DECODE
00033A                              307 TEST     EQU   GENERAL              TESTS STATEMENT                       DECODE
00033A                              308 TRAN     EQU   GENERAL              TRANS STATEMENT                       DECODE
                                    309 *                                                                          DECODE
000342 47F0 C1B8            001CA   310 EOP      B     FINISH               END OF PARAMETERS STATEMENT           DECODE
                                    311 *                                                                          DECODE
000346 47F0 C1B8            001CA   312 EOR      B     FINISH               END OF RUN OR END OF JOB              DECODE
                                    313 *                                                                          DECODE
00034A 47F0 C1B8            001CA   314 NOFIND   B     FINISH               ERROR - TYPE NOT RECOGNIZED           DECODE
```

```
CCDE   SUBROUTINE TC SEPARATE AND ANALYZE EACH PARAMETER                                                    PAGE   13

  LCC  CBJECT CODE    ADDR1 ADDR2  STMT   SCURCE STATEMENT                                                  14 SEP 70

                                   316 *              THIS SUBROUTINE SCANS THE PARAMETER CARD AND BUILDS      BLDSUB
                                   317 *         A LIST OF ALL PARAMETERS FOLLOWING THE FIRST COMMA OR         BLDSUB
                                   318 *         EQUAL SIGN.  THIS MEANS THAT MOST PARAMETERS COULD BE         BLDSUB
                                   319 *         SPECIFIED BY EITHER 'ABC,XX,YY,ZZ,ETC' OR AS 'ABC=XX,YY,ETC'.  BLDSUB
                                   320 *         THE LAST PARAMETER IS TERMINATED BY THE REST OF THE CARD      BLDSUB
                                   321 *         BEING BLANK, OR BY ENDING IN COLUMN 80.  EACH PARAMETER MAY   BLDSUB
                                   322 *         CONTAIN ANY SPECIAL CHARACTERS (INCLUDING BLANK), EXCEPT      BLDSUB
                                   323 *         A COMMA.  PARAMETERS THAT ARE LONGER THAN 8 CHARACTERS WILL   BLDSUB
                                   324 *         BE CONSIDERED 2 OR MORE PARAMETERS AND COUNTED ACCORDINGLY    BLDSUB
                                   325 *         WITH NO INDICATION OF THEIR OCCURANCE.                        BLDSUB
                                   326 *                                                                       BLDSUB
                                   327 *              THE OUTPUT OF THIS SUBROUTINE IS PLACED INTO SPECIFIC   BLDSUB
                                   328 *         ARRAYS IN THE COMMON BLOCK /INPARM/.  NUMERIC CONVERSIONS     BLDSUB
                                   329 *         ARE ATTEMPTED ON ALL PARAMETERS AND THOSE WHICH DON'T CONVERT BLDSUB
                                   330 *         WITHOUT ERROR ARE CONSIDERED CHARACTER.  THIS CONVERSION      BLDSUB
                                   331 *         IS PERFORMED IN THE EXTERNAL ROUTINE 'IHBFFFIO'.              BLDSUB
                                   332 *                                                                       BLDSUB
00034E 9200 D114      005FC        333 BLDSUBS  MVI   SUBSWB,0                  CLEAR INDICATOR SWITCHES       BLDSUB
000352 4110 B000            00000  334          LA    R01,PARM                  PICK UP ADDRESS OF CARD        BLDSUB
000356 4120 000A            0000A  335          LA    R02,10                    SET A SCAN LIMIT               BLDSUB
00035A 956B 1000      00000        336 FINDST   CLI   0(R01),C','               LOOK FOR A COMMA               BLDSUB
00035E 4780 C364            00376  337          BE    GOTPSTRT                                                 BLDSUB
000362 957E 1000      00000        338          CLI   0(R01),C'='               OR AN EQUAL SIGN               BLDSUB
000366 478C C364            00376  339          BE    GOTPSTRT                                                 BLDSUB
00036A 4111 0001            00001  340          LA    R01,1(R01)                ADD 1 TO ADDRESS               BLDSUB
00036E 4620 C348            0035A  341          BCT   R02,FINDST                COUNT DOWN AND LOOP            BLDSUB
000372 47FC C204            00216  342          B     SYNTAX                    SYNTAX ERROR WHEN NOT FOUND    BLDSUB
                                   343 *                                                                       BLDSUB
000376 5010 D104            005EC  344 GOTPSTRT ST    R01,PARMSTRT              SAVE THE ADDRESS               BLDSUB
00037A 4130 0001            00001  345          LA    R03,1                     INITIALIZE COUNT FIELD         BLDSUB
00037E 4122 0045            00045  346          LA    R02,69(R02)               SET SCAN LIMIT COUNT           BLDSUB
000382 4141 0001            00001  347          LA    R04,1(R01)                ADVANCE POINTER FOR SCAN       BLDSUB
000386 956B 4000      00000        348 COUNTPRM CLI   0(R04),C','               COUNT THE COMMAS               BLDSUB
00038A 4770 C380            00392  349          BNE   *+8                                                      BLDSUB
00038E 4133 0001            00001  350          LA    R03,1(R03)                ADD TO COUNT OF PARAMETERS     BLDSUB
000392 4144 0001            00001  351          LA    R04,1(R04)                INCREMENT ADDRESS              BLDSUB
000396 4620 C374            00386  352          BCT   R02,COUNTPRM              DECREMENT COUNTER              BLDSUB
                                   353 *                                                                       BLDSUB
00039A 5030 BC54            00054  354          ST    R03,INUMB                 STORE INITIAL COUNT            BLDSUB
                                   355 *                                                                       BLDSUB
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    BLDSUB
000396 4170 B058             0005B   356            LA    R07,SUBPRM      PARAMETERS ARRAY START              BLDSUB
00039A 4160 B198             00198   357            LA    R06,IPCLS       PARAMETER CLASS ARRAY               BLDSUB
00039E 4151 0001             00001   358            LA    R05,1(R01)      SCAN STARTING ADDRESS               BLDSUB
0003A2 4140 B050             00050   359            LA    R04,PARM+80     COLUMN 81                           BLDSUB
0003A6 1845                          360            SR    R04,R05         NUMBER OF COLUMNS LEFT              BLDSUB
                                     361 *                                INCLUDING THIS ONE.                 BLDSUB
0003A8 956B 5000       00000         362 SUBLP      CLI   0(R05),C','    CHECK FOR A COMMA                   BLDSUB
0003AC 4770 C3B2             003C4   363            BNE   NOTNULL         BRANCH IF IT ISN'T                  BLDSUB
0003B0 4155 0001             00001   364            LA    R05,1(R05)      BUMP ADDRESS                        BLDSUB
0003B4 4640 C412             00424   365            BCT   R04,NEXTONE     OMITTED PARAMETER COUNT DOWN        BLDSUB
0003B8 47F0 C436             00448   366            B     ENDSUB          IF LAST COLUMN SCAN THROUGH         BLDSUB
                                     367 *                                                                    BLDSUB
0003BC D203 6000 0188 00000  00670   368 NOTNULL    MVC   0(4,R06),=F'1'  SET INITIALLY TO CHARACTER          BLDSUB
0003C2 1817                          369            LR    R01,R07         GET SUBPARAMETER ADDRESS            BLDSUB
0003C4 4120 0008             00008   370            LA    R02,8           SET PARAMETER LENGTH                BLDSUB
0003C8 D200 1000 5000 00000  00000   371 MVCSUB     MVC   0(1,R01),0(R05) MOVE 1 BYTE                         BLDSUB
0003CC 4155 0001             00001   372            LA    R05,1(R05)      INCREMENT ACROSS CARD               BLDSUB
0003D0 4111 0001             00001   373            LA    R01,1(R01)      INCREMENT WITHIN SUBPRM             BLDSUB
0003D4 4640 C3B4             003F6   374            BCT   R04,*+8         DECREMENT COLUMN COUNT              BLDSUB
0003D8 47FC C436             00448   375            B     ENDSUB          IF THE LAST ONE WAS USED            BLDSUB
0003DC 956B 5000       00000         376            CLI   0(R05),C','     CHECK FOR THE COMMA SEPARATOR       BLDSUB
0003E0 4780 C406             00418   377            BE    DELIMIT         YES, THRU WITH THIS ONE             BLDSUB
0003E4 4620 C3BE             003D0   378            BCT   R02,MVCSUB      COUNT DOWN AND LOOP                 BLDSUB
0003E8 1834                          379            LR    R03,R04                                             BLDSUB
0003EA C630                          380            BCTR  R03,R00         SUBTRACT 1 FOR INSTRUCTION          BLDSUB
0003EC 423C C3E9             003FB   381            STC   R03,CLCTRM+1    STORE FOR COMPARE                   BLDSUB
0003F0 D500 5000 0CAC 00000  00588   382 CLCTRM     CLC   0(0,R05),BLANKS CHECK FOR REST FOR CARD BLANK       BLDSUB
0004CC 4780 C436             00448   383            BE    ENDSUB          YES, WE'RE THRU                     BLDSUB
                                     384 *                                                                    BLDSUB
0004C4 9680 D114       00FFC         385            OI    SUBSW8,X'80'    SET LONG PARAMETER FLAG             BLDSUB
0004C8 5830 B054             00054   386            L     R03,INUMB       ADD 1 TO PARAMETER COUNT            BLDSUB
0004CC 4133 0001             00001   387            LA    R03,1(R03)                                          BLDSUB
0004D0 5030 B054             00054   388            ST    R03,INUMB       STORE THE ADJUSTED COUNT            BLDSUB
0004D4 47F0 C412             00424   389            B     NEXTONE         GO TO NEXTONE SETUP                 BLDSUB
                                     390 *                                                                    BLDSUB
0004D8 4155 0001             00001   391 DELIMIT    LA    R05,1(R05)      ADJUST PAST COMMA                   BLDSUB
0004DC 4640 C412             00424   392            BCT   R04,NEXTONE     REMOVE 1 FROM COLUMN COUNT          BLDSUB
0004E0 47F0 C204             00216   393            B     SYNTAX          ERROR - COMMA IN COLUMN 80          BLDSUB
                                     394 *                                                                    BLDSUB
0004E4 4177 0008             00008   395 NEXTONE    LA    R07,8(R07)      POINT TO NEXT SUBPRM                BLDSUB
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000428 4166 0004             00004   396          LA    R06,4(R06)              POINT TO NEXT IPCLS              BLDSUB
00042C 4190 B234             00234   397          LA    R09,IPCLS+40*4-4        CALCULATE LIMIT ADDRESS          BLDSUB
000430 1569                          398          CR    R06,R09                 CHECK FOR TOO MANY PARAMETERS    BLDSUB
000432 4720 C2C4             00216   399          BH    SYNTAX                  ERROR IF MORE THAN 40            BLDSUB
000436 1834                          400          LR    R03,R04                 GET COUNT OF WHAT'S LEFT         BLDSUB
000438 C63C                          401          BCTR  R03,R00                 SUBTRACT 1 FOR COMPARE           BLDSUB
00043A 4230 C42D             00439   402          STC   R03,CLCBL+1             STORE LENGTH IN COMPARE          BLDSUB
00043E D500 5000 0CAC 00000  00588   403 CLCBL    CLC   0(0,R05),BLANKS         CHECK FOR REST OF CARD BLANK     BLDSUB
000444 4770 C39E             00380   404          BNE   SUBLP                   BRANCH TO GET NEXT PARM IF NOT   BLDSUB
                                     405 *                                                                       BLDSUB
000448 413C B058             00058   406 ENDSUB   LA    R03,SUBPRM              SET ADDRESS TO START             BLDSUB
00044C 4140 B198             00198   407          LA    R04,IPCLS               CLASS ARRAY                      BLDSUB
000450 4150 B238             00238   408          LA    R05,VALUE               REAL VALUE ARRAY                 BLDSUB
000454 4160 B2D8             002D8   409          LA    R06,IVALUE              INTEGER VALUE ARRAY              BLDSUB
000458 5870 B054             00054   410          L     R07,INUMB               NUMBER TO BE CHECKED             BLDSUB
                                     411 *                                                                       BLDSUB
00045C 5824 0000             00000   412 CNVLOOP  L     R02,0(R04)              GET CLASS INDICATOR              BLDSUB
000460 1222                          413          LTR   R02,R02                 TEST IT                          BLDSUB
000462 474C C4BA             004CC   414          BL    NEXTCNV                 BRANCH TO SKIP OMITTED PARMS     BLDSUB
000466 D507 3000 0CAC 00000  00588   415          CLC   0(8,R03),BLANKS         CHECK FOR A BLANK ONE            BLDSUB
00046C 478C C4BA             004CC   416          BE    NEXTCNV                 SKIP CHECKING THEM TOO           BLDSUB
                                     417 *                                                                       BLDSUB
                                     418          FPIN  0(R05),(R03),F8.0,S     CONVERT TO FLOATING POINT        BLDSUB
000470 41C5 0000             00000   419+         LA    0,0(R05) LOAD ADDRESS OF FP NUMBER
000474 1813                          420+         LR    1,R03 LOAD ADDRESS OF CHAR STRING
000476 58F0 D198             00680   421+         L     15,=V(IHBFFFIO) LOAD FP ROUTINE ADDRESS
00047A 45EF 3010             00010   422+         BAL   14,16(15) LINK TO FP INPUT ROUTINE
00047E C800                          423+         DC    AL1(8,0) FORMAT SPECIFICATION
000480 12FF                          424          LTR   R15,R15                 CHECK FOR CONVERSION ERRORS      BLDSUB
000482 4770 C4BA             004CC   425          BNZ   NEXTCNV                 LEAVE AS CHARACTER PARAMETER     BLDSUB
                                     426          FIX   0(R06),0(R05),S         MAKE AN INTEGER OF IT            BLDSUB
000486 41C6 0000             00000   427+         LA    0,0(R06) LOAD ADDRESS OF INTEGER
00048A 4115 0000             00000   428+         LA    1,0(R05) LOAD ADDRESS OF FP NUMBER
00048E 58F0 D198             00680   429+         L     15,=V(IHBFFFIO) LOAD FP ROUTINE ADDRESS
000492 05EF                          430+         BALR  14,15 LINK TO FIX(SINGLE) ROUTINE
000494 12FF                          431          LTR   R15,R15                 TEST FOR FIX ERROR               BLDSUB
000496 4770 C4B4             004C6   432          BNZ   REALNUM                 NUMBER TOO BIG TO FIX            BLDSUB
00049A 5826 0000             00000   433          L     R02,0(R06)              LOAD THE INTEGER VALUE           BLDSUB
00049E 1222                          434          LTR   R02,R02                 CHECK FOR POSSIBLE NEGATIVE      BLDSUB
0004A0 4740 C4B4             004C6   435          BL    REALNUM                 CONSIDER REAL IF IT IS           BLDSUB
```

```
DCDE    SUBROUTINE TO SEPARATE AND ANALYZE EACH PARAMETER                                                    PAGE   16

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                  14 SEP 70

                                     436           FLOAT (FPR0),0(R06),S          CONVERT IT BACK TO REAL          BLDSUB
0004A4 4100 0000             00000   437+          LA    0,FPR0 LOAD FP REGISTER NUMBER
0004A8 4116 0000             00000   438+          LA    1,0(R06) LOAD ADDRESS OF INTEGER NUMBER
0004AC 58F0 0198             00680   439+          L     15,=V(IHBFFFIO) LOAD FP ROUTINE ADDRESS
0004B0 45FF 0008             00008   440+          BAL   14,8(15) LINK TO FLOAT(SINGLE) ROUTINE
0004B4 7905 0000             00000   441           CE    FPR0,0(R05)              COMPARE THE REAL NUMBERS         BLDSUB
0004B8 477C 04B4             004C6   442           BNE   REALNUM                  BRANCH IF NOT EQUAL              BLDSUB
                                     443 *                                                                         BLDSUB
0004BC D703 400C 400C 00000 00000    444 INTNUM    XC    0(4,R04),0(R04)          SET CLASS TO INTEGER WITH A 0    BLDSUB
0004C2 47F0 04BA             004CC   445           B     NEXTCNV                  GO TRY NEXT ONE                  BLDSUB
                                     446 *                                                                         BLDSUB
0004C6 D203 400C 018C 00000 00674    447 REALNUM   MVC   0(4,R04),=F'2'           SET IT TO A REAL NUMBER          BLDSUB
                                     448 *                                                                         BLDSUB
0004CC 4133 0008             00008   449 NEXTCNV   LA    R03,8(R03)               INCREMENT SUBPRM                 BLDSUB
0004D0 4144 0004             00004   450           LA    R04,4(R04)               INCREMENT IPCLS                  BLDSUB
0004D4 4155 0004             00004   451           LA    R05,4(R05)               INCREMENT VALUE                  BLDSUB
0004D8 4166 0004             00004   452           LA    R06,4(R06)               INCREMENT IVALUE                 BLDSUB
0004DC 4670 C44A             0045C   453           BCT   R07,CNVLOOP              COUNT THE PARAMETERS DOWN        BLDSUB
                                     454 *                                                                         BLDSUB
0004E0 07F8                          455           BR    R08                      RETURN TO CALLER - DONE          BLDSUB
```

```
DCDE    WORK AREAS AND TABLES NEEDED FOR DECODING PARAMETERS                                                    PAGE   17

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                        14 SEP 70

0004E2 000000000000
0004E8 0000000000000000                457 SAVEAREA DC    9D'0'                    REGISTER SAVE AREA FOR OS       DCDEDATA
000530                                 458 DOUBLE   DS    D                        DOUBLE WORD WORK AREA           DCDEDATA
000538                                 459 WORK     DS    CL80                     MISC WORK AREA                  DCDEDATA
0005E8 404040404040404040              460 BLANKS   DC    CL80' '                  BLANKS FOR COMPARISONS          DCDEDATA
0005D8                                 461 CDTYP    DS    CL8                      AREA FOR CARD NAME              DCDEDATA
0005E0                                 462 TYPNUMB  DS    F                        CARD TYPE NUMBER                DCDEDATA
0005E4                                 463 PARMPTR  DS    A                        ADDRESS OF DELIMITER AFTER TYPE DCDEDATA
0005E8                                 464 SUBSCRPT DS    F                        HOLD AREA FOR LABEL SUBSCRIPT   DCDEDATA
0005EC                                 465 PARMSTRT DS    A                        ADDRESS OF START OF PARAMETERS  DCDEDATA
0005F0 1F001F1E1F1E1F1F                466 MONTBL   DC    AL1(31,0,31,30,31,30,31,31,30,31,30,31)                  DCDEDATA
0005F1                                 467 FEBDAY   EQU   MONTBL+1                 DAYS IN EACH MONTH OF DATE      DCDEDATA
                                       468 *                                                                        DCDEDATA
0005F0 00                              469 SUBSW8   DC    X'00'                    LONG PARAMETER SWITCH           DCDEDATA
```

```
DCDE   WORK AREAS AND TABLES NEEDED FOR DECODING PARAMETERS                                                    PAGE   18

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                      14 SEP 70

                                   471          REGS  STD                  REGISTER NAME EQUATES          DCDEDATA
000000                             472+R00      EQU   0
000001                             473+R01      EQU   1
000002                             474+R02      EQU   2
000003                             475+R03      EQU   3
000004                             476+R04      EQU   4
000005                             477+R05      EQU   5
000006                             478+R06      EQU   6
000007                             479+R07      EQU   7
000008                             480+R08      EQU   8
000009                             481+R09      EQU   9
00000A                             482+R10      EQU   10
00000B                             483+R11      EQU   11
00000C                             484+R12      EQU   12
00000D                             485+R13      EQU   13
00000E                             486+R14      EQU   14
00000F                             487+R15      EQU   15
                                   488          REGS  FPR                                                    DCDEDATA
000000                             489+FPR0     EQU   0
000002                             490+FPR2     EQU   2
000004                             491+FPR4     EQU   4
000006                             492+FPR6     EQU   6
                                   493 *                                                                     DCDEDATA
```

```
                                  495 PATCH    PATCH 100
0005FD CC
0005FE                            496+         DC    0F'0'
0005FE                            497+PATCH    DS    0CL100
0005FE C116                       498+         DC    S(*)
000600 C118                       499+         DC    S(*)
000602 C11A                       500+         DC    S(*)
000604 C11C                       501+         DC    S(*)
000606 C11E                       502+         DC    S(*)
000608 C120                       503+         DC    S(*)
00060A C122                       504+         DC    S(*)
00060C C124                       505+         DC    S(*)
00060E C126                       506+         DC    S(*)
000610 C128                       507+         DC    S(*)
000612 C12A                       508+         DC    S(*)
000614 C12C                       509+         DC    S(*)
000616 C12E                       510+         DC    S(*)
000618 C130                       511+         DC    S(*)
00061A C132                       512+         DC    S(*)
00061C C134                       513+         DC    S(*)
00061E C136                       514+         DC    S(*)
000620 C138                       515+         DC    S(*)
000622 C13A                       516+         DC    S(*)
000624 C13C                       517+         DC    S(*)
000626 C13E                       518+         DC    S(*)
000628 C140                       519+         DC    S(*)
00062A C142                       520+         DC    S(*)
00062C C144                       521+         DC    S(*)
00062E C146                       522+         DC    S(*)
000630 C148                       523+         DC    S(*)
000632 C14A                       524+         DC    S(*)
000634 C14C                       525+         DC    S(*)
000636 C14E                       526+         DC    S(*)
000638 C150                       527+         DC    S(*)
00063A C152                       528+         DC    S(*)
00063C C154                       529+         DC    S(*)
00063E C156                       530+         DC    S(*)
000640 C158                       531+         DC    S(*)
000642 C15A                       532+         DC    S(*)
000644 C15C                       533+         DC    S(*)
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT
000646 D15E                        534+          DC    S(*)
000648 D160                        535+          DC    S(*)
00064A D162                        536+          DC    S(*)
00064C D164                        537+          DC    S(*)
00064E D166                        538+          DC    S(*)
000650 D168                        539+          DC    S(*)
000652 D16A                        540+          DC    S(*)
000654 D16C                        541+          DC    S(*)
000656 D16E                        542+          DC    S(*)
000658 D170                        543+          DC    S(*)
00065A D172                        544+          DC    S(*)
00065C D174                        545+          DC    S(*)
00065E D176                        546+          DC    S(*)
000660 D178                        547+          DC    S(*)
                                   548 *
```

```
DCDE   WORK AREAS ANC TABLES NEEDED FOR DECCDING PARAMETERS                                                    PAGE   21

  LCC  CBJECT CODE    ACDR1 ADDR2  STMT   SCURCE STATEMENT                                                      14 SEP 70

000668                                 550 LITERALS LTCRG                          LITERALS USED IN THE PROGRAM    DCDLDATA
000668 CCOCCCCCCCOCCCCO                551                 =V(PRETTY,INPARM)
00067C CCCCC001                        552                 =F'1'
000674 CCCCCCC2                        553                 =F'2'
000678 CCCCCC28                        554                 =F'40'
00067C CCOCCC29                        555                 =F'41'
00068C CCOCCOCO                        556                 =V(IHBFFFIO)
                                       557 *                                                                       DCDEDATA
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

000684                                 DC    0F'0'                    FORCE FULL WORD ALIGNMENT          DCDEDATA
000684                                 TYPTBL DS    0CL12                    TYPE DECODING TABLE                DCDEDATA
                                       *             EACH TABLE ENTRY IS 12 BYTES LONG AND CONTAINS 3 FIELDS  DCDEDATA
                                       *             THE FIRST IS THE 8 CHARACTER CARD NAME.                  DCDEDATA
                                       *             THE SECOND IS A 1 BYTE BINARY NUMBER REPRESENTING THE    DCDEDATA
                                       *             CARD TYPE. THIS NUMBER IS PLACED IN IPARM IN /INPARM/.   DCDEDATA
                                       *             THE THIRD IS A 3 BYTE ADDRESS OF THE PROCESSING ROUTINE  DCDEDATA
                                       *             USED TO ACTUALLY DECODE THE PARAMETERS.                  DCDEDATA
000684 E3C9E3D3C54C4C40                DC    CL8'TITLE',AL1(1),AL3(TTLE)                                    DCDEDATA
00069C C4C1E3C14C4C4C40                DC    CL8'DATA',AL1(2),AL3(DATA)                                     DCDEDATA
00069C C2D3D6C3D24C4C40                DC    CL8'BLOCK',AL1(3),AL3(BLKTRTRP)                                DCDEDATA
0006A8 E3D9C5C1E34C4C40                DC    CL8'TREAT',AL1(3),AL3(BLKTRTRP)                                DCDEDATA
0006B4 D9C5D74C4C4C4C40                DC    CL8'REP  ',AL1(3),AL3(BLKTRTRP)                                DCDEDATA
0006CC D3C1C2C5D34C4C40                DC    CL8'LABEL',AL1(4),AL3(LBLS)                                    DCDEDATA
0006CC C3D6C4C9D5C74C40                DC    CL8'CODING',AL1(5),AL3(CODING)                                 DCDEDATA
0006D8 E2D6E4D9C3C54C40                DC    CL8'SOURCE',AL1(6),AL3(SRCE)                                   DCDEDATA
0006E4 C4C9E2E3D9C9C240                DC    CL8'DISTRIB',AL1(7),AL3(DIST)                                  DCDEDATA
0006FC C3D3C1E2E2C5E240                DC    CL8'CLASSES',AL1(8),AL3(CLSS)                                  DCDEDATA
0006FC C3D3C1E2E24C4C40                DC    CL8'CLASS  ',AL1(8),AL3(CLSS) ALLOWABLE ALIAS                  DCDEDATA
000708 E3C5E2E3E24C4C40                DC    CL8'TESTS',AL1(9),AL3(TEST)                                    DCDEDATA
000714 E3C5E2E34C4C4C40                DC    CL8'TEST ',AL1(9),AL3(TEST)   ALLOWABLE ALIAS                  DCDEDATA
00072C E3D9C1D5E24C4C40                DC    CL8'TRANS',AL1(10),AL3(TRAN)                                   DCDEDATA
00072C E3D9C1D54C4C4C40                DC    CL8'TRAN ',AL1(10),AL3(TRAN)  ALLOWABLE ALIAS                  DCDEDATA
000738 C5D5C44C4C4C4C40                DC    CL8'END',AL1(11),AL3(EOP)     ALIAS OF END OF PARAMETERS       DCDEDATA
000744 C5D5C440D6C64CD7                DC    CL8'END OF P',AL1(11),AL3(EOP)                                 DCDEDATA
00075C C5D5C440D6C64CD9                DC    CL8'END OF R',AL1(12),AL3(EOR)                                 DCDEDATA
00075C                                 TYPLAST EQU   *                        LAST VALID TYPE FOLLOWS          DCDEDATA
00075C C5D5C440D6C64CD1                DC    CL8'END OF J',AL1(12),AL3(EOR)  ALIAS OF END OF RUN            DCDEDATA
00076E E4D5D2D5D6E6D540                DC    CL8'UNKNOWN',AL1(13),AL3(NOFIND)   ERROR CONDITION             DCDEDATA
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        14 SEP 70
                                    589 *              THIS DUMMY SECTION IS EQUIVALENT TO THE COMMON          PRETTY
                                    590 *              BLOCK /PRETTY/, WHICH IS USED TO CONTAIN INFORMATION    PRETTY
                                    591 *              WHICH IS USED TO LABEL THE PRINTED OUTPUT OF THE        PRETTY
                                    592 *              EXPERIMENT.  SOME OF THIS INFORMATION IS SUPPLIED       PRETTY
                                    593 *              BY THE USER, WHILE OTHER IS PROVIDED AUTOMATICALLY      PRETTY
                                    594 *              BY THE SUBROUTINE 'DECODE'.  THE INFORMATION            PRETTY
                                    595 *              CONTAINED IN THIS BLOCK INCLUDES THE EXPERIMENT TITLE,  PRETTY
                                    596 *              VARIABLE LABELS, DATE OF RUN, TIME OF RUN, PAGE NUMBER, PRETTY
                                    597 *              AND CURRENT PAGE LINE NUMBER.                           PRETTY
000000                              598 PRTY     DSECT                  DSECT FOR /PRETTY/                   PRETTY
000000                              599 TITLE    DS    10D              REAL*8 TITLE(10)                     PRETTY
000050                              600 BLOCK    DS    D                REAL*8 BLOCK    BLOCK LABEL          PRETTY
000058                              601 TREAT    DS    D                REAL*8 TREAT    TREATMENT LABEL      PRETTY
000060                              602 REPLI    DS    D                REAL*8 REPLI    REPLICATION LBL      PRETTY
000068                              603 LABEL    DS    40D              REAL*8 LABEL(40)                     PRETTY
0001A8                              604 DATE     DS    D                REAL*8 DATE     MM/DD/YY             PRETTY
0001B0                              605 TIME     DS    D                REAL*8 TIME     HH:MM:SS             PRETTY
0001B8                              606 PAGE     DS    F                INTEGER*4 PAGE                       PRETTY
0001BC                              607 LINE     DS    F                INTEGER*4 LINE                       PRETTY
0001C0                              608 LENGPRTY EQU   *-PRTY           LENGTH OF THIS BLOCK                 PRETTY
```

```
                                610 *              THIS COMMON BLOCK IS USED TO PASS INPUT PARAMETER        INPARM
                                611 *              INFORMATION BETWEEN SUBROUTINES.  IT CONTAINS THE        INPARM
                                612 *              PARAMETER INFORMATION OF 1 PARAMETER CARD IN ITS RAW     INPARM
                                613 *              (CARD IMAGE) FORM FOR ANALYSIS BY THE SUBROUTINE         INPARM
                                614 *              'DECODE', WHICH FILLS THE REST OF THE BLOCK WITH THE     INPARM
                                615 *              CONVERTED AND DECODED VALUES OF ALL THE PARAMETERS       INPARM
                                616 *              ENCOUNTERED.                                             INPARM
000CC0                          617 PARMS    DSECT                DSECT FOR COMMON BLOCK /INPARM/          INPARM
000CC0                          618 PARM     DS    10D            REAL*8 PARM(10)  CARD IMAGE              INPARM
000D50                          619 IPARM    DS    F              INTEGER*4 IPARM  TYPE OF CARD            INPARM
                                620 *                             1 = TITLE STATEMENT                      INPARM
                                621 *                             2 = DATA STATEMENT                       INPARM
                                622 *                             3 = BLOCK, TREAT, OR REP STMT            INPARM
                                623 *                             4 = LABEL SPECIFICATIONS                 INPARM
                                624 *                             5 = CODING STATEMENT                     INPARM
                                625 *                             6 = SOURCE SPECIFICATION                 INPARM
                                626 *                             7 = DISTRIB PARAMETERS                   INPARM
                                627 *                             8 = CLASSES SPECIFICATIONS               INPARM
                                628 *                             9 = TESTS CARD                           INPARM
                                629 *                             10 = TRANS OPTION STATEMENT              INPARM
                                630 *                             11 = END OF PARAMETERS OR END            INPARM
                                631 *                             12 = END OF RUN OR END OF JOB            INPARM
                                632 *                             13 = UNKNOWN PARAMETER                   INPARM
                                633 *                             14 = SYNTAX ERROR DETECTED               INPARM
                                634 *                                                                      INPARM
000D54                          635 INUMB    DS    F              INTEGER*4 INUMB     A COUNT              INPARM
                                636 *                             OF THE NUMBER OF PARAMETERS              INPARM
                                637 *                             FOUND, INCLUDING OMITTED ONES            INPARM
000D58                          638 SUBPRM   DS    40D            REAL*8 SUBPRM(40)   AN ARRAY             INPARM
                                639 *                             CONTAINING THE CHARACTER IMAGES          INPARM
                                640 *                             OF EACH PARAMETER FOUND, LEFT            INPARM
                                641 *                             JUSTIFIED, PADDED WITH                   INPARM
                                642 *                             BLANKS.  PARAMETERS LONGER               INPARM
                                643 *                             THAN 8 CHARACTERS ARE COUNTED            INPARM
                                644 *                             AS 2 OR MORE PARAMETERS.                 INPARM
                                645 *                             OMITTED AND MISSING PARAMETERS           INPARM
                                646 *                             ARE LEFT BLANK.                          INPARM
                                647 *                                                                      INPARM
000158                          648 IPCLS    DS    40F            INTEGER*4 IPCLS(40)  AN                  INPARM
                                649 *                             INTEGER ARRAY CONTAINING A               INPARM
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

                                  650 *                                   NUMBER AS AN INDICATOR OF THE   INPARM
                                  651 *                                   TYPE OF THE CORRESPONDING       INPARM
                                  652 *                                   PARAMETER IN SUBPRM.            INPARM
                                  653 *                                   -1 = PARAMETER OMITTED          INPARM
                                  654 *                                   0 = POSITIVE NUMERIC INTEGER    INPARM
                                  655 *                                   1 = CHARACTER STRING            INPARM
                                  656 *                                   2 = NEGATIVE INTEGER OR         INPARM
                                  657 *                                       A REAL NUMBER               INPARM
                                  658 *                                                                   INPARM
000238                            659 VALUE    DS    40F                   REAL*4 VALUE(40)    AN ARRAY    INPARM
                                  660 *                                   CONTAINING THE REAL NUMBER      INPARM
                                  661 *                                   VALUE OF EACH PARAMETER.        INPARM
                                  662 *                                   OMITTED AND CHARACTER PARAMETER INPARM
                                  663 *                                   SPECIFICATIONS ARE ZEROS.       INPARM
000208                            664 IVALUE   DS    40F                   INTEGER*4 IVALUE(40)  AN ARRAY  INPARM
                                  665 *                                   CONTAINING THE INTEGER VALUE    INPARM
                                  666 *                                   OF EACH PARAMETER.  OMITTED     INPARM
                                  667 *                                   AND CHARACTER PARAMETERS ARE    INPARM
                                  668 *                                   REPRESENTED WITH ZEROS.         INPARM
                                  669 *                                                                   INPARM
                                  670 *              ALL NUMERIC PARAMETERS WILL BE REPRESENTED IN BOTH       INPARM
                                  671 *              VALUE AND IVALUE IN THEIR REAL AND INTEGER MODES         INPARM
                                  672 *              RESPECTIVELY.  IVALUE WILL CONTAIN A TRUNCATED           INPARM
                                  673 *              VALUE OF VALUE IF IT WAS NOT A WHOLE NUMBER.             INPARM
                                  674 *                                                                   INPARM
                                  675 *              THE CLASS OF ANY PARAMETER IS DETERMINED BY THE          INPARM
                                  676 *              FOLLOWING RULES:                                         INPARM
                                  677 *              IF AN ERROR OCCURS IN THE NUMERIC CONVERSION OF          INPARM
                                  678 *              THE PARAMETER TO FLOATING POINT, THE PARAMETER IS        INPARM
                                  679 *              CONSIDERED CHARACTER (IPCLS = 1).                        INPARM
                                  680 *              THE NUMBER IS STORED IN BOTH REAL AND INTEGER FORMS      INPARM
                                  681 *              IN VALUE AND IVALUE.                                     INPARM
                                  682 *              IF THE NUMBER IS NEGATIVE OR FRACTIONAL, IT IS           INPARM
                                  683 *              CONSIDERED REAL (IPCLS = 2).                             INPARM
                                  684 *              IF THE NUMBER CAN BE REPRESENTED AS A POSITIVE           INPARM
                                  685 *              INTEGER, THEN IT IS CONSIDERED ONE (IPCLS = 0).          INPARM
                                  686 *              SINCE BOTH FORMS OF A NUMERIC PARAMETER ARE PROVIDED,    INPARM
                                  687 *              THE CONTEXT OF THE PARAMETER CARD CAN BE USED TO         INPARM
                                  688 *              DETERMINE THE ACTUAL TYPE OF A NUMERIC PARAMETER.        INPARM
                                  689 *                                                                   INPARM
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000378                           690 RC       DS    F          INTEGER*4 RC   USED ONLY BY      INPARM
                                 691 *                         THE CALLING PROGRAMS TO HANDLE  INPARM
                                 692 *                         SPECIAL RETRUN CODE INFORMATION INPARM
00037C                           693 LENGPARM EQU   *-PARMS     LENGTH OF /INPARM/              INPARM
                                 694          END                                             DECODE
```

SYMBOL	LEN	VALUE	DEFN	REFERENCES													
ANOTETR	6	000288	238	221													
BLANKS	80	000588	460	52	113	159	162	213	382	403	415						
BLDSLBS	4	000348	333	274	299												
BLKTRTRP	4	000208	202	569	570	571											
BLOCK	8	000058	600	224													
CDTYP	8	000508	461	113	117	137											
CLCBL	6	000438	403	402													
CLCTRP	6	000388	382	381													
CLSS	4	000338	306	576	577												
CNVLCCP	4	000458	412	453													
CODING	4	000338	303	573													
COUNTPRM	4	000388	348	352													
DATA	4	000338	302	568													
DATE	8	0001A8	604	52	77	78	79	106	109								
DECCDE	1	000CCC	31														
DELIMIT	4	000418	391	377													
DIST	4	000338	305	575													
DOUBLE	8	000538	458	63	64	73	75	76	77	83	85	86	90	90	91	104	105
				106	107	108	109	266	267								
ENDBIR	4	000258	224	239													
ENDLBLP	4	000284	256	252													
ENDSLB	4	000448	406	366	375	383											
EOP	4	000342	310	582	583												
EOR	4	000346	312	584	586												
FEBDAY	1	000581	467	81	88												
FINDST	4	000358	336	341													
FINISH	4	0001CA	179	209	296	300	310	312	314								
FPRC	1	000CCC	489	437	441												
FPR2	1	000CC2	490														
FPR4	1	000CC4	491														
FPR6	1	000CC6	492														
GENERAL	4	000338	299	302	303	304	305	306	307	308							
GOTDATE	6	000CF2	113	53	234												
GOTPCN	2	000CC4	103	98													
GOTPARM	6	000158	144	138													
GOTPSTRT	4	000378	344	337	339												
INTALP	6	0004BC	444														
INUMB	4	000C54	635	146	146	173	283	354	386	388	410						
IPARM	4	000C5C	619	145	208												
IPCLS	4	000198	648	150	151	151	174	357	397	407							

SYMBOL	LEN	VALUE	DEFN	REFERENCES												
IVALUE	4	0002C8	664	153	153	409										
LABEL	8	0000C8	603	281												
LBLLP	4	0002A4	251	254												
LBLPR	6	0002D4	266	265												
LBLS	4	0002BA	243	572												
LENGFARM	1	000378	693													
LENGPRTY	1	000100	608													
LINE	4	0001BC	607													
LITERALS	1	000668	550													
LKLFTLE	4	0001BC	165	168												
LOOKFARM	4	000138	132	121	125	127	129									
MONLP	4	0000BE	96	101												
MONTBL	1	0005FC	466	94	467											
MVCHTR	6	000236	217	222												
MVCSLB	6	0003FC	371	378												
MVCTTLE	6	000200	198	197												
MVLBLLP	6	000324	291	294												
NEXTCNV	4	000400	449	414	416	425	445									
NEXTCNE	4	000424	395	365	389	392										
NOFIND	4	000348	314	587												
NOTIME	4	0000E8	73	61												
NOTNLLL	6	000304	368	363												
OKEQ	4	00021E	211	204												
PAGE	4	000188	606													
PARM	8	000000	618	116	185	215	233	238	244	246	248	266	334	359		
PARMPTR	4	0005E4	463	132	160	202										
PARMS	1	000000	617	50	693											
PARMSTRT	4	0005EC	465	344												
PATCH	100	0005FE	497													
PMOVE	6	000100	119	130												
PRTY	1	000000	598	49	608											
RC	4	000378	690													
REALNUM	6	000406	447	432	435	442										
REPLI	8	0000C0	602	230												
R00	1	000000	472	55	55	60	60	63	380	401						
R01	1	000001	473	73	74	75	91	97	103	104	115	120	155	156	160	161	161
				165	167	167	192	198	334	336	338	340	340	344	347	358	369
				371	373	373											
R02	1	000002	474	84	85	93	99	99	107	116	119	122	122	124	126	128	132
				163	168	185	186	188	192	194	197	202	203	211	211	217	218

SYMBCL	LEN	VALUE	DEFN	REFERENCES													
				218	220	224	227	230	232	238	243	267	268	268	270	273	279
				280	281	281	291	292	292	335	341	346	346	352	370	378	412
				413	413	433	434	434									
R03	1	0C0CC3	475	94	96	100	100	117	119	123	123	133	137	139	144	155	193
				194	195	196	196	198	212	222	248	249	258	260	282	291	293
				293	345	350	350	354	379	380	381	386	387	387	388	400	401
				402	406	415	420	449	449								
R04	1	CC0CC4	476	95	95	96	97	103	134	139	214	217	219	219	249	251	253
				253	256	258	260	261	263	263	265	283	284	284	286	294	347
				348	351	351	359	360	365	374	379	392	400	407	412	444	444
				447	450	450											
R05	1	CC0CC5	477	135	286	287	288	358	360	362	364	364	371	372	372	376	382
				391	391	403	408	419	428	441	451	451					
R06	1	0C0CC6	478	357	368	396	396	398	409	427	433	438	452	452			
R07	1	CC0CC7	479	356	369	395	395	410	453								
R08	1	CC0CCE	480	274	299	455											
R09	1	CC0CC9	481	397	398												
R10	1	CC0CCA	482	48	49												
R11	1	CC0CCB	483	48	50												
R12	1	CC0CCC	484	36	37												
R13	1	CC0CCD	485	46													
R14	1	CC0CCE	486														
R15	1	0C0CCF	487	424	424	431	431										
SAVEAREA	8	0C04F8	457	40	46												
SCANFARM	6	CC014C	137	139													
SRCE	4	CC033A	304	574													
SUBLF	4	C003BC	362	404													
SUBPRM	8	CC0C58	638	147	148	148	149	149	282	356	406						
SUBSCRPT	4	CC05E8	464	273	279	287											
SUBSET	4	CC07EC	273	245													
SUBSW8	1	CC05FC	469	275	333	385											
SYNTAX	4	CC021C	208	247	257	259	262	269	271	276	285	289	342	393	399		
TEST	4	CC033A	307	578	579												
THRLETR	4	CC077C	236														
THRLLELS	4	CC0336	296														
THRLTTL	4	CC01CC	173	199	236												
TIME	8	CC01BC	605	66	67	68	69	70									
TITLE	8	CC0CCC	599	159	196												
TRAN	4	CC033A	308	580	581												
TREAT	8	CC0C58	601	227													

14 SEP 70

SYMBOL	LEN	VALUE	DEFN	REFERENCES												
TTLAST	2	000EC	192	187												
TTLE	6	000E8	159	567												
TTLFRST	4	000DC	185	166	188											
TYPLAST	1	00075	585	135												
TYPNUMB	4	0005FC	462	114	114	144	145									
TYPTEL	12	000684	560	133	134											
VALUE	4	000238	659	152	152	408										
WORK	80	000538	459	64	66	68	70	82	82	83	84	162	213	214	215	225	228
				232	233											

POS.ID	REL.ID	FLAGS	ADDRESS
C1	C1	CE	0C068D
C1	01	C8	000699
C1	01	C8	0C06A5
C1	C1	C8	0006B1
C1	C1	CE	0C06BD
C1	01	C8	0006C9
C1	C1	C8	0006D5
C1	C1	C8	0006E1
C1	01	CE	0006ED
C1	01	C8	0006F9
C1	C1	C8	0C0705
C1	C1	C8	0C0711
C1	C1	CE	000710
C1	01	C8	000729
C1	C1	C8	000735
C1	C1	CE	000741
C1	01	C8	00074D
C1	01	C8	000759
C1	C1	C8	000765
C1	C1	C8	0C0771
C1	C2	1C	000668
C1	03	1C	00066C
C1	C4	1C	000680

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

SYMBOL	TYPE	ID	ADDR	LENGTH	LD ID
ECDAT	SD	01	000000	000108	
ERRSAV	ER	02			
ERRSTR	ER	03			
FLAG	ER	04			
IHCUATBL	ER	05			
UNITS	ER	06			

```
EODD   PROCESS CONVERSION ERRORS FOR END OF DATA CARD                                                    PAGE    1

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  14 SEP 70

                                  2 ************************************************************************ EODAD
                                  3 *                                                                      * EODAD
                                  4 *              THIS ROUTINE RECEIVES CONTROL FROM THE FORTRAN DATA     * EODAD
                                  5 *        CONVERSION ROUTINE WHEN AN ERROR IS DETECTED.  THE CALL IS    * EODAD
                                  6 *        OF THE FORMAT:                                                * EODAD
                                  7 *              CALL EODAD (IRETCD,IERNO,BADCHAR)                       * EODAD
                                  8 *              IRETCD IS AN INTEGER*4 VARIABLE SET TO INDICATE         * EODAD
                                  9 *        SYSTEM (FORTRAN) CORRECTION OF THE ERROR OR USER (THIS        * EODAD
                                 10 *        ROUTINE) CORRECTION.                                          * EODAD
                                 11 *              IERNO IS THE ERROR NUMBER. THIS SHOULD BE 215.          * EODAD
                                 12 *              BADCHAR IS THE ACTUAL CHARACTER THAT CAUSED THE ERROR.  * EODAD
                                 13 *                                                                      * EODAD
                                 14 *              THIS ROUTINE DETERMINES WHETHER OR NOT THE ERROR WAS    * EODAD
                                 15 *        CAUSED BY AN ACTUAL ERROR IN THE DATA, OR BY ATTEMPTED        * EODAD
                                 16 *        PROCESSING OF THE 'END OF DATA' CARD.  THIS INFORMATION IS    * EODAD
                                 17 *        COMMUNICATED TO THE USER IN THE COMMON BLOCK /FLAG/.          * EODAD
                                 18 *        THE UNIT NUMBER ON WHICH THE ERROR IS ASSUMED TO HAVE         * EODAD
                                 19 *        OCCURRED IS CONTAINED IN THE FIELD, 'IPT', IN THE COMMON      * EODAD
                                 20 *        BLOCK /UNITS/.                                                * EODAD
                                 21 *                                                                      * EODAD
                                 22 *              THIS ROUTINE IS SERIALLY REUSABLE, BUT IS NOT           * EODAD
                                 23 *        REENTRANT.                                                    * EODAD
                                 24 *                                                                      * EODAD
                                 25 ************************************************************************ EODAD
```

```
EODC   PROCESS CONVERSION ERRORS FOR END OF DATA CARD                                                    PAGE    2

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  14 SEP 70

                                    27 EODAD    PRIME BR=R12,SA=SAVEAREA       ENTER AND SET UP LINKAGE        EODAD
000000                              28+EODAD    START 0
000000 47F0 F00A          0000A     29+         B     10(0,15) BRANCH AROUND ID
000004 C5                           30+         DC    AL1(5)
000005 C5D6C4C1C4                   31+         DC    CL5'EODAD' IDENTIFIER
00000A 90EC D00C          0000C     32+         STM   14,12,12(13) SAVE REGISTERS
00000E C5C0                         33+         BALR  R12,0 CREATE BASE REGISTER
000010                              34+         USING *,R12
000010 1830                         35+         LR    3,0 SAVE -0-
000012 1821                         36+         LR    2,1 SAVE PARAMETER REGISTER
000014 4110 C120          00130     37+         LA    1,SAVEAREA GET ADDRESS OF SAVE AREA
000018 5010 0008          00008     38+         ST    1,8(13) STORE FORWARD POINTER
00001C 50D1 C004          00004     39+         ST    13,4(1) STORE BACKWARD POINTER
000020 18D1                         40+         LR    13,1 SET UP REG FOR SAVE AREA
000022 1812                         41+         LR    1,2 RESTORE PARAMETER REGISTER
000024 18C3                         42+         LR    0,3 RESTORE -0-
                                    43 *                                                                    EODAD
00013C                              44          USING SAVEAREA,R13             USE THE SAVE AREA AS ANOTHER    EODAD
000026 58B0 D058          00188     45          L     R11,=V(FLAG)             LOAD COMMON BLOCK ADDRESS       EODAD
000000                              46          USING FLAGS,R11                ESTABLISH DSECT ADDRESSIBILITY  EODAD
                                    47 *                                                                    EODAD
00002A 9824 1000          00000     48          LM    R02,R04,0(R01)           LOAD THE PARAMETER LIST         EODAD
                                    49 *              R02 IS THE ADDRESS OF IRETCD                          EODAD
                                    50 *              R03 IS THE ADDRESS OF IERNO (215)                     EODAD
                                    51 *              R04 IS THE ADDRESS OF BADCHAR                         EODAD
                                    52 *                                                                    EODAD
00002E 5850 D05C          0018C     53          L     R05,=V(IHCUATBL)         GET FORTRANS UNIT TABLE ADDRESS EODAD
000000                              54          USING UATBL,R05                SET UP ADDRESSIBILITY           EODAD
000032 5860 D060          00190     55          L     R06,=V(UNITS)            GET PROGRAM UNIT TABLE ADDRESS  EODAD
000000                              56          USING UNITNOS,R06              ESTABLISH ADDRESSIBILITY        EODAD
000036 5860 601C          0001C     57          L     R06,IPT                  LOAD THE UNIT NUMBER            EODAD
                                    58          DROP  R06                      DROP THE REGISTER               EODAD
                                    59 *                                                                    EODAD
00003A C660                         60          BCTR  R06,R00                  SUBTRACT 1 FROM THE UNIT NUMBER EODAD
00003C 8960 0004          00004     61          SLL   R06,4                    MULTIPLY RESULT BY 16           EODAD
00004C 1A56                         62          AR    R05,R06                  ALTER UNIT TABLE ADDRESS        EODAD
                                    63 *                                                                    EODAD
000042 5860 5008          00008     64          L     R06,UNITBLK              LOAD ADDRESS OF UNIT BLOCK      EODAD
000000                              65          USING UBLOCK,R06               SET UP ADDRESSIBILITY           EODAD
000046 587C 6004          00004     66          L     R07,BUFF1                LOAD ADDRESS OF BUFFER 1        EODAD
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    EODAD
00004A 588C 60C8             00008    67           L     R08,BUFF2          LOAD ADDRESS OF BUFFER 2           EODAD
                                      68           DROP  RC5,RC6            DROP NOW UNNECESSARY REGS          EODAD
00004E D24F B008 7000 C0008  00000    69           MVC   BUFFER(80),0(R07)  PUT BUFFERS INTO COMMON BLOCK      EODAD
000054 D24F B058 8000 C0058  00000    70           MVC   BUFFER+80(80),0(R08)                                  EODAD
00005A D703 8000 8000 00000  00000    71           XC    EOD,EOD            ZERO INDICATORS                    EODAD
000060 D703 8004 8004 00004  00004    72           XC    CDCOL,CDCOL                                           EODAD
000066 D703 80A8 80A8 000A8  000A8    73           XC    BUFNO,BUFNO                                           EODAD
                                      74 *                                                                     EODAD
00006C 4177 0000             00000    75           LA    R07,0(R07)         CLEAR POSSIBLE NON-ZERO            EODAD
000070 4188 0000             00000    76           LA    R08,0(R08)         HI-ORDER BYTES                     EODAD
                                      77 *                                                                     EODAD
000074 4197 004F             0004F    78           LA    R09,79(R07)        BUFFER 1 COLUMN 80                 EODAD
                                      79 *                                                                     EODAD
000078 4144 0000             00000    80           LA    R04,0(R04)         CLEAR END OF PARM INDICATOR        EODAD
00007C 1947                           81           CR    R04,R07            CHECK AGAINST BUFFER 1, COL1       EODAD
00007E 474C 0080             00090    82           BL    ISBUFF2            IF LOW, MUST BE BUFFER 2           EODAD
000082 1949                           83           CR    R04,R09            CHECK AGAINST BUFFER 1, COL 80     EODAD
000084 472C 0080             00090    84           BH    ISBUFF2            IF LARGE, IS BUFFER 2              EODAD
000088 4180 0001             00001    85           LA    R08,1              SET BUFFER 1 NUMBER                EODAD
00008C 47FC 0096             00096    86           B     DOCHECK                                               EODAD
                                      87 *                                                                     EODAD
000090 1878                           88 ISBUFF2   LR    R07,R08            MOVE THIS BUFFER ADDRESS INTO 1    EODAD
000092 418C 0002             00002    89           LA    R08,2              SET BUFFER NUMBER                  EODAD
                                      90 *                                                                     EODAD
000096 508C 80A8             000A8    91 DOCHECK   ST    R08,BUFNO          STORE THE BUFFER NUMBER            EODAD
00009A 1847                           92           SR    R04,R07            CALCULATE THE CARD COLUMN          EODAD
00009C 4144 0001             00001    93           LA    R04,1(R04)                                            EODAD
0000A0 5040 8004             00004    94           ST    R04,CDCOL          STORE THE COLUMN IN ERROR          EODAD
                                      95 *                                                                     EODAD
0000A4 D50B 7000 0064 00000  00194    96           CLC   0(12,R07),=C'END OF DATA '    CHECK FOR END CARD      EODAD
0000AA 4780 00C0             000C0    97           BE    ECDCK              BRANCH IF THAT CAUSED THE ERROR    EODAD
0000AE D504 7000 0070 00000  001A0    98           CLC   0(5,R07),=C'TITLE'   CHECK FOR THE NEXT TITLE CARD      EODAD
0000B4 4770 00B0             000B0    99           BNE   CONVERR            ERROR IF NOT                       EODAD
0000B8 418C 0003             00003   100           LA    R08,3              SET EOD SIGNAL FOR A TITLE CARD    EODAD
0000BC 47F0 00C4             000C4   101           B     EODSTORE                                              EODAD
                                     102 *                                                                     EODAD
0000C0 1788                          103 CONVERR  XR    R08,R08            ZERO REGISTER                      EODAD
0000C2 5082 0000             00000   104           ST    R08,0(R02)         STORE TO INDICATE SYSTEM FIX       EODAD
0000C6 C680                          105           BCTR  R08,R00            MAKE IT A MINUS 1                  EODAD
0000C8 508C 80C0             000C0   106           ST    R08,EOD            SET THE ERROR FLAG                 EODAD
```

```
EODD   PROCESS CONVERSION ERRORS FOR END OF DATA CARD                                                    PAGE    4

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                14 SEP 70

000000 47F0 C10A             0011A   107          B     RETURN                  GO TO RETURN TO FORTRAN          EODAD
                                     108 *                                                                   EODAD
000000 4180 0001             00001   109 EODCK    LA    R08,1                   'END OF DATA' CARD               EODAD
000004 5080 B000             00000   110 EODSTORE ST    R08,EOD                 SET END OF DATA FLAG             EODAD
000008 4180 0001             00001   111          LA    R08,1                                                    EODAD
000000 5082 0000             00000   112          ST    R08,0(R02)              SET TO INDICATE USER FIX (NONE)  EODAD
                                     113 *                                                                   EODAD
000000 9240 7000       00000         114          MVI   0(R07),C' '             BLANK OUT BUFFER                 EODAD
000004 D24E 7001 7000  00001 00000   115          MVC   1(79,R07),0(R07)                                         EODAD
                                     116 *                                                                   EODAD
000004 5030 D048             00178   117          ST    R03,CALLPARM            STORE ADDRESS OF ERROR NUMBER    EODAD
000004 4110 D048             00178   118          LA    R01,CALLPARM            LOAD PARAMETER ADDRESS FOR CALL  EODAD
                                     119          CALL  ERRSAV                  GO GET ERROR TABLE ENTRY         EODAD
000002 0700                          120+         CNOP  0,4
000004 47F0 00EC             000FC   121+         B     *+8 BRANCH AROUND VCON
000008 00000000                      122+IHB0003B DC    V(ERRSAV) ENTRY POINT ADDRESS
000000 58F0 00E8             000F8   123+         L     15,IHB0003B LOAD 15 WITH ENTRY ADR
000100 05EF                          124+         BALR  14,15 BRANCH TO ENTRY POINT
                                     125 *                                                                   EODAD
000102 4380 D052             00182   126          IC    R08,ERRCOUNT            GET NUMBER OF ERRORS             EODAD
000106 0680                          127          BCTR  R08,R00                 SUBTRACT 1 FOR THIS ERROR        EODAD
000108 4280 D052             00182   128          STC   R08,ERRCOUNT            STORE IT BACK IN THE TABLE       EODAD
                                     129 *                                                                   EODAD
                                     130          CALL  ERRSTR                  RESTORE THE MODIFIED ENTRY       EODAD
000100                               131+         CNOP  0,4
000100 47F0 C104             00114   132+         B     *+8 BRANCH AROUND VCON
000110 00000000                      133+IHB0004B DC    V(ERRSTR) ENTRY POINT ADDRESS
000114 58F0 C100             00110   134+         L     15,IHB0004B LOAD 15 WITH ENTRY ADR
000118 05EF                          135+         BALR  14,15 BRANCH TO ENTRY POINT
                                     136 *                                                                   EODAD
                                     137 RETURN   TERME RC=0                    RETURN TO FORTRAN                EODAD
00011A 58DD 0004             00004   138+RETURN   L     13,4(13) RESTORE REGISTER -13-
00011E 98EC D00C             0000C   139+         LM    14,12,12(13) RESTORE THE REGISTERS
000122 92FF D00C       00000         140+         MVI   12(13),X'FF' SET RETURN INDICATION
000126 41F0 0000             00000   141+         LA    15,0(0,0) LOAD RETURN CODE
00012A 07FE                          142+         BR    14 RETURN
                                     143 *                                                                   EODAD
```

```
EODD   PROCESS CONVERSION ERRORS FOR END OF DATA CARD                                                    PAGE    5

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                14 SEP 70

00012C C0C0C0C0
00013C C0C0C0C0C0000C00                 145 SAVEAREA DC    9D'C'                    OS SAVEAREA                    EODAD
00017B C0C0C0C0B0C0C1B0                 146 CALLPARM DC    A(0),AL1(128),AL3(TABENT)     ERRSAV/ERRSTR PARAMETERS  EODAD
00018C                                  147 TABENT   DS    0D                       ERROR TABLE ENTRY              EODAD
00018D                                  148 ERRMAX   DS    AL1                      MAXIMUM NUMBER OF ERRORS       EODAD
00018E                                  149 ERRMESG  DS    AL1                      NUMBER OF MESSAGES TO PRINT    EODAD
00018E                                  150 ERRCOUNT DS    AL1                      NUMBER OF ERRORS OCCURRED      EODAD
00018E                                  151 ERRCPTN  DS    XL1                      ERROR OPTION BYTES             EODAD
00018E                                  152 ERRCUTNE DS    A                        ADDRESS OF THIS ROUTINE        EODAD
                                        153 *                                                                      EODAD
                                        154          REGS  STD                      REGISTER NAME EQUATES          EODAD
000000                                  155+R00      EQU   0
000001                                  156+R01      EQU   1
000002                                  157+R02      EQU   2
000003                                  158+R03      EQU   3
000004                                  159+R04      EQU   4
000005                                  160+R05      EQU   5
000006                                  161+R06      EQU   6
000007                                  162+R07      EQU   7
000008                                  163+R08      EQU   8
000009                                  164+R09      EQU   9
00000A                                  165+R10      EQU   10
00000B                                  166+R11      EQU   11
00000C                                  167+R12      EQU   12
00000D                                  168+R13      EQU   13
00000E                                  169+R14      EQU   14
00000F                                  170+R15      EQU   15
                                        171 *                                                                      EODAD
00018B                                  172 LITERALS LTORG                          LITERALS USED                  EODAD
00018B C0C0C0C0                         173                  =V(FLAG)
00018C C0C0C0C0                         174                  =V(IHCUATBL)
00019C C0C0C0C0                         175                  =V(UNITS)
000194 C5D5C44CD6C64CC4                 176                  =C'END OF DATA '
0001AC E3C9E3D3C5                       177                  =C'TITLE'
                                        178 PATCH    PATCH 50                       PROGRAM PATCH AREA             EODAD
0001A5 CC
0001A6                                  179+         DC    0H'0'
0001A6                                  180+PATCH    DS    0CL50
0001A6 DC76                             181+         DC    S(*)
0001A8 DC78                             182+         DC    S(*)
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

0001AA CC7A                        183+          DC    S(*)
0001AC CC7C                        184+          DC    S(*)
0001AE CC7E                        185+          DC    S(*)
0001B0 CC80                        186+          DC    S(*)
0001B2 CC82                        187+          DC    S(*)
0001B4 CC84                        188+          DC    S(*)
0001B6 CC86                        189+          DC    S(*)
0001B8 CC88                        190+          DC    S(*)
0001BA CC8A                        191+          DC    S(*)
0001BC CC8C                        192+          DC    S(*)
0001BE CC8E                        193+          DC    S(*)
0001C0 CC9C                        194+          DC    S(*)
0001C2 CC92                        195+          DC    S(*)
0001C4 CC94                        196+          DC    S(*)
0001C6 CC96                        197+          DC    S(*)
0001C8 CC98                        198+          DC    S(*)
0001CA CC9A                        199+          DC    S(*)
0001CC CC9C                        200+          DC    S(*)
0001CE CC9E                        201+          DC    S(*)
0001D0 CCA0                        202+          DC    S(*)
0001D2 CCA2                        203+          DC    S(*)
0001D4 CCA4                        204+          DC    S(*)
0001D6 CCA6                        205+          DC    S(*)
                                   206 *
```

EODAD

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

                                   208 *                                                                       EODAD
                                   209 *              DUMMY SECTION FOR THE COMMON BLOCK /FLAG/, USED FOR      EODAD
                                   210 *              COMMUNICATIONS OF ERRORS TO THE USER.  THE FIELDS IN     EODAD
                                   211 *              THIS COMMON BLOCK CAN BE INTERROGATED TO DETERMINE       EODAD
                                   212 *              IF AN ERROR OCCURRED OR IF THE 'END OF DATA' WAS READ.   EODAD
                                   213 *                                                                       EODAD
000000                             214 FLAGS    DSECT                          DSECT FOR COMMON BLOCK /FLAG/   EODAD
000000                             215 EOD      DS    F                        INTEGER*4 EOD - WILL CONTAIN    EODAD
                                   216 *                                       0 IF READ WAS OK - THIS ROUTINE EODAD
                                   217 *                                            WON'T EVEN BE ENTERED.     EODAD
                                   218 *                                       1 IF THE 'END OF DATA' WAS READ EODAD
                                   219 *                                       2 IF PHYSICAL END OF FILE IS    EODAD
                                   220 *                                            REACHED -'THIS ROUTINE     EODAD
                                   221 *                                            WILL NOT BE ENTERED.       EODAD
                                   222 *                                       3 IF A 'TITLE' STMT WAS READ    EODAD
                                   223 *                                       -1 IF A CONVERSION ERROR        EODAD
000004                             224 CDCOL    DS    F                        INTEGER*4 CDCOL - COLUMN OF     EODAD
                                   225 *                                            CARD THAT IS IN ERROR      EODAD
000008                             226 BUFFER   DS    20D                     REAL*8 BUFFER((10,2) THE BUFFERS EODAD
0000A8                             227 BUFNO    DS    F                        INTEGER*4 BUFNO  WHICH BUFFER   EODAD
0000AC                             228 LENGFLAG EQU   *-FLAGS                  LENGTH OF THIS DSECT            EODAD
                                   229 *                                                                       EODAD
                                   230 *              DUMMY SECTION FOR THE COMMON BLOCK /UNITS/, USED TO      EODAD
                                   231 *              HOLD THE UNIT NUMBERS FOR FORTRAN I/O                    EODAD
                                   232 *                                                                       EODAD
000000                             233 UNITNOS  DSECT                          DSECT FOR COMMON BLOCK /UNITS/  EODAD
000000                             234 RDR      DS    F                        INTEGER*4 RDR  PARAMETER INPUT  EODAD
000004                             235 PCH      DS    F                        INTEGER*4 PCH  PUNCHED OUTPUT   EODAD
000008                             236 LST      DS    F                        INTEGER*4 LST  PRINTED OUTPUT   EODAD
000000                             237 UT1      DS    F                        INTEGER*4 UT1  WORK UNIT        EODAD
000010                             238 UT2      DS    F                        INTEGER*4 UT2  WORK UNIT        EODAD
000014                             239 UT3      DS    F                        INTEGER*4 UT3  WORK UNIT        EODAD
000018                             240 UT4      DS    F                        INTEGER*4 UT4  WORK UNIT        EODAD
00001C                             241 IPT      DS    F                        INTEGER*4 IPT  DATA INPUT UNIT  EODAD
000020                             242 OUT      DS    F                        INTEGER*4 OUT  DATA OUTPUT UNIT EODAD
000024                             243 LENGUNIT EQU   *-UNITNOS                LENGTH OF THIS DSECT            EODAD
                                   244 *                                                                       EODAD
                                   245 *              DUMMY SECTION FOR THE FORTRAN UNIT ADDRESS TABLE,        EODAD
                                   246 *              IHCUATBL.  IT IS USED IN THIS PROGRAM TO RETRIEVE        EODAD
                                   247 *              THE UNIT BLOCK ADDRESS                                   EODAD
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT			14 SEP 70
				248	*			EDDAD
000000				249	UATBL	DSECT		DSECT FOR IHCUATBL EDDAD
000000				250	UNITNO	DS	H	LAST UNIT NUMBER USED EDDAD
000002				251	UATSIZE	DS	H	SIZE OF THIS TABLE EDDAD
000004				252	UATUNITS	DS	F	NUMBER OF UNITS EDDAD
000008				253	UNITNTRY	DS	0XL16	SINGLE UNIT NUMBER ENTRY EDDAD
000008				254	UNITBLK	DS	A	ADDRESS OF UNIT BLOCK EDDAD
00000C				255	UNITDFLT	DS	XL8	DEFAULTS FOR THIS UNIT EDDAD
000014				256	UNITLIST	DS	F	DIRECT ACCESS UNITS DATA ONLY EDDAD
				257	*			EDDAD
				258	*			DSECT FOR THE FORTRAN UNIT BLOCK WHICH CONTAINS THE EDDAD
				259	*			ADDRESSES OF THE BUFFERS OF THE UNIT IN WHICH EDDAD
				260	*			THE ERROR OCCURRED. EDDAD
				261	*			EDDAD
000000				262	UBLOCK	DSECT		DSECT FOR THE UNIT BLOCK EDDAD
000000				263	ABYTE	DS	XL1	SEE A FORTRAN G OR H PLM EDDAD
000001				264	BBYTE	DS	XL1	FOR THE EXACT MEANINGS OF EDDAD
000002				265	CBYTE	DS	XL1	THESE FIELDS EDDAD
000003				266	DBYTE	DS	XL1	EDDAD
000004				267	BUFF1	DS	A	ADDRESS OF BUFFER 1 EDDAD
000008				268	BUFF2	DS	A	ADDRESS OF BUFFER 2 EDDAD
00000C				269	CUREUFF	DS	A	EDDAD
000010				270	RECPTR	DS	A	EDDAD
000014				271	LASTDECB	DS	A	EDDAD
000018				272	SWMASK	DS	F	EDDAD
00001C				273	SKLDECB1	DS	CL20	EDDAD
000030				274	LIVECNT1	DS	F	EDDAD
000034				275	SKLDECB2	DS	CL20	EDDAD
000048				276	LIVECNT2	DS	F	EDDAD
00004C				277	SKELDCB	DS	CL88	EDDAD
0000A4				278	LENGUBLK	EQU	*-UBLOCK	LENGTH OF THE UNIT BLOCK EDDAD
				279	*			EDDAD
				280		END		EDDAD

SYMBOL	LEN	VALUE	DEFN	REFERENCES			
ABYTE	1	000000	263				
BBYTE	1	000001	264				
BUFFER	8	000008	226	69	70		
BUFF1	4	000004	267	66			
BUFF2	4	000008	268	67			
BUFNO	4	0000A8	227	73	73	91	
CALLFARM	4	000178	146	117	118		
CBYTE	1	000002	265				
CODEL	4	0000C4	224	72	72	94	
CONVERR	2	000000	103	99			
CORELFF	4	000000	269				
DBYTE	1	000003	266				
DDCHECK	4	000096	91	86			
ECD	4	000000	215	71	71	106	110
ECDAD	1	000000	28				
ECDCK	4	00000C	109	97			
EDDSTORE	4	0000D4	110	101			
ERRCOUNT	1	000182	150	126	128		
ERRMAX	1	000180	148				
ERRMSG	1	000181	149				
ERRRTN	1	000183	151				
ERRRTNE	4	000184	152				
FLAGS	1	000000	214	46	228		
IHB000B8	4	0000F8	122	123			
IHB000B8	4	00011C	133	134			
IPT	4	00001C	241	57			
ISBUFF2	2	000090	88	82	84		
LASTLECB	4	000014	271				
LENGFLAG	1	0000AC	228				
LENGLBLK	1	0000A4	278				
LENGUNIT	1	000024	243				
LITERALS	1	000188	172				
LIVECNT1	4	00003C	274				
LIVECNT2	4	000048	276				
LST	4	000008	236				
OUT	4	00002C	242				
PATCH	50	0001A6	180				
PCH	4	000004	235				
RDR	4	000000	234				
RECPTR	4	00001C	270				

SYMBOL	LEN	VALUE	DEFN	REFERENCES												
RETURN	4	000118	138	107												
R00	1	000000	155	60	105	127										
R01	1	000001	156	48	118											
R02	1	000002	157	48	104	112										
R03	1	000003	158	117												
R04	1	000004	159	48	80	80	81	83	92	93	93	94				
R05	1	000005	160	53	54	62	68									
R06	1	000006	161	55	56	57	58	60	61	62	64	65	68			
R07	1	000007	162	66	69	75	75	78	81	88	92	96	98	114	115	115
R08	1	000008	163	67	70	76	76	85	88	89	91	100	103	103	104	105	106
				109	110	111	112	126	127	128						
R09	1	000009	164	78	83											
R10	1	00000A	165													
R11	1	00000B	166	45	46											
R12	1	00000C	167	33	34											
R13	1	00000D	168	44												
R14	1	00000E	169													
R15	1	00000F	170													
SAVEAREA	8	000130	145	37	44											
SKELECB	88	000040	277													
SKLEECB1	20	000010	273													
SKLEECB2	20	000034	275													
SWMASK	4	000018	272													
TABENT	8	000180	147	146												
UATBL	1	000000	249	54												
UATSIZE	2	000002	251													
UATUNITS	4	000004	252													
UBLOCK	1	000000	262	65	278											
UNITBLK	4	000008	254	64												
UNITDFLT	8	000000	255													
UNITLIST	4	000014	256													
UNITNC	2	000000	250													
UNITNCS	1	000000	233	56	243											
UNITNTRY	16	00000E	253													
UT1	4	000000	237													
UT2	4	000010	238													
UT3	4	000014	239													
UT4	4	000018	240													

POS.ID	REL.ID	FLAGS	ADDRESS
C1	C1	C8	00017C
C1	02	1C	0000F8
C1	C3	1C	000110
C1	C4	1C	000188
01	05	1C	00018C
C1	C6	1C	000190

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

```
SYMBOL   TYPE ID  ADDR  LENGTH LD ID
IHBFFFIC  SD  01 000000 000BDE
```

```
FPRT   FLOATING POINT ROUTINES - FIX, FLOAT, INPUT, OUTPUT                                                    PAGE    1

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

000000                                 2 IHBFFFIO CSECT                                                          00000200
000000                                 3          USING *,R15                                                    00000300
000000 47F0 F038            00038      4 FIX1     B     FIXS          0        FIX - SINGLE PRECISION          00000400
000004 47F0 F03C            0003C      5 FIX2     B     FIXD          4        FIX - DOUBLE PRECISION          00000500
000008 47F0 F122            00122      6 FLOAT1   B     FLOATS        8        FLOAT - SINGLE PRECISION        00000600
000000 47F0 F126            00126      7 FLOAT2   B     FLOATD       12        FLOAT - DOUBLE PRECISION        00000700
000010 47F0 F17C            0017C      8 FPIN1    B     IPTSF        16        INPUT - SINGLE PRECISION FW.D   00000800
000014 47F0 F180            00180      9 FPIN2    B     IPTDF        20        INPUT - DOUBLE PRECISION FW.D   00000900
000018 47F0 F184            00184     10 FPIN3    B     IPTSE        24        INPUT - SINGLE PRECISION FW.D   00001000
000010 47F0 F188            00188     11 FPIN4    B     IPTDE        28        INPUT - DOUBLE PRECISION EW.D   00001100
000020 47F0 F47C            0047C     12 FPOLT1   B     OUTSF        32        OUTPUT - SINGLE PRECISION FW.D  00001200
000024 47F0 F480            00480     13 FPOLT2   B     OUTDF        36        OUTPUT - DOUBLE PRECISION FW.D  00001300
000028 47F0 F484            00484     14 FPOLT3   B     OUTSE        40        OUTPUT - SINGLE PRECISION EW.D  00001400
000020 47F0 F488            00488     15 FPOLT4   B     OUTDE        44        OUTPUT - DOUBLE PRECISION EW.D  00001500
                                      16 *                                                                      00001600
000030 C9C8C2C6C6C6C9D6               17          DC    C'IHBFFFIO'           MODULE NAME                     00001700
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

000038 968C F864       C0864          19 FIXS     CI    FPIND,SINGLE                                                00001900
00003C 5020 F828              00828   20 FIXC     ST    R02,SAVER+8          SAVE LINK REGISTER                     00002000
000040 4120 F06E              0008E   21          LA    R02,FIXRTN           SET ENTRY EXIT                         00002100
                                      22 *                                                                            00002200
000044 9001 F820              00820   23 ENTRY    STM   R00,R01,SAVER        SAVE REGS 0 AND 1                      00002300
000048 903F F82C              0082C   24          STM   R03,R15,SAVER+12     STORE 3 THRU 15                        00002400
00004C 6000 F7D8              007D8   25          STD   FPR0,FPSAVE          STORE FLOATING POINT REGS              00002500
000050 6020 F7E0              007E0   26          STD   FPR2,FPSAVE+8                                                00002600
000054 6040 F7E8              007E8   27          STD   FPR4,FPSAVE+16                                               00002700
000058 6060 F7F0              007F0   28          STD   FPR6,FPSAVE+24                                               00002800
00005C 4130 000D              0000D   29          LA    R03,R13              LOAD MAXIMUM VALID REGISTER            00002900
000060 1903                           30          CR    R00,R03              CHECK TO SEE IF REGISTERS              00003000
000062 4720 F06A              0006A   31          BH    *+8                  CONTAIN REGISTER NUMBERS,              00003100
000066 9640 F864       C0864          32          CI    FPIND,R00REG         RATHER THAN ADDRESSES                  00003200
00006A 1913                           33          CR    R01,R03                                                     00003300
00006C 4720 F074              00074   34          BH    *+8                                                         00003400
000070 9620 F864       C0864          35          CI    FPIND,R01REG                                                00003500
000074 0430                           36          SPM   R03                  FOR MASKING INTERRUPTS                 00003600
000076 2800                           37          SDR   FPR0,FPR0            ZERO OUT THE FP REGS                   00003700
000078 2820                           38          LDR   FPR2,FPR0                                                    00003800
00007A 2840                           39          LDR   FPR4,FPR0                                                    00003900
00007C 2860                           40          LDR   FPR6,FPR0                                                    00004000
00007E 9200 F11D       0011D          41          MVI   RETCODE,0            RESET THE RETURN CODE                  00004100
000082 9240 F868       C0868          42          MVI   CHAR,C' '            BLANK THE I/O WORK AREA                00004200
000086 D21E F869 F868 C0869 C0868     43          MVC   CHAR+1(31),CHAR                                              00004300
00008C 07F2                           44          BR    R02                  RETURN TO PARTICULAR ROUTINE           00004400
                                      45 *                                                                            00004500
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT				14 SEP 70
0000EE	912C F864	C0864		47	FIXRTN	TM	FPIND,R01REG	IS REG 1 A REG NUMBER	00004700
0000F2	4780 F09A		0009A	48		BZ	FIXLE	NO, BRANCH	00004800
0000F6	458C F7BE		007BE	49		BAL	R08,TSTFPR	CONVERT REGISTER TO ADDRESS	00004900
0000FA	7821 CCCC		00000	50	FIXLE	LE	FPR2,0(R01)	LOAD FP NUMBER	00005000
0000FE	9180 F864	C0864		51		TM	FPIND,SINGLE	IS IT SP	00005100
000102	4710 F0AA		000AA	52		BO	*+8	YES, BRANCH	00005200
000106	6821 CCCC		00000	53		LC	FPR2,0(R01)	NO, LOAD DP	00005300
00010A	2002			54		LPDR	FPR0,FPR2	LOAD THE NUMBER POSITIVE	00005400
00010C	6E00 F7F8		007F8	55		AW	FPR0,FXFLTCON	ADD 0*16**14	00005500
000110	6000 F800		00800	56		STD	FPR0,FPWRK1	STORE IT	00005600
000114	4780 F0CA		000CA	57		BZ	FIXZERO	BRANCH IF RESULT WAS A ZERO	00005700
000118	D503 F800 F7F8	C0800	C07F8	58		CLC	FPWRK1(4),FXFLTCON	CHECK SIZE OF NUMBER	00005800
00011E	4770 F218		00218	59		BNE	NUMBERR	ERROR IF TOO BIG	00005900
000122	918C F804	C0804		60		TM	FPWRK1+4,X'80'	CHECK HI ORDER BIT	00006000
000126	4710 F218		00218	61		BO	NUMBERR	NUMBER TOO BIG IF IT-S A 1	00006100
00012A	5820 F804		00804	62	FIXZERO	L	R02,FPWRK1+4	LOAD THE INTEGER	00006200
00012E	2222			63		LTDR	FPR2,FPR2	WAS FP NUMBER NEGATIVE	00006300
000130	4780 F0D6		000D6	64		BNL	*+6	NO, BRANCH	00006400
000134	1122			65		LNR	R02,R02	YES, COMPLEMENT THE INTEGER	00006500
000136	1810			66		LR	R01,R00	LOAD THE INTEGER ADDRESS	00006600
000138	914C F864	C0864		67		TM	FPIND,R00REG	IS REG 0 A REG NUMBER	00006700
00013C	4780 F0E8		000E8	68		BZ	FIXST	NO, BRANCH	00006800
000140	8910 0002		00002	69		SLL	R01,2	MULT NUMBER BY 4	00006900
000144	4111 F820		00820	70		LA	R01,SAVER(R01)	CALCULATE STORAGE ADDRESS	00007000
000148	5021 0000		00000	71	FIXST	ST	R02,0(R01)	STORE THE NUMBER WHERE IT BELON	00007100
				72	*				00007200

```
FPRT   FLOATING POINT ROUTINES - GENERAL EXIT                                                                    PAGE    4

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                        14 SEP 70

000CEC 68CC F7D8            007D8    74 EXIT     LD    FPR0,FPSAVE          RESTORE THE FP REGS                 00007400
000CF0 6820 F7E0            007E0    75          LD    FPR2,FPSAVE+8                                            00007500
000CF4 6840 F7E8            007E8    76          LD    FPR4,FPSAVE+16                                           00007600
000CF8 6860 F7F0            007F0    77          LD    FPR6,FPSAVE+24                                           00007700
000CFC 98CF F820            00820    78          LM    R00,R15,SAVER        RESTORE THE GP REGISTERS            00007800
0001C0 C4E0                          79          SPM   R14                  RESTORE THE PROGRAM MASK            00007900
0001C2 9200 F865      C0865          80          MVI   IPTIND,0             RESET I/O INDICATOR                 00008000
0001C6 9110 F864      C0864          81          TM    FPIND,FPIOSW         IS IT A CONVERSION ROUTINE RETU     00008100
0001CA 9200 F864      C0864          82          MVI   FPIND,0              RESET THE INDICATOR                 00008200
0001CE 4710 F118            00118    83          BO    CONVRTN              BRANCH IF AN I/O OPERATION          00008300
000112 4400 F11A            0011A    84          EX    R00,LOADRTN          LOAD RETURN CODE                    00008400
000116 07FE                          85          BR    R14                  RETURN TO CALLER                    00008500
                                     86 *                                                                        00008600
000118 070F                          87 CONVRTN  NOPR  R15                  KILL OVERLAP ON MODEL 91            00008700
00011A 41FC 0000            00000    88 LOADRTN  LA    R15,0                LOAD THE RETURN CODE                00008800
00011C                               89 RETCODE  EQU   LOADRTN+3            RETURN CODE ADDRESS                 00008900
00011E 47FE 0002            00002    90          B     2(R14)               RETURN AROUND FORMAT                00009000
                                     91 *                                                                        00009100
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

000122 9680 F864       C0864          93 FLOATS   CI    FPIND,SINGLE          INDICATE SINGLE PRECISION        00009300
000126 5020 F828              00828   94 FLOATD   ST    R02,SAVER+8           SAVE ENTRY REGISTER              00009400
00012A 4520 F044              00044   95          BAL   R02,ENTRY             GO TO GENERAL ENTRY              00009500
                                      96 *                                                                       00009600
00012E 9120 F864       C0864          97          TM    FPIND,R01REG          IS REG 1 A REGISTER NUMBER       00009700
000132 4780 F13F              0013E   98          BZ    FLTINTAD              BRANCH IF NOT                    00009800
000136 8910 0002              00002   99          SLL   R01,2                 MULT REG # BY 4                  00009900
00013A 4111 F820              00820  100          LA    R01,SAVER(R01)        CALCULATE ADDRESS IN SAVE AREA   00010000
00013E 5821 0000              00000  101 FLTINTAD L     R02,0(R01)            LOAD ACTUAL NUMBER               00010100
000142 1012                          102          LPR   R01,R02               LOAD THE ABSOLUTE VALUE          00010200
000144 D207 F800 F7F8 C0800 007F8    103          MVC   FPWRK1,FXFLTCON       SET EXPONENT                     00010300
00014A 5010 F804              00804  104          ST    R01,FPWRK1+4          STORE INTEGER                    00010400
00014E 6A00 F800              00800  105          AD    FPR0,FPWRK1           ADD AND NORMALIZE                00010500
000152 1222                          106          LTR   R02,R02               TEST FOR NEGATIVE INTEGER        00010600
000154 478C F15A              0015A  107          BNL   *+6                   BRANCH IF NOT NEGATIVE           00010700
000158 2100                          108          LNDR  FPR0,FPR0             COMPLEMENT THE FLOATING NUMBER   00010800
00015A 1810                          109          LR    R01,R00               GET FP NUMBER ADDRESS            00010900
00015C 9140 F864       C0864         110          TM    FPIND,R00REG          IS FP NUMBER TO BE PUT IN FP RE  00011000
000160 478C F168              00168  111          BZ    FPSTE                 NO, BRANCH AROUND ADDRESS CALCU  00011100
000164 4580 F7BE              007BE  112          BAL   R08,TSTFPR            CONVERT REGISTER TO ADDRESS      00011200
000168 7001 0000              00000  113 FPSTE    STE   FPR0,0(R01)           STORE SINGLE                     00011300
00016C 9180 F864       C0864         114          TM    FPIND,SINGLE          IS SP ALL THAT IS WANTED         00011400
000170 471C F0EC              000EC  115          BC    EXIT                  YES, GO TO EXIT                  00011500
000174 6001 0000              00000  116          STD   FPR0,0(R01)           NO, STORE DOUBLE                 00011600
000178 47F0 F0EC              000EC  117          B     EXIT                  GO TO EXIT                       00011700
                                     118 *                                                                       00011800
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT				14 SEP 70
000178	96F0 F864	00864		120	IPTSF	OI	FPIND,SINGLE	INDICATE SINGLE PRECISION	00012000
00017C	47F0 F18C		0018C	121	IPTCF	B	IPTCOMN	GO TO COMMON INPUT ENTRY	00012100
000180	96F0 F864	00864		122	IPTSE	OI	FPIND,SINGLE	INDICATE SINGLE PRECISION	00012200
000184	96C8 F864	00864		123	IPTCE	OI	FPIND,EFMT	INDICATE E FORMAT	00012300
				124	*				00012400
00018C	5020 F828		00828	125	IPTCOMN	ST	R02,SAVER+8	STORE ENTRY REGISTER	00012500
000190	9610 F864	00864		126		OI	FPIND,FPIOSW	INDICATE SPECIAL RETURN	00012600
000194	4520 F044		00044	127		BAL	R02,ENTRY	GO TO ENTRY ROUTINE	00012700
				128	*				00012800
000198	D201 F866 E000	00866	00000	129		MVC	FORMAT,0(R14)	MOVE THE FORMAT	00012900
00019E	9520 F866	00866		130		CLI	W,32	CHECK -W-	00013000
0001A2	4720 F210		00210	131		BH	FMTERR	IF TOO BIG	00013100
0001A6	9501 F866	00866		132		CLI	W,1	CHECK MINIMUM SPECIFICATIONS	00013200
0001AA	4740 F210		00210	133		BL	FMTERR	IF IN ERROR	00013300
				134	*				00013400
0001AE	1B22			135		SR	R02,R02		00013500
0001B0	4320 F866		00866	136		IC	R02,W		00013600
0001B4	C620			137		BCTR	R02,R00		00013700
0001B6	4420 F7A6		007A6	138		EX	R02,MVCFMT	MOVE THE CHARACTER STRING	00013800
0001BA	1222			139		LTR	R02,R02	TEST FOR ONLY 1 LONG	00013900
0001BC	4780 F1D4		001C4	140		BZ	SETEXP		00014000
				141	*				00014100
0001C0	4130 F868		00868	142		LA	R03,CHAR		00014200
0001C4	95C5 3000	00000		143	LOOKEXPN	CLI	0(R03),C'E'	LOOK FOR AN -E-	00014300
0001C8	4780 F1DC		001DC	144		BE	ENDEXPNT		00014400
0001CC	4133 0001		00001	145		LA	R03,1(R03)		00014500
0001D0	4620 F1C4		001C4	146		BCT	R02,LOOKEXPN	LOOP BACK FOR NEXT CHARACTER	00014600
0001D4	92C0 F8A9	008A9		147	SETEXP	MVI	EXPNT,0	SET EXPONENT TO ZERO	00014700
0001D8	47F0 F280		00280	148		B	SIGNSCAN		00014800
				149	*				00014900
0001DC	9240 3000	00000		150	ENDEXPNT	MVI	0(R03),C' '	KILL THE -E-	00015000
0001E0	96C8 F864	00864		151		OI	FPIND,EFMT	FORCE E FORMAT	00015100
0001E4	4133 0001		00001	152		LA	R03,1(R03)		00015200
0001E8	91F0 3000	00000		153		TM	0(R03),X'F0'	IS IT A NUMBER?	00015300
0001EC	47EC F1F8		001F8	154		BNO	*+12	NO, MAYBE IT-S A SIGN	00015400
0001F0	95F9 3000	00000		155		CLI	0(R03),C'9'	IS IT A TOO BIG NUMBER	00015500
0001F4	47DC F244		00244	156		BNH	EXPNTDIG	NO, IT-S A NUMBER	00015600
0001F8	954E 3000	00000		157		CLI	0(R03),C'+'	HOW ABOUT A REDUNDENT CHAR	00015700
0001FC	4780 F22C		0022C	158		BE	NEGEXPNT+4	YES, IT IS	00015800
0002CC	9560 3000	00000		159		CLI	0(R03),C'-'	IS IT A VALID MINUS	00015900

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT			14 SEP 70
0002C4	478C F228		00228	160		BE	NEGEXPNT	BRANCH IF IT IS	00016000
				161	*				00016100
0002C8	9208 F11D		C011D	162	CHARERR	MVI	RETCODE,INVALCHR	SET INVALID CHARACTER RETURN	00016200
0002CC	47F0 F0EC		000EC	163		B	EXIT		00016300
				164	*				00016400
0002D0	9208 F11D		C011D	165	FMTERR	MVI	RETCODE,INVALFMT	SET FORMAT ERROR RETURN CODE	00016500
0002D4	47F0 F0EC		000EC	166		B	EXIT	EXIT	00016600
				167	*				00016700
0002D8	9204 F11D		C011D	168	NUMBERR	MVI	RETCODE,INVALNUM	SET ERROR FOR A BAD BOY	00016800
0002DC	47F0 F0EC		000EC	169		B	EXIT	AND EXIT	00016900
				170	*				00017000
0002E0	9210 F11D		C011D	171	PROGERR	MVI	RETCODE,INVALPRG	PROGRAM ERROR RETURN	00017100
0002E4	47F0 F0EC		000EC	172		B	EXIT		00017200
				173	*				00017300
0002E8	964D F865		C0865	174	NEGEXPNT	OI	IPTIND,EXPNEG	SET THAT THE EXPONENT IS NEG	00017400
0002EC	9240 3000		C0000	175		MVI	0(R03),C' '	BLANK THE SIGN OUT	00017500
0002F0	4133 C001		00001	176		LA	R03,1(R03)	FIX THE POINTER	00017600
0002F4	91F0 3000		C0000	177		TM	0(R03),X'F0'	IS THERE A NUMBER THERE	00017700
0002F8	47E0 F208		00208	178		BNO	CHARERR	NO, CHARACTER IS INVALID	00017800
0002FC	95F9 3000		C0000	179		CLI	0(R03),C'9'		00017900
000300	4720 F208		00208	180		BH	CHARERR	ERROR IF GREATER THAN 9	00018000
				181	*				00018100
000304	91F0 3001		00001	182	EXPNTDIG	TM	1(R03),X'F0'	ARE 2 DIGITS PRESENT?	00018200
000308	47E0 F254		00254	183		BNO	*+12	NO, CONVERT ONLY 1	00018300
00030C	95F9 3001		C0001	184		CLI	1(R03),C'9'	CHECK FOR VALID DIGIT	00018400
000310	47D0 F25E		0025E	185		BNH	DOUBEXP	VALID, CONVERT 2 DIGITS	00018500
000314	F270 F808 3000		C0808	186		PACK	FPWRK2,0(1,R03)	NO, PACK ONLY ONE DIGIT	00018600
00031A	47F0 F26C		0026C	187		B	EXPCONV		00018700
00031E	F271 F808 3000		C0808	188	DOUBEXP	PACK	FPWRK2,0(2,R03)	PACK 2 DIGITS	00018800
000324	9240 3000		C0000	189		MVI	0(R03),C' '	BLANK OUT 1ST DIGIT	00018900
000328	4133 C001		00001	190		LA	R03,1(R03)	MESS AROUND WITH THE POINTER	00019000
				191	*				00019100
00032C	9240 3000		C0000	192	EXPCONV	MVI	0(R03),C' '	BLANK OUT 1ST OR 2ND DIGIT	00019200
000330	954C 3001		C0001	193		CLI	1(R03),C' '	A BLANK MUST FOLLOW LAST DIGIT	00019300
000334	477D F208		00208	194		BNE	CHARERR	CHARACTER OF THE EXPONENT	00019400
000338	4F30 F808		C0808	195		CVB	R03,FPWRK2	CONVERT IT TO BINARY	00019500
00033C	423C F8A9		008A9	196		STC	R03,EXPNT	SAVE THE EXPONENT	00019600
				197	*				00019700
000340	4320 F866		00866	198	SIGNSCAN	IC	R02,W	INITIALIZE FOR LEFT SHIFT SCAN	00019800
000344	954C F868		00868	199	SHIFTL	CLI	CHAR,C' '	IS THERE A LEADING BLANK	00019900

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

0002E8 4770 F29A             0029A  200          BNE   CHKWIN                 NO, BRANCH                      00020000
0002EC D21F F868 F869 00868 00869  201          MVC   CHAR(32),CHAR+1         YES, REMOVE IT                  00020100
0002F2 4620 F284             00284  202          BCT   R02,SHIFTL              LOOK BACK                       00020200
0002F6 47F0 F440             00440  203          B     FPINST                  GO TO INPUT EXIT                00020300
                                    204 *                                                                        00020400
0002FA 4220 F866             00866  205 CHKWIN   STC   R02,W                   STORE THE NEW -W-               00020500
0002FE 9501 F866      00866         206          CLI   W,1                     CHECK FOR ONLY 1 DIGIT LEFT     00020600
0002A2 4780 F2BE             002BE  207          BE    DECIN                                                   00020700
                                    208 *                                                                        00020800
0002A6 954E F868      00868         209          CLI   CHAR,C'+'               CHECK FOR A SIGN                00020900
0002AA 4780 F2BA             002BA  210          BE    NUMNEGIN                                                00021000
0002AE 9560 F868      00868         211          CLI   CHAR,C'-'               CHECK FOR NEGATIVE              00021100
0002B2 4770 F2BE             002BE  212          BNE   DECIN                   BRANCH IF NOT A SIGN            00021200
0002B6 9680 F865      00865         213          OI    IPTIND,IPTNEG           SET THE NEGATIVE SWITCH         00021300
0002BA 92F0 F868      00868         214 NUMNEGIN MVI   CHAR,C'0'               KILL THE SIGN                   00021400
                                    215 *                                                                        00021500
0002BE 4130 F868             00868  216 DECIN    LA    R03,CHAR                SET FOR SCAN                    00021600
0002C2 9540 3000      00000         217 DECLOOK  CLI   0(R03),C' '             LOOK FOR LAST CHAR              00021700
0002C6 4780 F2D2             002D2  218          BE    LASTCHAR                YEAH, FOUND IT                  00021800
0002CA 4133 0001             00001  219          LA    R03,1(R03)                                              00021900
0002CE 47F0 F2C2             002C2  220          B     DECLOOK                                                 00022000
0002D2 4140 F887             00887  221 LASTCHAR LA    R04,CHAR+31             LAST CHARACTER IN STRING        00022100
0002D6 1853                         222          LR    R05,R03                 MOVE TO R05 TO KEEP R03 INTACT  00022200
0002D8 1945                         223 CHARLOOP CR    R04,R05                 ARE WE THRU?                    00022300
0002DA 4780 F2EE             002FE  224          BE    CHARTHRU                YES, STRING IS OK               00022400
0002DE 9540 5000      00000         225          CLI   0(R05),C' '             CHECK FOR BLANK                 00022500
0002E2 4770 F208             00208  226          BNE   CHARERR                 IF NOT, THAT'S A NO-NO          00022600
0002E6 4155 0001             00001  227          LA    R05,1(R05)              INCREMENT                       00022700
0002EA 47F0 F2D8             002D8  228          B     CHARLOOP                LOOP BACK FOR NEXT CHARACTER    00022800
                                    229 *                                                                        00022900
0002EE 4140 F868             00868  230 CHARTHRU LA    R04,CHAR                SET THE SCAN POINTER            00023000
0002F2 9501 F866      00866         231          CLI   W,1                     CHECK IF ONLY 1 DIGIT LONG      00023100
0002F6 4780 F310             00310  232          BE    NODEC                                                   00023200
0002FA 1943                         233 DECLOOP  CR    R04,R03                 CHECK FOR THE END               00023300
0002FC 4780 F310             00310  234          BE    NODEC                                                   00023400
0003CC 954B 4000      00000         235          CLI   0(R04),C'.'             IS THERE A DECIMAL POINT        00023500
0003C4 4780 F316             00316  236          BE    FNDDECML                YES                             00023600
0003C8 4144 0001             00001  237          LA    R04,1(R04)              INCREMENT                       00023700
0003CC 47F0 F2FA             002FA  238          B     DECLOOP                                                 00023800
                                    239 *                                                                        00023900
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

000310 1755                          240 NODEC     XR    R05,R05                SET THE EXPONENT TO ZERO          00024000
000312 47F0 F326            00326    241           B     ADJEXPNT                                                 00024100
                                     242 *                                                                        00024200
000316 1853                          243 FNDDECML  LR    R05,R03                LOAD BLANK ADDRESS                00024300
000318 0630                          244           BCTR  R03,R00                CHANGE REGISTER 3                 00024400
00031A 1854                          245           SR    R05,R04                                                  00024500
00031C 0650                          246           BCTR  R05,R00                                                  00024600
00031E 4450 F79A            0079A    247           EX    R05,DECMVC             SHIFT ONE TO OVERLAY DECIMAL PT   00024700
000322 9620 F865      00865          248           OI    IPTIND,IGND            IGNORE -D- SPECIFICATIONS         00024800
                                     249 *                                                                        00024900
000326 1355                          250 ADJEXPNT  LCR   R05,R05                SET THE EXPONENT TO NEGATIVE      00025000
000328 1766                          251           XR    R06,R06                CLEAR R06                         00025100
00032A 436C F8A9            008A9    252           IC    R06,EXPNT                                                00025200
00032E 9140 F865      00865          253           TM    IPTIND,EXPNEG          IS THE EXPONENT UP TO NOW NEG     00025300
000332 478C F338            00338    254           BZ    *+6                    NO, BRANCH                        00025400
000336 1366                          255           LCR   R06,R06                YES, FIX THAT LITTLE RED WAGON    00025500
000338 94BF F865      00865          256           NI    IPTIND,255-EXPNEG      RESET THE NEG EXPNT INDICATOR     00025600
00033C 1A56                          257           AR    R05,R06                ADD EM TOGETHER                   00025700
00033E 9120 F865      00865          258           TM    IPTIND,IGND            SHOULD THE -D- BE USED            00025800
000342 4710 F34E            0034E    259           BC    TSTSIGN                BRANCH IF NOT                     00025900
000346 1766                          260           XR    R06,R06                CLEAR IT AGAIN                    00026000
000348 436C F867            00867    261           IC    R06,D                  LOAD -D- BECAUSE THERE'S NO '.'   00026100
00034C 1856                          262           SR    R05,R06                SUBTRACT IT OFF                   00026200
00034E 1255                          263 TSTSIGN   LTR   R05,R05                IS IT NOW NEGATIVE                00026300
000350 478C F35A            0035A    264           BNL   *+10                   BRANCH IF IT'S POSITIVE           00026400
000354 9640 F865      00865          265           OI    IPTIND,EXPNEG          SET INDICATOR                     00026500
000358 1055                          266           LPR   R05,R05                FIX THE REGISTER                  00026600
00035A 4250 F8A9            008A9    267           STC   R05,EXPNT              AND STORE IT ALL BACK             00026700
                                     268 *                                                                        00026800
00035E 4140 F868            00868    269           LA    R04,CHAR                                                 00026900
000362 1943                          270 NUMBLOOP  CR    R04,R03                IS THIS LOOP THRU?                00027000
000364 478C F380            00380    271           BE    CNVRTIPT               YES, TIME TO CONVERT              00027100.
000368 91F0 4000      00000          272           TM    0(R04),X'F0'           IS THE CHARACTER A NUMBER         00027200
00036C 47EC F208            00208    273           BNO   CHARERR                                                  00027300
000370 95F9 4000      00000          274           CLI   0(R04),C'9'                                              00027400
000374 4720 F208            00208    275           BH    CHARERR                ERROR IF GREATER THAN 9           00027500
000378 4144 0001            00001    276           LA    R04,1(R04)                                               00027600
00037C 47F0 F362            00362    277           B     NUMBLOOP                                                 00027700
                                     278 *                                                                        00027800
000380 414C F868            00868    279 CNVRTIPT  LA    R04,CHAR               LOAD POINTER                      00027900
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

0003E4 1B43                          280          SR    R04,R03          GET LENGTH                          00028000
0003E6 4780 F218            00218    281          BZ    NUMBERR          ERROR IF NO LENGTH                  00028100
0003EA 1C44                          282          LPR   R04,R04          CHANGE TO POSITIVE                  00028200
0003EC 1766                          283          XR    R06,R06                                              00028300
0003EE 1876                          284          LR    R07,R06          PRELOAD THE SUM REGISTERS           00028400
0003F0 4180 0009            00009    285          LA    R08,9            LOAD CONSTANT                       00028500
0003F4 1948                          286          CR    R04,R08          CHECK LENGTH                        00028600
0003F6 47D0 F3C0            003C0    287          BNH   SECHALF          BRANCH IF NINE OF LESS              00028700
0003FA 1B48                          288          SR    R04,R08          SUBTRACT 9                          00028800
0003FC 4180 000A            0000A    289          LA    R08,10                                               00028900
0003A0 1948                          290          CR    R04,R08          COMPARE TO 10                       00029000
0003A2 4720 F218            00218    291          BH    NUMBERR          YES, THIS IS A GOOF                 00029100
0003A6 C640                          292          BCTR  R04,R00          FIX FOR PACKER                      00029200
0003A8 4440 F7A0            007A0    293          EX    R04,PACK         PACK IT                             00029300
0003AC 4F70 F808            00808    294          CVB   R07,FPWRK2       CONVERT IT TO BINARY                00029400
0003B0 5C60 F860            00860    295          M     R06,TENNINE      MULTIPLY IT BY 10**9                00029500
0003B4 4114 F869            00869    296          LA    R01,CHAR+1(R04)  SET ADDRESS FOR MOVE                00029600
0003B8 4140 0009            00009    297          LA    R04,9            SET LENGTH                          00029700
0003BC 4440 F7A6            007A6    298          EX    R04,MVCFMT       SHIFT LEFT FOR NEXT CONVERT         00029800
                                     299 *                                                                    00029900
0003C0 C640                          300 SECHALF  BCTR  R04,R00          FIX FOR PACK                        00030000
0003C2 4440 F7A0            007A0    301          EX    R04,PACK         PACK WHATS LEFT                     00030100
0003C6 4F50 F808            00808    302          CVB   R05,FPWRK2       CONVERT IT                          00030200
0003CA 1E75                          303          ALR   R07,R05          ADD IT TO WHAT WAS BEFORE           00030300
0003CC 47C0 F3D6            003D6    304          BC    12,*+10          SKIP ADD IF NO CARRY                00030400
0003D0 4180 0001            00001    305          LA    R08,1            LOAD A 1                            00030500
0003D4 1A68                          306          AR    R06,R08          ADD THE OVERFLOW                    00030600
0003D6 9067 F800            00800    307          STM   R06,R07,FPWRK1   STORE IT IN CORE                    00030700
0003DA 91FF F800      00800          308          TM    FPWRK1,X'FF'     IS THE HI ORDER CLEAR               00030800
0003DE 924E F800      00800          309          MVI   FPWRK1,X'4E'     SET NORMAL EXPONENT                 00030900
0003E2 4780 F406            00406    310          BZ    NOBIGIN          BRANCH IF IT WAS CLEAR              00031000
0003E6 8C60 0004            00004    311          SRDL  R06,4            SHIFT OVER 4                        00031100
0003EA 9067 F800            00800    312          STM   R06,R07,FPWRK1   RE-STORE THE NUMBER                 00031200
0003EE 910F F800      00800          313          TM    FPWRK1,X'0F'     TEST AGAIN                          00031300
0003F2 924F F800      00800          314          MVI   FPWRK1,X'4F'     SET AN ADJUSTED EXPONENT            00031400
0003F6 4780 F406            00406    315          BZ    NOBIGIN          BRANCH IF NOW OK                    00031500
0003FA 8C60 0004            00004    316          SRDL  R06,4            DO IT AGAIN                         00031600
0003FE 9067 F800            00800    317          STM   R06,R07,FPWRK1                                       00031700
0004C2 9250 F800      00800          318          MVI   FPWRK1,X'50'     THIS TIME DON'T CHECK               00031800
                                     319 *                                                                    00031900
```

```
  LOC   OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    00032000
000C6 6A60 F8CC             00CC   32C NOBIGIN  AD    FPR6,FPWRK1      ADD AND NORMALIZE                    00032100
000CA 918C F865       C0865        321         TM    IPTIND,IPTNEG    WAS IT NEGATIVE                      00032200
000CE 47EC F414             00414  322         BNC   *+6              BRANCH IF NOT                        00032300
00412 2366                         323         LCDR  FPR6,FPR6        COMPLEMENT IT IF IT WAS              00032400
00414 91FF F8A9       C08A9        324         TM    EXPNT,X'FF'      CHECK IF EXPNT EXISTS                00032500
00418 4780 F440             00440  325         BZ    FPINST           IF NOT, WE'RE ALMOST THRU            00032600
0041C 68C0 F810             C0810  326         LD    FPRC,ONE                                              00032700
00420 6820 F818             C0818  327         LD    FPR2,TEN         LOAD CONSTANTS FOR EXPONENT FIX      00032800
00424 1744                         328         XR    R04,R04                                               00032900
00426 4340 F8A9             00BA9  329         IC    R04,EXPNT        LOAD EXPNT POWER FOR 10              00033000
0042A 2CC2                         330 FPINMULT MDR   FPRC,FPR2        MULTIPLY BY 10                       00033100
0042C 4640 F42A             0042A  331         BCT   R04,FPINMULT     LOOP UNTIL PRODUCT IS FORMED         00033200
                                   332 *                                                                    00033300
00430 9140 F865       C0865        333         TM    IPTIND,EXPNEG    IS THE EXPONENT NEGATIVE             00033400
00434 4710 F43E             0043E  334         BC    DIVIN            YES, BRANCH                          00033500
00438 2C6C                         335         MDR   FPR6,FPRO        MULTIPLY BY WHATEVER                 00033600
0043A 47FC F44C             C044C  336         B     FPINST           GO TO INPUT EXIT                     00033700
0043E 2C60                         337 DIVIN    DDR   FPR6,FPRC        DIVIDE BY WHATEVER                   00033800
                                   338 *                                                                    00033900
00440 181C                         339 FPINST   LR    R01,R00          MOVE FP NUMBER ADDRESS               00034000
00442 9140 F864       C0864        340         TM    FPIND,RCCREG     IS IT A REGISTER NUMBER              00034100
00446 478C F44E             C044E  341         BZ    INSTORE          NO, BRANCH                           00034200
0044A 458C F7BE             C07BE  342         BAL   R08,TSTFPR       CONVERT REGISTER TO ADDRESS          00034300
0044E 9180 F864       C0864        343 INSTORE  TM    FPIND,SINGLE     IS IT SINGLE PRECISION               00034400
00452 4710 F45E             0045E  344         BC    INROUND          YES, NUMBER MUST BE ROUNDED          00034500
00456 6041 CCC0             00000  345         STD   FPR6,0(R01)      YES, STORE ACCORDINGLY               00034600
0045A 47FC FCFC             00FEC  346         B     EXIT             GO TO YOU KNOW WHERE                 00034700
                                   347 *                                                                    00034800
0045E 6060 F800             00BC0  348 INRCUND  STD   FPR6,FPWRK1     STORE DOUBLE                         00034900
00462 582C F8C0             00BC0  349         L     R02,FPWRK1       LOAD SHORT HALF                      00035000
00466 9180 F804       C08C4        350         TM    FPWRK1+4,X'80'   DOES IT NEED ROUNDING               00035100
0046A 4780 F474             00474  351         BZ    *+10             NO, BRANCH                           00035200
0046E 413C CC01             C0CC1  352         LA    R03,1                                                 00035300
00472 1E23                         353         ALR   R02,R03          ROUND IT                             00035400
00474 5021 CC00             00000  354         ST    R02,0(R01)       STORE IT                             00035500
00478 47F0 FCEC             00FEC  355         B     EXIT             AND LEAVE THIS PLACE                 00035600
                                   356 *                                                                    00035700
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

00047C 9680 F864       C0864          358 OUTSF    OI    FPIND,SINGLE          INDICATE SINGLE PRECISION       00035800
C004EC 47FC F48C              0048C   359 CUTCF    B     OUTCOMN               GO TO COMMON OUTPUT ENTRY       00035900
0004E4 9680 F864       C0864          360 CUTSE    OI    FPIND,SINGLE          INDICATE SINGLE PRECISION       00036000
0004E8 9608 F864       C0864          361 OUTCE    OI    FPIND,EFMT            INDICATE E FORMAT               00036100
                                      362 *                                                                      00036200
0004EC 5C20 F828              00828   363 CUTCCMN  ST    R02,SAVER+8           STORE ENTRY REGISTER            00036300
000490 9610 F864       C0864          364          OI    FPIND,FPIOSW          INDICATE I/O RETURN AROUND FMT  00036400
C00494 4520 F044              00044   365          BAL   R02,ENTRY             GO TO ENTRY ROUTINE             00036500
                                      366 *                                                                      00036600
000498 D201 F866 ECCC C0866 00000     367          MVC   FORMAT,0(R14)         MOVE THE FORMAT SPECS           00036700
00049E 9520 F866       C0866          368          CLI   W,32                  CHECK IT FOR ERRORS             00036800
C004A2 4720 F210              00210   369          BH    FMTERR                                                00036900
C004A6 95C2 F866       C0866          370          CLI   W,2                                                   00037000
0004AA 47D0 F210              00210   371          BNH   FMTERR                                                00037100
C004AE 413C 0002              00002   372          LA    R03,2                 W-D DIFF                        00037200
C004B2 9108 F864       C0864          373          TM    FPIND,EFMT            IS IT E                         00037300
C004B6 478C F4BE              0048E   374          BZ    *+8                                                   00037400
0004BA 413C 0006              00006   375          LA    R03,6                 YES, DIFFERENT W-D              00037500
0004BE 1722                           376          XR    R02,R02                                               00037600
C004CC 4320 F866              00866   377          IC    R02,W                                                 00037700
0004C4 1823                           378          SR    R02,R03                                               00037800
C004C6 4330 F867              00867   379          IC    R03,D                                                 00037900
0004CA 1923                           380          CR    R02,R03                                               00038000
0004CC 474C F210              00210   381          BL    FMTERR                                                00038100
                                      382 *                                                                      00038200
0004D0 9120 F864       00864          383          TM    FPIND,R01REG          IS R01 (FPADDRESS) A REG NUMBER 00038300
0004D4 478C F4DC              004DC   384          BZ    LDFPOUT               BRANCH IF NOT                   00038400
0004D8 4580 F7BE              007BE   385          BAL   R08,TSTFPR            CONVERT REGISTER TO ADDRESS     00038500
C004DC 7801 0000              00000   386 LDFPOUT  LE    FPR0,0(R01)           LOAD SINGLE                     00038600
C004EC 9180 F864       C0864          387          TM    FPIND,SINGLE          IS IT SP                        00038700
0004E4 4710 F4EC              004EC   388          BC    *+8                                                   00038800
C004E8 6801 0000              00000   389          LD    FPR0,0(R01)           NO, LOAD DOUBLE                 00038900
                                      390 *                                                                      00039000
0004EC 2AC2                           391          ADR   FPR0,FPR2             FORCE NORMALIZATION             00039100
C004EE 22CC                           392          LTDR  FPR0,FPR0             IS IT NEGATIVE                  00039200
C004FC 478C F4FA              004FA   393          BNL   *+10                  BRANCH IF NOT NEG               00039300
0004F4 9610 F865       C0865          394          CI    IPTIND,OUTNEG         SET SIGN INDICATOR              00039400
0004F8 2000                           395          LPDR  FPR0,FPR0             LOAD AS POSITIVE                00039500
                                      396 *                                                                      00039600
C004FA 413C F868              00868   397          LA    R03,CHAR              WORK AREA ADDRESS               00039700
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

0004FE 432C F866             00866   398          IC    R02,W                 FIELD LENGTH                  00039800
000502 1A32                          399          AR    R03,R02               LAST CHAR +1 IN R03           00039900
000504 9108 F864      00864          400          TM    FPIND,EFMT            IS IT E FORMAT                00040000
000508 478C F518             00518   401          BZ    FTYPFMT                                             00040100
00050C 418C 0004             00004   402          LA    R08,4                                               00040200
00051C 1B38                          403          SR    R03,R08               SUBTRACT 4 TO BYPASS EXPONENT 00040300
000512 D203 3000 F8A5 00000 008A5    404          MVC   0(4,R03),EXP          SET UP DUMMY EXPONENT         00040400
000518 432C F867             00867   405 FTYPFMT  IC    R02,D                 DECIMAL LENGTH                00040500
00051C 1843                          406          LR    R04,R03               MOVE                          00040600
00051E 1B42                          407          SR    R04,R02               BACK R04 UP TO DEC PT         00040700
000520 0640                          408          BCTR  R04,R00                                             00040800
000522 0640                          409          BCTR  R04,R00                                             00040900
000524 D202 4000 F8A2 00000 008A2    410          MVC   0(3,R04),OUTZERO                                    00041000
00052A 4144 0003             00003   411          LA    R04,3(R04)                                          00041100
00052E 9108 F864      00864          412          TM    FPIND,EFMT            CHECK FOR -E- FORMAT          00041200
000532 478C F53A             0053A   413          BZ    *+8                   BRANCH IF NOT -E- FORMAT      00041300
000536 9205 3000      00000          414          MVI   0(R03),C'E'           RESTORE THE -E-               00041400
00053A 1B34                          415          SR    R03,R04               GET THE LENGTH                00041500
00053C 47D0 F548             00548   416          BNH   EDITOR                BRANCH IF NO ZEROS NEEDED     00041600
000540 0630                          417          BCTR  R03,R00               FIX LENGTH FOR MOVE           00041700
000542 0640                          418          BCTR  R04,R00               SET BACK FOR MOVE             00041800
000544 4430 F782             00782   419          EX    R03,ZFILL             SET UP THE FIELD              00041900
                                     420 *                                                                   00042000
000548 2200                          421 EDITOR   LTDR  FPR0,FPR0             TEST FOR A ZERO               00042100
00054A 4780 F788             00788   422          BZ    FPOUTMVC              GO TO OUTPUT EXIT IF IT IS    00042200
00054E 1722                          423          XR    R02,R02               CLEAR R02                     00042300
000550 7000 F800             00800   424          STE   FPR0,FPWRK1           STORE FOR EXPONENT TEST       00042400
000554 922D F574      00574          425          MVI   INSTMDV,DDR           SET INSTRUCTION FOR A DIVIDE  00042500
000558 954E F800      00800          426          CLI   FPWRK1,X'4E'          COMPARE THE EXPONENT          00042600
00055C 478C F586             00586   427          BE    FIXEXPTH              BRANCH IF ALREADY OK          00042700
000560 4720 F560             00560   428          BH    *+12                  BRANCH IF DIVIDE OK           00042800
000564 922C F574      00574          429          MVI   INSTMDV,MDR           SET TO MULTIPLY INSTEAD OF DIV 00042900
000568 9640 F865      00865          430          OI    IPTIND,EXPNEG         SET NEGATIVE EXPONENT FLAG    00043000
00056C 6820 F818             00818   431          LD    FPR2,TEN              LOAD FP 10                    00043100
000570 2860                          432          LDR   FPR6,FPR0             SAVE THE NUMBER               00043200
000572 070F                          433          NOPR  R15                   MODEL 91 CATCH UP INSTRUCTION 00043300
000574 2002                          434 INSTMDV  MDR   FPR0,FPR2             MULTIPLY OR DIVIDE            00043400
000576 4122 0001             00001   435          LA    R02,1(R02)                                          00043500
00057A 7000 F800             00800   436          STE   FPR0,FPWRK1                                         00043600
00057E 954E F800      00800          437          CLI   FPWRK1,X'4E'          IS IT WHERE I WANT IT?        00043700
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

0005E2 4770 F574            00574   438          BNE   INSTMDV                 NO, DO IT AGAIN                 00043800
                                    439 *                                                                        00043900
0005E6 4220 F8A9            008A9   440 FIXEXPTH STC   R02,EXPNT               YES, STORE IT                   00044000
0005EA 6000 F800            00800   441          STD   FPR0,FPWRK1                                             00044100
0005EE 9200 F800      00800         442          MVI   FPWRK1,0                KILL EXPONENT (4E)              00044200
0005F2 9867 F800            00800   443          LM    R06,R07,FPWRK1                                          00044300
0005F6 5D60 F860            00860   444          D     R06,TENNINE             DIVIDE BY 10**9                 00044400
0005FA 4E60 F808            00808   445          CVD   R06,FPWRK2              RIGHT HALF                      00044500
0005FE 4E70 F800            00800   446          CVD   R07,FPWRK1              LEFT HALF                       00044600
0005A2 960F F807      00807         447          CI    FPWRK1+7,X'0F'          FIX THE SIGN                    00044700
0005A6 960F F80F      00800         448          CI    FPWRK2+7,X'0F'          FIX THE SIGN                    00044800
0005AA F3E7 F869 F800 00889 00800   449          UNPK  OUTCHAR(15),FPWRK1                                      00044900
0005B0 F384 F898 F808 00898 00808   450          UNPK  OUTCHAR+15(9),FPWRK2+3(5)                               00045000
0005B6 9108 F864      00864         451          TM    FPIND,EFMT              IS IT E FORMAT                  00045100
0005BA 4780 F688            00688   452          BZ    FFORMAT                 NO, BRANCH                      00045200
0005BE 1733                         453          XR    R03,R03                 CLEAR 3                         00045300
0005C0 4330 F866            00866   454          IC    R03,W                   WIDTH OF FIELD                  00045400
0005C4 4180 0006            00006   455          LA    R08,6                                                   00045500
0005C8 1838                         456          SR    R03,R08                 SUBTRACT 6                      00045600
0005CA 4380 F867            00867   457          IC    R08,D                                                   00045700
0005CE 1838                         458          SR    R03,R08                 SUBTRACT -D-                    00045800
0005D0 4160 F888            00888   459          LA    R06,OUTCHAR-1           INITIALIZE FOR LOOP             00045900
0005D4 4166 0001            00001   460 SIGLOOPE LA    R06,1(R06)              INCREMENT                       00046000
0005D8 95F0 6000      00000         461          CLI   0(R06),C'0'             IS IT SIGNIFICANT               00046100
0005DC 4780 F5D4            005D4   462          BE    SIGLOOPE                NO, BRANCH                      00046200
0005E0 95FF 6000      00000         463          CLI   0(R06),X'FF'            IS IT THE END OF FIELD INDICATO 00046300
0005E4 4780 F788            00788   464          BE    FPOUTMVC                YES, IT A ZERO NUMBER           00046400
0005E8 9240 F868      00868         465          MVI   CHAR,C' '               SET SIGN FIELD                  00046500
0005EC 9110 F865      00865         466          TM    IPTIND,OUTNEG           IS IT NEGATIVE                  00046600
0005F0 4780 F5F8            005F8   467          BZ    *+8                     NO, BRANCH                      00046700
0005F4 9260 F868      00868         468          MVI   CHAR,C'-'               YES, SET SIGN                   00046800
                                    469 *                                                                        00046900
0005F8 4140 F869            00869   470          LA    R04,CHAR+1                                              00047000
0005FC 0630                         471          BCTR  R03,R00                 ADJUST POINTER                  00047100
0005FE 1233                         472          LTR   R03,R03                 CHECK FOR NO PLACES LEFT OF .   00047200
0006C0 4740 F608            00608   473          BL    *+8                     BRANCH AROUND EX IF NEG         00047300
0006C4 4430 F7B8            007B8   474          EX    R03,EXPLFT                                              00047400
0006C8 4143 4002            00002   475          LA    R04,2(R03,R04)          ADJUST ADDRESS                  00047500
0006CC 4163 6001            00001   476          LA    R06,1(R03,R06)                                          00047600
000610 4150 F8A1            008A1   477          LA    R05,OUTCHAR+24          LOAD LAST ADDRESS               00047700
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

000614 1733                          478          XR    R03,R03                                                  00047800
000616 4330 F867             00867   479          IC    R03,D                                                    00047900
00061A 1233                          480          LTR   R03,R03                TEST THE DECIMAL                00048000
00061C 4780 F65A             0065A   481          BZ    EMVCOK                                                   00048100
00062C 0630                          482          BCTR  R03,R00                                                  00048200
000622 4430 F788             00788   483          EX    R03,EXPLFT             MOVE THE DECIMALS               00048300
000626 4173 6001             00001   484          LA    R07,1(R03,R06)                                           00048400
00062A 4143 4002             00002   485          LA    R04,2(R03,R04)         POINT TO SIGN OF EXPONENT       00048500
                                     486 *                                                                       00048600
00062E 1975                          487          CR    R07,R05                WAS TOO MUCH MOVED?             00048700
000630 47D0 F65A             0065A   488          BNH   EMVCOK                 NO, BRANCH                      00048800
000634 1B75                          489          SR    R07,R05                TAKE DIFFERENCE                 00048900
000636 1814                          490          LR    R01,R04                SAVE R04                        00049000
000638 4143 4001             00001   491          LA    R04,1(R03,R04)                                           00049100
00063C 1B47                          492          SR    R04,R07                                                  00049200
00063E 95FF 4000       00000         493          CLI   0(R04),X'FF'           IS THE CHARACTER A X'FF'?       00049300
000642 477C F220             00220   494          BNE   PROGERR                NO, I BLEW IT                   00049400
000646 92F0 4000       00000         495          MVI   0(R04),C'0'            FIX IT                          00049500
00064A 0670                          496          BCTR  R07,R00                                                  00049600
00064C 1277                          497          LTR   R07,R07                IS IT ZERO?                     00049700
00064E 4780 F65A             0065A   498          BZ    EMVCOK                 YES, MOVE IS COMPLETE           00049800
000652 0670                          499          BCTR  R07,R00                                                  00049900
000654 4470 F782             00782   500          EX    R07,ZFILL                                                00050000
000658 1841                          501          LR    R04,R01                RESTORE REG 4                   00050100
                                     502 *                                                                       00050200
00065A 1B56                          503 EMVCOK   SR    R05,R06                DO A LITTLE ARITHMETIC          00050300
00065C 4330 F8A9             008A9   504          IC    R03,EXPNT                                                00050400
000660 9140 F865       00865         505          TM    IPTIND,EXPNEG          TEST FOR A NEGATIVE EXPONENT    00050500
000664 4780 F66A             0066A   506          BZ    *+6                                                      00050600
000668 1133                          507          LNR   R03,R03                SET IT NEGATIVE FOR CALC        00050700
00066A 1A35                          508          AR    R03,R05                ADD PLACEMENT FACTOR            00050800
00066C 1233                          509          LTR   R03,R03                IS IT NEGATIVE                  00050900
00066E 4780 F676             00676   510          BNL   *+8                    NO, BRANCH                      00051000
000672 9260 4000       00000         511          MVI   0(R04),C'-'            SET IT MINUS                    00051100
000676 4E30 F800             00800   512          CVD   R03,FPWRK1                                               00051200
00067A 96CF F807       00807         513          OI    FPWRK1+7,X'0F'                                           00051300
00067E F311 4001 F806 00001 00806    514          UNPK  1(2,R04),FPWRK1+6(2)   PUT IT IN CHARACTER FIELD       00051400
000684 47F0 F788             00788   515          B     FPOUTMVC               GO TO MOVE IT TO USER           00051500
                                     516 *                                                                       00051600
000688 1733                          517 FFORMAT  XR    R03,R03                CLEAR REG 3                     00051700
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT			14 SEP 70
0006EA	1823			518		LR	R02,R03	SAME FOR REG 2	00051800
0006EC	433C F866		00866	519		IC	R03,W	GET FIELD WIDTH	00051900
0006F0	4320 F867		00867	520		IC	R02,D	GET DECIMAL WIDTH	00052000
0006F4	1832			521		SR	R03,R02	GET DIFFERENCE	00052100
0006F6	C63C			522		BCTR	R03,R00	AND ADJUST IT	00052200
0006F8	C630			523		BCTR	R03,R00		00052300
				524	*				00052400
0006FA	416C F888		00838	525		LA	R06,OUTCHAR-1	INITIALIZE POINTER	00052500
0006FE	4166 CC01		0CCC1	526	FLCCPDEC	LA	R06,1(R06)	INCREMENT	00052600
000702	95F0 6000		C0000	527		CLI	0(R06),C'0'	IS IT ZERO	00052700
000706	4780 F69E		0069E	528		BE	FLCCPDEC	YES, CONTINUE TO LOOP	00052800
00070A	95FF 6000		C0000	529		CLI	C(R06),X'FF'	IS IT THE TERMINATOR	00052900
00070E	4780 F788		00788	530		BE	FPOUTMVC	YES, EXIT TO USER	00053000
000712	415C F8A1		008A1	531		LA	R05,OUTCHAR+24		00053100
000716	1856			532		SR	R05,R06	GET LENGTH	00053200
000718	1777			533		XR	R07,R07		00053300
00071A	437C F8A9		008A9	534		IC	R07,EXPNT	LOAD EXPONENT	00053400
00071E	914C F865		C0865	535		TM	IPTIND,EXPNEG	IS IT NEAGTIVE	00053500
000722	4780 F6C8		006C8	536		BZ	*+6	NO, BRANCH	00053600
000726	1177			537		LNR	R07,R07	YES, COMPLEMENT	00053700
000728	1A57			538		AR	R05,R07	NUMBER OF DIGITS NEEDED TO	00053800
00072A	1953			539		CR	R05,R03	LEFT OF DECIMAL POINT	00053900
00072C	4720 F218		00218	540		BH	NUMBERR	BRANCH IF TOO BIG	00054000
000730	414C F869		00869	541		LA	R04,CHAR+1		00054100
000734	1A43			542		AR	R04,R03	ADD -W-	00054200
000736	1B45			543		SR	R04,R05	ADJUST	00054300
000738	C65C			544		BCTR	R05,R00	FIX FOR MVC	00054400
00073A	1255			545		LTR	R05,R05	IS LENGTH NEGATIVE?	00054500
00073C	4740 F6E4		006E4	546		BL	*+8	YES, SKIP MOVE	00054600
000740	445C F7B8		007B8	547		EX	R05,EXPLFT	MOVE THE GOODIES	00054700
000744	4165 6001		00001	548		LA	R06,1(R05,R06)	SET THE NEW POINTER VALUE	00054800
000748	4170 F8A1		008A1	549		LA	R07,OUTCHAR+24		00054900
00074C	1967			550		CR	R06,R07	COMPARE THE 2	00055000
00074E	4780 F766		00766	551		BE	SIGNOUT	IF EQUAL	00055100
000752	4740 F71C		0071C	552		BL	DECPLC	IF LESS THAN	00055200
000756	1867			553		SR	R06,R07	CLEAN THE FIELD UP	00055300
000758	4145 4001		00001	554		LA	R04,1(R05,R04)	DOCTOR EVERYTHING	00055400
00075C	1846			555		SR	R04,R06		00055500
00075E	95FF 4000		C0000	556		CLI	C(R04),X'FF'	CHECK FOR DELIMITER	00055600
000762	477C F22C		00220	557		BNE	PROGERR	BLOW UP IF ITS NOT FF	00055700

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    00056600
000726 92F0 4000       00000        558          MVI   0(R04),C'0'     CLEAR IT                                00056600
00072A C660                         559          BCTR  R06,R00                                                 00056900
00072C 1266                         560          LTR   R06,R06        TEST FOR NO MORE                         00056000
00072E 47B0 F766              00766 561          BE    SIGNOUT        IF NO MORE                               00056100
000732 C660                         562          BCTR  R06,R00        ADJUST FOR MVC                           00056200
000734 4460 F782              00782 563          EX    R06,ZFILL      ZERO FILL IT                             00056300
000738 47F0 F766              00766 564          B     SIGNOUT        GO TO SET THE  SIGN                      00056400
                                    565 *                                                                       00056500
00073C 4145 4002              00002 566 DECPLC   LA    R04,2(R05,R04) SET NEW ADDRESS                          00056600
000740 1B76                         567          SR    R07,R06        GET A DIFFERENCE                         00056700
000742 1927                         568          CR    R02,R07        CHECK THE LENGTH AGAINST -D-             00056800
000744 47B0 F72A              00728 569          BNL   *+6            BRANCH IF R07 IS OK                      00056900
000748 1B72                         570          LR    R07,R02        OTHERWISE, REPLACE WITH -D-              00057000
00074A 1277                         571          LTR   R07,R07        IS IT NONE                               00057100
00074C 4780 F766              00766 572          BZ    SIGNOUT        YES, WE-RE ALMOST THRU                   00057200
000750 C670                         573          BCTR  R07,R00        FIX FOR EXECUTE                          00057300
000752 4470 F788              00788 574          EX    R07,EXPLFT     MOVE THE DECIMAL PALCES                  00057400
000756 4177 6001              00001 575          LA    R07,1(R07,R06) NEXT CHARACTER                           00057500
00075A 4150 F8A1              008A1 576          LA    R05,OUTCHAR+24 DELIMITER CHARACTER                      00057600
00075E 1965                         577          CR    R06,R05        COMPARE THEM                             00057700
000760 4700 F766              00766 578          BNH   SIGNOUT        BRANCH IF TOO MUCH WAS NOT MOVE          00057800
000764 1B65                         579          SR    R06,R05        GET THE LENGTH NEEDED                    00057900
000766 4147 4001              00001 580          LA    R04,1(R07,R04) GET WHERE                                00058000
00076A 1B46                         581          SR    R04,R06        ADJUST SLIGHTLY                          00058100
00076C 95FF 4000       00000        582          CLI   0(R04),X'FF'   IS IT WHAT IT SHOULD BE                  00058200
000770 4770 F220              00220 583          BNE   PROGERR        NO, WE BLEW IT AGAIN                     00058300
000774 92F0 4000       00000        584          MVI   0(R04),C'0'    CLEAR THE DELIMITER                      00058400
000778 C660                         585          BCTR  R06,R00        FIX LENGTH FOR AMOUNT LEFT               00058500
00077A 1266                         586          LTR   R06,R06        TEST IT                                  00058600
00077C 4780 F766              00766 587          BZ    SIGNOUT        IF ONLY 1 CHARACTER                      00058700
000780 C66C                         588          BCTR  R06,R00        ADJUST FOR MOVE                          00058800
000782 4460 F782              00782 589          EX    R06,ZFILL      ZERO FILL THE AREA                       00058900
                                    590 *                                                                       00059000
000786 9110 F865       00865        591 SIGNOUT  TM    IPTIND,OUTNEG  IS A MINUS SIGN NEEDED                   00059100
00078A 4780 F788              00788 592          BZ    FPOLTMVC       NO, MOVE TO USER AREA                    00059200
00078E 414C F868              00868 593          LA    R04,CHAR       SET POINTER                              00059300
000792 954C 4000       00000        594 SIGNOUTL CLI   0(R04),C' '    IS IT A BLANK                            00059400
000796 4770 F782              00782 595          BNE   SETSIGN        NO, BRANCH                               00059500
00079A 4144 0001              00001 596          LA    R04,1(R04)     INCREMENT                                00059600
00079E 47F0 F772              00772 597          B     SIGNOUTL       LOOP BACK TO TEST NEXT CHARACTE          00059700
```

```
FPRT   FLOATING POINT ROUTINES - CONVERT FLOATING OUTPUT                                                    PAGE   18

  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  14 SEP 70

000782 C640                          598 SETSIGN  BCTR  R04,R00                BACK UP ONE                  00059800
000784 9260 4000      00000          599          MVI   0(R04),C'-'            SET THE MINUS SIGN           00059900
                                     600 *                                                                   00060000
000788 1810                          601 FPOUTMVC LR    R01,R00                HERE WE ARE, ALMOST THRU     00060100
00078A 1722                          602          XR    R02,R02                CLEAR R02                    00060200
00078C 4320 F866            00866    603          IC    R02,W                  LOAD FIELD WIDTH             00060300
000790 C620                          604          BCTR  R02,R00                ADJUST FOR MOVE              00060400
000792 4420 F7AC            007AC    605          EX    R02,MVCCHAR            MOVE TO USER-S AREA          00060500
000796 47F0 F0EC            000EC    606          B     EXIT                                                00060600
                                     607 *                                                                   00060700
```

```
FPRT   FLOATING POINT ROUTINES - EXECUTED INSTRUCTIONS, MISC.                                                    PAGE   19

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                        14 SEP 70

0007SA D200 4000 4001 00000 00001   609 DECMVC   MVC   0(0,R04),1(R04)        SHIFT LEFT ONE BYTE                00060900
0007A0 F270 F808 F868 00808 00868   610 PACK     PACK  FPWRK2(8),CHAR(0)                                          00061000
0007A6 D200 F868 1000 00868 00000   611 MVCFMT   MVC   CHAR(0),0(R01)         INSTRUCTION FOR MOVING CHARACTR    00061100
0007AC D200 1000 F868 00000 00868   612 MVCCHAR  MVC   0(0,R01),CHAR                                             00061200
0007B2 D200 4001 4000 00001 00000   613 ZFILL    MVC   1(0,R04),0(R04)        CONCATINATE RIGHT                  00061300
0007B8 D200 4000 6000 00000 00000   614 EXPLFT   MVC   0(0,R04),0(R06)        MOVE FROM ONE AREA TO ANOTHER      00061400
                                    615 *                                                                          00061500
0007BE 4210 F8AA            008AA   616 TSTFPR   STC   R01,FPREGTST           STORE THE FP REG NUMBER            00061600
0007C2 9101 F8AA      C08AA         617          TM    FPREGTST,X'01'         IS IT ODD?                         00061700
0007C6 C718                         618          BCR   1,R08                  YES, ASSUME IT-S AN ADDRESS        00061800
0007C8 9506 F8AA      C08AA         619          CLI   FPREGTST,FPR6          IS IT TOO LARGE?                   00061900
0007CC C728                         620          BCR   2,R08                  YES, TREAT AS ADDRESS              00062000
0007CE 8910 0002            00002   621          SLL   R01,2                  MULTIPLY BY 4                      00062100
0007D2 4111 F7D8            C07D8   622          LA    R01,FPSAVE(R01)        ADD SAVE AREA ADDRESS              00062200
0007D6 C7F8                         623          BR    R08                    REGISTER NUMBER IS VALID           00062300
                                    624 *                                                                          00062400
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

0007D8 0000000000000000                 626 FPSAVE   DC    4D'0'                  FP REGISTER SAVE AREA          00062600
0007F8 4E00000000000000                 627 FXFLTCON DC    XL8'4E00000000000000'  FIX/FLOAT CONSTANT             00062700
0008C0 0000000000000000                 628 FPWRK1   DC    D'0'                                                  00062800
0008C8 0000000000000000                 629 FPWRK2   DC    D'0'                                                  00062900
000810 4110000000000000                 630 ONE      DC    D'1'                                                  00063000
000818 41A0000000000000                 631 TEN      DC    D'10'                                                 00063100
00082C 0000000000000000                 632 SAVER    DC    16F'0'                 REGISTER SAVE AREA             00063200
00086C 3B9ACA00                         633 TENNINE  DC    F'1000000000'          10 ** 9                        00063300
000864 00                               634 FPIND    DC    X'0'                   INTERNAL SWITCHES              00063400
000865 00                               635 IPTIND   DC    X'00'                  INDICATOR BYTE                 00063500
000866                                  636 FORMAT   DS    0CL2                   AREA TO HOLD THE FORMAT CODE   00063600
000866                                  637 W        DS    AL1                    TOTAL FIELD WIDTH              00063700
000867                                  638 D        DS    AL1                    FRACTIONAL FIELD WIDTH         00063800
000868 404040404040404040               639 CHAR     DC    CL32' '                AREA FOR CHARACTER STRING      00063900
000888 40                               640 CHARTERM DC    C' '                   AREA TERMINATOR                00064000
000889 404040404040404040               641 OUTCHAR  DC    CL24' '                                               00064100
0008A1 FF                               642 OUTTERM  DC    X'FF'                                                 00064200
0008A2 F04BF0                           643 OUTZERO  DC    C'0.0'                 ZERO FOR OUTPUT                00064300
0008A5 C54EF0F0                         644 EXP      DC    C'E+00'                EXPONENT FOR E FORMAT          00064400
0008A9 00                               645 EXPNT    DC    X'0'                   EXPONENT HOLD AREA             00064500
0008AA 00                               646 FPREGTST DC    X'0'                   USED FOR FP REG TESTING        00064600
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT			14 SEP 70
0000E0				648	SINGLE	EQU	X'80'	SINGLE PRECISION INDICATOR 00064800
0000E4C				649	ROOREG	EQU	X'4C'	ROO IS A REG # INDICATOR 00064900
0000E2C				650	R01REG	EQU	X'20'	R01 IS A REG # INDICATOR 00065000
0000E10				651	FPICSW	EQU	X'10'	INDICATE RETURN AROUND FORMAT 00065100
0000E8				652	EFMT	EQU	X'08'	E FORMAT INDICATOR 00065200
				653	*			00065300
0000EC				654	IPTNEG	EQU	X'80'	NUMBER SHOULD BE NEGATIVE 00065400
0000E40				655	EXPNEG	EQU	X'4C'	EXPONENT IS NEGATIVE 00065500
0000E2C				656	IGNC	EQU	X'2C'	IGNORE -D- OF FORMAT 00065600
0000E10				657	OUTNEG	EQU	X'10'	OUTPUT NUMBER IS NEGATIVE 00065700
				658	*			00065800
0000E2C				659	MDR	EQU	X'2C'	MULTIPLY DOUBLE REGISTER 00065900
0000E2D				660	DDR	EQU	X'2D'	DIVIDE DOUBLE REGISTER 00066000
				661	*			00066100
				662		REGS		00066200
0000E0				663+	R00	EQU	0	
0000E1				664+	R01	EQU	1	
0000E2				665+	R02	EQU	2	
0000E3				666+	R03	EQU	3	
0000E4				667+	R04	EQU	4	
0000E5				668+	R05	EQU	5	
0000E6				669+	R06	EQU	6	
0000E7				670+	R07	EQU	7	
0000E8				671+	R08	EQU	8	
0000E9				672+	R09	EQU	9	
0000EA				673+	R10	EQU	10	
0000EB				674+	R11	EQU	11	
0000EC				675+	R12	EQU	12	
0000ED				676+	R13	EQU	13	
0000EE				677+	R14	EQU	14	
0000EF				678+	R15	EQU	15	
				679		REGS	FPR	00066300
0000E0				680+	FPR0	EQU	0	
0000E2				681+	FPR2	EQU	2	
0000E4				682+	FPR4	EQU	4	
0000E6				683+	FPR6	EQU	6	
				684	*			00066400
0000E4				685	INVALNUM	EQU	4	RETURN CODE FOR INVALID NUMBER 00066500
0000E8				686	INVALCHR	EQU	8	RETURN CODE FOR INVALID CHARS 00066600
0000EC				687	INVALFMT	EQU	12	RETURN CODE FOR INVALID FORMAT 00066700

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000C1C                            688 INVALPRG EQU   16                  RETURN CODE FOR PROGRAM ERROR   00066800
                                  689 *                                                               00066900
                                  690          PATCH 50                  PATCH AREA                    00067000
000BAB C0
000BAC                            691+         DC    0H'0'
000BAC                            692+         DS    0CL50
000BAC F8AC                       693+         DC    S(*)
000BAE F8AE                       694+         DC    S(*)
000BEC F8BC                       695+         DC    S(*)
000BE2 F8B2                       696+         DC    S(*)
000BE4 F8B4                       697+         DC    S(*)
000BE6 F8B6                       698+         DC    S(*)
000BE8 F8B8                       699+         DC    S(*)
000BEA F8BA                       700+         DC    S(*)
000BEC F8BC                       701+         DC    S(*)
000BEE F8BE                       702+         DC    S(*)
000BC0 F8C0                       703+         DC    S(*)
000BC2 F8C2                       704+         DC    S(*)
000BC4 F8C4                       705+         DC    S(*)
000BC6 F8C6                       706+         DC    S(*)
000BC8 F8C8                       707+         DC    S(*)
000BCA F8CA                       708+         DC    S(*)
000BCC F8CC                       709+         DC    S(*)
000BCE F8CE                       710+         DC    S(*)
000BD0 F8D0                       711+         DC    S(*)
000BD2 F8D2                       712+         DC    S(*)
000BD4 F8D4                       713+         DC    S(*)
000BD6 F8D6                       714+         DC    S(*)
000BD8 F8D8                       715+         DC    S(*)
000BDA F8DA                       716+         DC    S(*)
000BDC F8DC                       717+         DC    S(*)
                                  718          END                                                     00067100
```

SYMBOL	LEN	VALUE	DEFN	REFERENCES													
ADJEXPNT	2	0000326	250	241													
CHAR	32	0000B6E	639	42	43	43	142	199	201	201	209	211	214	216	221	230	269
				279	296	397	465	468	470	541	593	610	611	612			
CHARERR	4	0002C8	162	178	180	194	226	273	275								
CHARLOOP	2	0002D8	223	228													
CHARTERM	1	0000B6	640														
CHARTHRU	4	0002EE	230	224													
CHKMIN	4	00029A	205	200													
CNVRTIPT	4	0003BC	279	271													
CONVRTN	2	000118	87	83													
D	1	0000B7	638	261	379	405	457	479	520								
DDR	1	0000D0	660	425													
DECIN	4	0002BE	216	207	212												
DECLOCK	4	0002C2	217	220													
DECLOOP	2	0002EA	233	238													
DECMVC	6	00039A	609	247													
DECPLC	4	000710	566	552													
DIVIN	2	000348	337	334													
DOUBEXP	6	0002BE	188	185													
EDITOR	2	000548	421	416													
EFMT	1	0000D8	652	123	151	361	373	400	412	451							
EMVOOK	2	000E5A	503	481	488	498											
ENTRY	4	000044	23	95	127	365											
EXIT	4	0000FC	74	115	117	163	166	169	172	346	355	606					
EXP	4	000BA5	644	404													
EXPCONV	4	000268	192	187													
EXPLFT	6	0007B8	614	474	483	547	574										
EXPNEG	1	0000C4	655	174	253	256	265	333	430	505	535						
EXPNT	1	000BA5	645	147	196	252	267	324	329	440	504	534					
EXPNTDIG	4	000244	182	156													
FFORMAT	2	000AB8	517	452													
FIXC	4	0000D0	20	5													
FIXEXPTH	4	000586	440	427													
FIXLE	4	0000DA	50	48													
FIXRTN	4	0000BE	47	21													
FIXS	4	0000D8	19	4													
FIXST	4	0000E8	71	68													
FIXZERO	4	0000CA	62	57													
FIX1	4	0000C0	4														
FIX2	4	0000C4	5														

SYMBOL	LEN	VALUE	DEFN	REFERENCES												
FLOATC	4	CC0128	94	7												
FLOATS	4	CC0122	93	6												
FLOATI	4	CC00CF	6													
FLOAT2	4	CC00CC	7													
FLCCFDEC	4	CC069E	526	528												
FLTINTAD	4	CC013E	101	98												
FMTERR	4	CC021C	165	131	133	369	371	381								
FNDDECML	2	CC0316	243	236												
FNDEXPNT	4	CC01DC	150	144												
FORMAT	2	CC0B6E	636	129	367											
FPINC	1	CC0B64	634	19	32	35	47	51	67	81	82	93	97	110	114	120	122
				123	126	151	340	343	358	360	361	364	373	383	387	400	412
				451													
FPINMULT	2	CC042A	330	331												
FPINST	2	CC044C	339	203	325	336										
FPIN1	4	CC0C1C	8													
FPIN2	4	CC0C14	9													
FPIN3	4	CC0C18	10													
FPIN4	4	CC0C1C	11													
FPICSW	1	CC0C1C	651	81	126	364										
FPOLYMVC	2	CC0788	601	422	464	515	530	592								
FPOLY1	4	CC0C2C	12													
FPOLY2	4	CC0C24	13													
FPOLY3	4	CC0C28	14													
FPOLY4	4	CC0C2C	15													
FPRECTST	1	CC09AA	646	616	617	619										
FPRC	1	CC0CCC	680	25	37	37	38	39	40	54	55	56	74	105	108	108	113
				116	326	330	335	337	386	389	391	392	392	395	395	421	421
				424	432	434	436	441									
FPR2	1	CC0CC2	681	26	38	50	53	54	63	63	75	327	330	391	431	434	
FPR4	1	CC0CC4	682	27	39	76											
FPR6	1	CC0CC6	683	28	40	77	320	323	323	335	337	345	348	432	619		
FPSAVE	8	CC07C8	626	25	26	27	28	74	75	76	77	622					
FPSTE	4	CC0168	113	111													
FPWRK1	8	CC08CC	628	56	58	60	62	103	104	105	307	308	309	312	313	314	317
				318	320	348	349	350	424	426	436	437	441	442	443	446	447
				449	512	513	514										
FPWRK2	8	CC0BC8	629	186	188	195	294	302	445	448	450	610					
FTYPEMT	4	CC051E	405	401													
FXFLTCCN	8	CC07F8	627	55	58	103											

SYMBOL	LEN	VALUE	DEFN	REFERENCES												
IGNC	1	0000C2C	656	248	258											
IMBEFFIC	1	000CCC	2													
INRCLNC	4	0C045E	348	344												
INSTRCV	2	000574	434	425	429	438										
INSTORE	4	000446	343	341												
INVALCHR	1	0C0CC8	686	162												
INVALFMT	1	0C0CCC	687	165												
INVALNUM	1	0C0CC4	685	168												
INVALPRG	1	0C0C1C	688	171												
IPTCOMN	4	0C018C	125	121												
IPTCE	4	000188	123	11												
IPTCF	4	000184	121	9												
IPTIND	1	000865	635	80	174	213	248	253	256	258	265	321	333	394	430	466	505
				535	591												
IPTNEG	1	0000C8C	654	213	321												
IPTSE	4	0C0184	122	10													
IPTSF	4	000170	120	8													
LASTCHAR	4	000272	221	218													
LDFPCLT	4	000400	386	384													
LOADRTN	4	000114	88	84	89												
LOOKEXPA	4	000104	143	146													
MOR	1	0000C2C	659	429													
MVCCHAR	6	000744	612	605													
MVCFMT	6	000746	611	138	298												
NEGEXPNT	4	000228	174	158	160												
NOBICIN	4	0004C6	320	310	315												
NODEC	2	000310	240	232	234												
NUMBERR	4	000218	168	59	61	281	291	540									
NUMBLCCP	2	000362	270	277													
NUMNEGIN	4	000284	214	210													
ONE	8	000810	630	326													
OUTCHAR	24	000894	641	449	450	459	477	525	531	549	576						
OUTCOMN	4	000480	363	359													
OUTCE	4	0004E8	361	15													
OUTCF	4	000488	359	13													
OUTNEG	1	000C1C	657	394	466	591											
OUTSE	4	000484	360	14													
OUTSF	4	000470	358	12													
OUTTERM	1	000841	642														
OUTZERO	3	000842	643	410													

SYMBOL	LEN	VALUE	DEFN	REFERENCES													
PACK	6	000?AC	61C	293	301												
PRUGERR	4	0002?C	171	494	557	583											
RETCODE	4	000116	89	41	162	165	168	171									
ROO	1	000000	663	23	30	66	78	84	109	137	244	246	292	300	339	408	409
				417	418	471	482	496	499	522	523	544	559	562	573	585	588
				598	601	604											
ROOREG	1	000040	649	32	67	110	340										
RO1	1	000001	664	23	33	50	53	66	69	70	70	71	99	100	100	101	102
				104	109	113	116	296	339	345	354	386	389	490	501	601	611
				612	616	621	622	622									
RO1REG	1	000020	650	35	47	97	383										
RO2	1	000002	665	20	21	44	62	65	65	71	94	95	101	102	106	106	125
				127	135	135	136	137	138	139	139	146	198	202	205	349	353
				354	363	365	376	376	377	378	380	398	399	405	407	423	423
				435	435	440	518	520	521	568	570	602	602	603	604	605	
RO3	1	000003	666	24	29	30	33	36	142	143	145	145	150	152	152	153	155
				157	159	175	176	176	177	179	182	184	186	188	189	190	190
				192	193	195	196	216	217	219	219	222	233	243	244	270	280
				352	353	372	375	378	379	380	397	399	403	404	406	414	415
				417	419	453	453	454	456	458	471	472	472	474	475	476	478
				478	479	480	480	482	483	484	485	491	504	507	507	508	509
				509	512	517	517	518	519	521	522	523	539	542			
RO4	1	000004	667	221	223	230	233	235	237	237	245	269	270	272	274	276	276
				279	280	282	282	286	288	290	292	293	296	297	298	300	301
				328	328	329	331	406	407	408	409	410	411	411	415	418	470
				475	475	485	485	490	491	491	492	493	495	501	511	514	541
				542	543	554	554	555	556	558	566	566	580	580	581	582	584
				593	594	596	596	598	599	609	609	613	613	614			
RO5	1	000005	668	222	223	225	227	227	240	240	243	245	246	247	250	250	257
				262	263	263	266	266	267	302	303	477	487	489	503	508	531
				532	538	539	543	544	545	545	547	548	554	566	576	577	579
RO6	1	000006	669	251	251	252	255	255	257	260	260	261	262	283	283	284	295
				306	307	311	312	316	317	443	444	445	459	460	460	461	463
				476	476	484	503	525	526	526	527	529	532	548	548	550	553
				555	559	560	560	562	563	567	575	577	579	581	585	586	586
				588	589	614											
RO7	1	000007	670	284	294	303	307	312	317	443	446	484	487	489	492	496	497
				497	499	500	533	533	534	537	537	538	549	550	553	567	568
				570	571	571	573	574	575	575	580						
RO8	1	000008	671	49	112	285	286	288	289	290	305	306	342	385	402	403	455

SYMBOL	LEN	VALUE	DEFN	REFERENCES													
				456	457	458	618	620	623								
R09	1	0C0CC9	672														
R10	1	CC0CCA	673														
R11	1	CC0CCB	674														
R12	1	CC0CCC	675														
R13	1	CC0CCD	676	29													
R14	1	CC0CCE	677	79	85	90	129	367									
R15	1	CC0CCF	678	3	24	78	87	88	433								
SAVER	4	CC0B2C	632	20	23	24	70	78	94	100	125	363					
SECHALF	2	CC01CC	3C0	287													
SETEXP	4	CC01D4	147	140													
SETSIGN	2	CC07B2	598	595													
SHIFTL	4	CC0284	199	202													
SIGLECPE	4	CC05F4	460	462													
SIGNCUT	4	CC0766	591	551	561	564	572	578	587								
SIGNCUTL	4	CC0772	594	597													
SIGNSCAN	4	CC02BC	198	148													
SINGLE	1	CC0C8C	648	19	51	93	114	120	122	343	358	360	387				
TEN	8	CC0B18	631	327	431												
TENNINE	4	CC0B6C	633	295	444												
TSTFFR	4	CC07BE	616	49	112	342	385										
TSTSIGN	2	CC034E	263	259													
W	1	CC0B66	637	130	132	136	198	205	206	231	368	370	377	398	454	519	603
ZFILL	6	CC07B2	613	419	500	563	589										

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

```
TIME                                    EXTERNAL SYMBOL DICTIONARY                                    PAGE    1

SYMBOL   TYPE ID  ADDR  LENGTH LD ID                                                                  14 SEP 70

INTIME    SD  01 000000 0001C6
OUTIME    LD     000078         01
```

```
TIME    INTIME AND OUTIME - PROVIDE TIMER SERVICES TO FORTRAN                                    PAGE    1

 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          14 SEP 70

                                  2 ************************************************************************ TIMER
                                  3 *                                                                      * TIMER
                                  4 *              THIS SUBROUTINE IS USED TO PROVIDE FORTRAN WITH         * TIMER
                                  5 *        TIMER SERVICES.  THIS ALLOWS THE FORTRAN PROGRAM ACCESS TO    * TIMER
                                  6 *        THE TIMER IN A MANNER SIMILAR TO ASSEMBLY LANGUAGE.           * TIMER
                                  7 *        THIS ROUTINE CAN BE CONSIDERED SIMILAR TO THE ORIGINAL        * TIMER
                                  8 *        INTIME ROUTINE, BUT IT DIFFERS IN SEVERAL IMPORTANT WAYS.     * TIMER
                                  9 *        THESE ARE:                                                    * TIMER
                                 10 *              A SEPARATE ENTRY POINT IS PROVIDED FOR THE 'START'      * TIMER
                                 11 *              TIME AND THE 'END' TIME.  HENCE, A PROGRAM CAN          * TIMER
                                 12 *              GET A NEW START TIME AT ANY TIME, AND WHEN THE END      * TIMER
                                 13 *              IS GOTTEN, THE TIMER CONTINUES TO COUNT THE TIME        * TIMER
                                 14 *              SINCE THE LAST START TIME.                              * TIMER
                                 15 *              A SECOND PARAMETER CAN BE SPECIFIED TO GIVE THE AMOUNT  * TIMER
                                 16 *              OF TIME AFTER WHICH THE PROGRAM SHOULD BE ABENDED.      * TIMER
                                 17 *              IF THIS AMOUNT IS NOT SPECIFIED, THE PROGRAM WILL       * TIMER
                                 18 *              BE ABENDED AFTER 1 HOUR.  THIS OPTIONAL AMOUNT IS       * TIMER
                                 19 *              SPECIFIED IN HUNDREDTH OF AN HOUR.  A SPECIFICATION     * TIMER
                                 20 *              OF ZERO OR LESS IS THE SAME AS IF NO AMOUNT WAS GIVEN.  * TIMER
                                 21 *              THE NAME OF THE SUBROUTINE TO CALL FOR THE START        * TIMER
                                 22 *              TIME IS 'INTIME' AND THE NAME OF THE SUBROUTINE         * TIMER
                                 23 *              TO CALL FOR THE STOP TIME IS 'OUTIME'.                  * TIMER
                                 24 *              ALL PARAMETERS ARE INTEGER*4 VARIABLES WITH THE TIME    * TIMER
                                 25 *              ALWAYS SPECIFIED IN UNITS OF 1/100 OF A SECOND.         * TIMER
                                 26 *                                                                      * TIMER
                                 27 *              THIS ROUTINE IS COMPLETELY SERIALLY REUSABLE.           * TIMER
                                 28 *        HOWEVER, A SECOND CALL TO INTIME WILL SET A NEW TIME          * TIMER
                                 29 *        INTERVAL AND PROVIDE A NEW START TIME.  A SECOND CALL         * TIMER
                                 30 *        TO OUTIME WILL GIVE THE TOTAL TIME SINCE THE LAST CALL        * TIMER
                                 31 *        TO INTIME, NOT SINCE THE LAST CALL TO OUTIME.  THIS           * TIMER
                                 32 *        ROUTINE IS NOT REENTRANT.                                     * TIMER
                                 33 *                                                                      * TIMER
                                 34 ************************************************************************ TIMER
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  14 SEP 70
                                    36 INTIME  PRIME BR=R12,SA=SAVEAREA      CALL INTIME(START,LIMIT)        TIMER
000000                              37+INTIME  START 0
000000 47F0 F00C            00000   38+        B     12(0,15) BRANCH AROUND ID
000004 C6                           39+        DC    AL1(6)
000005 C9D5E3C9D4C5                 40+        DC    CL6'INTIME' IDENTIFIER
00000B CC
00000C 90EC D00C            0000C   41+        STM   14,12,12(13) SAVE REGISTERS
000010 C5C0                         42+        BALR  R12,0 CREATE BASE REGISTER
000012                              43+        USING *,R12
000012 1830                         44+        LR    3,0 SAVE -0-
000014 1821                         45+        LR    2,1 SAVE PARAMETER REGISTER
000016 411C C0F6            00108   46+        LA    1,SAVEAREA GET ADDRESS OF SAVE AREA
00001A 501D 0008            00008   47+        ST    1,8(13) STORE FORWARD POINTER
00001E 50D1 C004            00004   48+        ST    13,4(1) STORE BACKWARD POINTER
000022 18D1                         49+        LR    13,1 SET UP REG FOR SAVE AREA
000024 1812                         50+        LR    1,2 RESTORE PARAMETER REGISTER
000026 1803                         51+        LR    0,3 RESTORE -0-
                                    52 *                                                                    TIMER
000028 5821 C000            00000   53         L     R02,0(R01)          GET START TIME ADDRESS          TIMER
00002C 5840 C18A            0019C   54         L     R04,HOUR            LOAD 1 HR TIME IN R04           TIMER
000030 9180 1000      00000         55         TM    0(R01),X'80'        CHECK FOR NO TIME LIMIT         TIMER
000034 4710 C036            00048   56         BC    SETIME              IF NONE, SET TO 1 HOUR          TIMER
000038 5831 C004            00004   57         L     R03,4(R01)          LOAD ADDRESS OF TIME LIMIT      TIMER
00003C 5833 0000            00000   58         L     R03,0(R03)          LOAD THE TIME LIMIT             TIMER
                                    59 *                                                                    TIMER
00004C 1233                         60         LTR   R03,R03             TEST FOR A ZERO LIMIT           TIMER
000042 470C C036            00048   61         BNH   SETIME              IF ZERO OR NEG, USE 1 HOUR      TIMER
000046 1843                         62         LR    R04,R03             SET LIMIT AS SPECIFIED          TIMER
                                    63 *                                                                    TIMER
000048 5040 C186            00198   64 SETIME  ST    R04,TIMEVAL         STORE THE TIME LIMIT            TIMER
                                    65 *                                                                    TIMER
                                    66         STIMER TASK,LIMIT,BINTVL=TIMEVAL    SET THE TIMER          TIMER
00004C 4110 C186            00198   67+        LA    1,TIMEVAL LOAD PARAMETER REG 1
000050 41C0 C08A            000CC   68+        LA    0,LIMIT LOAD PARAMETER REG 0
000054 41F0 C010            00010   69+        LA    14,16(0,0) LOAD FLAG BYTE
000058 89EC 0018            00018   70+        SLL   14,24(0) SHIFT TO HI-ORDER BYTE
00005C 16CE                         71+        OR    0,14 AND PACK WITHEXIT ADDR
00005E CA2F                         72+        SVC   47 ISSUE STIMER SVC
                                    73 *                                                                    TIMER
000060 1755                         74         XR    R05,R05             ZERO THE REGISTER               TIMER
```

```
TIME     INTIME AND OUTIME - PROVIDE TIMER SERVICES TO FORTRAN                                                    PAGE    3

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                        14 SEP 70

000062 5052 C00C             00000    75          ST    R05,0(R02)                SET THE START TIME (ZERO)        TIMER
                                      76 *                                                                          TIMER
                                      77 INRET    TERME RC=0                      RETURN TO FORTRAN                TIMER
000066 58DD 0004             00004    78+INRET    L     13,4(13) RESTORE REGISTER -13-
00006A 98EC D00C             0000C    79+         LM    14,12,12(13) RESTORE THE REGISTERS
00006E 92FF D00C       C000C          80+         MVI   12(13),X'FF' SET RETURN INDICATION
000072 41FC C000             00000    81+         LA    15,0(0,0) LOAD RETURN CODE
000076 07FE                           82+         BR    14 RETURN
                                      83 *                                                                          TIMER
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

                                    85 OUTIME   PRIME BR=R12,SA=SAVEAREA,CSECT=NO   CALL OUTIME(END)          TIMER
000078 47F0 F000            0000C   86+OUTIME   B     12(0,15) BRANCH AROUND ID
00007C 06                           87+         DC    AL1(6)
00007D D6E4E3C9D4C5                 88+         DC    CL6'OUTIME' IDENTIFIER
000083 00
000084 90FC D000            0000C   89+         STM   14,12,12(13) SAVE REGISTERS
000088 05C0                         90+         BALR  R12,0 CREATE BASE REGISTER
00008A                              91+         USING *,R12
00008A 1830                         92+         LR    3,0 SAVE -0-
00008C 1821                         93+         LR    2,1 SAVE PARAMETER REGISTER
00008E 411C 007E            00108   94+         LA    1,SAVEAREA GET ADDRESS OF SAVE AREA
000092 501D 0008            00008   95+         ST    1,8(13) STORE FORWARD POINTER
000096 50D1 0004            00004   96+         ST    13,4(1) STORE BACKWARD POINTER
00009A 18D1                         97+         LR    13,1 SET UP REG FOR SAVE AREA
00009C 1812                         98+         LR    1,2 RESTORE PARAMETER REGISTER
00009E 1803                         99+         LR    0,3 RESTORE -0-
                                   100          ENTRY OUTIME                    SPECIFY THIS AS AN ENTRY POINT  TIMER
                                   101 *                                                                       TIMER
0000A0 5821 0000            00000  102          L     R02,0(R01)                LOAD ADDRESS OF END TIME        TIMER
                                   103 *                                                                       TIMER
                                   104          TTIMER ,                        GET REMAINING TIME              TIMER
0000A4 1B11                        105+         SR    1,1 NO CANCELLATION
0000A6 0A2E                        106+         SVC   46 ISSUE TTIMER SVC
                                   107 *                                                                       TIMER
0000A8 8000 0020            00020  108          SRDL  R00,32                    SHIFT REMAINDER FOR DIVISION    TIMER
0000AC 5D00 0116            001A0  109          D     R00,TUT100                CONVERT TO 1/100 OF SECONDS     TIMER
                                   110 *                                                                       TIMER
0000B0 5830 010E            00198  111          L     R03,TIMEVAL               LOAD THE ORIGINAL AMOUNT        TIMER
0000B4 1B31                        112          SR    R03,R01                   SUBTRACT AMOUNT LEFT            TIMER
0000B6 5032 0000            00000  113          ST    R03,0(R02)                STORE THE ENDING TIME           TIMER
                                   114 *                                                                       TIMER
                                   115 OUTRET   TERME RC=0                      RETURN TO FORTRAN               TIMER
0000BA 58DD 0004            00004  116+OUTRET   L     13,4(13) RESTORE REGISTER -13-
0000BE 98EC D00C            0000C  117+         LM    14,12,12(13) RESTORE THE REGISTERS
0000C2 92FF D00C      0000C        118+         MVI   12(13),X'FF' SET RETURN INDICATION
0000C6 41F0 0000            00000  119+         LA    15,0(0,0) LOAD RETURN CODE
0000CA 07FE                        120+         BR    14 RETURN
                                   121 *                                                                       TIMER
```

```
                                    123 LIMIT    PRIME BR=R12,SA=TIMEAREA,CSECT=NO   ENTRY WHEN TIME RUNS OUT   TIMER
000000 47F0 F00A            0000A   124+LIMIT    B     10(0,15) BRANCH AROUND ID
000004 05                           125+         DC    AL1(5)
000005 D3D9D4C9E3                   126+         DC    CL5'LIMIT' IDENTIFIER
00000A 90EC D00C            00000   127+         STM   14,12,12(13) SAVE REGISTERS
00000E 05C0                         128+         BALR  R12,0 CREATE BASE REGISTER
000010                              129+         USING *,R12
000010 183C                         130+         LR    3,0 SAVE -0-
000012 1821                         131+         LR    2,1 SAVE PARAMETER REGISTER
000014 411C C074            0015C   132+         LA    1,TIMEAREA GET ADDRESS OF SAVE AREA
000018 501D C008            00008   133+         ST    1,8(13) STORE FORWARD POINTER
00001C 50D1 C004            00004   134+         ST    13,4(1) STORE BACKWARD POINTER
000020 18D1                         135+         LR    13,1 SET UP REG FOR SAVE AREA
000022 1812                         136+         LR    1,2 RESTORE PARAMETER REGISTER
000024 1803                         137+         LR    0,3 RESTORE -0-
                                    138 *                                                                  TIMER
                                    139          ABORT X'322'              ABEND WITH TIME EXCESSION CODE   TIMER
000026 C700                         140+         CNOP  0,4 *               ALIGN THE COMPLETION CODE
000028 47F0 C020            00FFC   141+         B     *+8 *               BRANCH AROUND IT
00002C 00                           142+         DC    X'00' *             DUMP/STEP INDICATOR
00002D 322000                       143+         DC    AL3(3284992) *      SYSTEM/USER COMPLETION CODE
000030 581C C01C            00FF8   144+         L     1,*-4 *             LOAD COMPLETION CODE
000034 0A0D                         145+         SVC   13 *                SVC TO ABEND ROUTINE
                                    146 *                                                                  TIMER
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    14 SEP 70

000102 000000000000
000108 000000000000000000          148 SAVEAREA DC    9D'0'          OS SAVEAREA FOR INTIME & OUTIME TIMER
000150 000000000000000000          149 TIMEAREA DC    9D'0'          OS SAVEAREA FOR TIME LIMIT UP   TIMER
                                   150 *                                                              TIMER
000198 00000000                    151 TIMEVAL  DC    F'0'           TIME LIMIT SET BY INTIME        TIMER
                                   152 *                                                              TIMER
00019C 00057E40                    153 HOUR     DC    A(3600*100)    1 HOUR - 3600 SECONDS           TIMER
                                   154 *                                                              TIMER
0001A0 00000180                    155 TUT100   DC    A(10000/26)    CONVERSION FROM TU TO 1/100 SEC TIMER
                                   156 *                                                              TIMER
                                   157          REGS  STD            REGISTER EQUATES                TIMER
000000                             158+R00      EQU   0
000001                             159+R01      EQU   1
000002                             160+R02      EQU   2
000003                             161+R03      EQU   3
000004                             162+R04      EQU   4
000005                             163+R05      EQU   5
000006                             164+R06      EQU   6
000007                             165+R07      EQU   7
000008                             166+R08      EQU   8
000009                             167+R09      EQU   9
00000A                             168+R10      EQU   10
00000B                             169+R11      EQU   11
00000C                             170+R12      EQU   12
00000D                             171+R13      EQU   13
00000E                             172+R14      EQU   14
00000F                             173+R15      EQU   15
                                   174 *                                                              TIMER
0001A8                             175 LITERALS LTORG                PROGRAM LITERALS                TIMER
                                   176 *                                                              TIMER
                                   177 PATCH    PATCH 30             PROGRAM PATCH AREA              TIMER
0001A8                             178+         DC    0H'0'
0001A8                             179+PATCH    DS    0CL30
0001A8 0000                        180+         DC    S(*)
0001AA 000E                        181+         DC    S(*)
0001AC 0000                        182+         DC    S(*)
0001AE 00D2                        183+         DC    S(*)
0001B0 00D4                        184+         DC    S(*)
0001B2 00D6                        185+         DC    S(*)
0001B4 00D8                        186+         DC    S(*)
```

```
TIME     INTIME AND CUTIME - PROVIDE TIMER SERVICES TO FORTRAN                                    PAGE    7

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                       14 SEP 70

0001E6 CODA                          187+          DC     S(*)
0001E8 CODC                          188+          DC     S(*)
0001EA CODE                          189+          DC     S(*)
0001EC COEC                          190+          DC     S(*)
0001EE COE2                          191+          DC     S(*)
0001C0 COE4                          192+          DC     S(*)
0001C2 COE6                          193+          DC     S(*)
0001C4 COE8                          194+          DC     S(*)
                                     195 *                                                          TIMER
                                     196           END                                              TIMER
```

SYMBOL	LEN	VALUE	DEFN	REFERENCES								
HOUR	4	000190	153	54								
INRET	4	0000E6	78									
INTIME	1	000000	37									
LIMIT	4	000000	124	68								
LITERALS	1	0001A8	175									
OUTIME	4	0000E8	86	100								
OUTRET	4	0000FA	116									
PATCH	30	0001A8	179									
R00	1	000000	158	108	109							
R01	1	000001	159	53	55	57	102	112				
R02	1	000002	160	53	75	102	113					
R03	1	000003	161	57	58	58	60	60	62	111	112	113
R04	1	000004	162	54	62	64						
R05	1	000005	163	74	74	75						
R06	1	000006	164									
R07	1	000007	165									
R08	1	000008	166									
R09	1	000009	167									
R10	1	00000A	168									
R11	1	00000B	169									
R12	1	00000C	170	42	43	90	91	128	129			
R13	1	00000D	171									
R14	1	00000E	172									
R15	1	00000F	173									
SAVEAREA	8	000108	148	46	94							
SETIME	4	0000C8	64	56	61							
TIMEAREA	8	00015C	149	132								
TIMEVAL	4	000198	151	64	67	111						
TUTICC	4	0001AC	155	109								

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

```
F12S-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LET,LIST,MAP,SIZE=(240K,100K)
           VARIABLE OPTIONS USED - SIZE=(245760,102400)
IEWCCCC                INCLUDE OBJLIB(IKMSTATS)
IEWCCCC                ENTRY MAIN
```

MODULE MAP

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
GDFIT	C0	FBA								
ANFARM	FC0	1722								
MCCC	26E8	E6E								
TREND	3558	C9E								
MAIN	41F8	532								
FRCMAN	4730	E2A								
RCFARM	5560	14DE								
DCTEST	6A40	3AC								
MEDIAN	6DF0	152E								
CHINCP	8320	11C2								
CHIRXC	94E8	D2A								
DURBIN	A218	D22								
KRSKAL	AF40	F62								
MANWHT	BDA8	E02								
TAL	CBB0	E26								
RFC	D9D8	138A								
WILCCX	EC68	1066								
MCNMAR	FCC0	8C6								
OFLCW	10998	106								
RCFMT	10AA0	3D8								
RCDATA	10E78	16BC								
PRLBLS	12528	4BE								
RANKER	129E8	AD2								
ANCTR	134C0	1FC								
CNTFMT	136C0	C0								
HEAC	13780	1B4								
SCNTST	13938	9B6								
WRDATA	142F0	2A4								
PCIS	14598	264								
DECODE	148C0	774								
ECDAC	14F78	1D8								
IHDFFFIO	15150	910								
CDTR	15A60	B92								
BINCM	165F8	416								
INTIME	16A10	1C6								
			CUTIME	16A88						
DLCAM	16BC8	24E								
SETIC	16E28	104								
TRANS	16F30	A8								
IHCLEXP	16FD8	288								
			DEXP	16FD8						
IHCLLCG	17260	200								

NAME	ORIGIN	LENGTH
IHCLSORT	17460	158
IHCECOMH	175C0	F31
IHCCOMH2	184F8	560
IHCFCVTH	18A68	1195
IHCEFATH	19CC0	512
IHCEFIOS	1A118	127C
IHCERRM	1B398	58C
IHCUCPT	1B958	300
IHCETRCH	1BC58	28E
IHCLATBL	1BEE8	208
IHCFCPT	1C0F0	304
IHCIBERH	1C3F8	C8
CONTBL	1C4C0	A8
DIST	1C568	1C
FLAG	1C578	AC
FORM	1C628	3DC
INFARM	1C9F8	37C
HYPTST	1CD78	5C
PRETTY	1CDD8	1CC
TYPES	1CF98	CC
UNITS	1D058	24
USER	1D080	B8
WORKA	1D138	4FC0

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
DLOG10	17260	DLOG	17278				
DSORT	17460						
IBCOM#	175C0	FDIOCS#	1767C	INTSWTCH	184DE		
SEQCASD	18780						
ADCON#	18A68	FCVAOUTP	18B12	FCVLOUTP	18BA2	FCVZOUTP	18CF2
FCVIOUTP	19098	FCVECUTP	1959A	FCVCOUTP	19784	INT6SWCH	19A9B
ARITH#	19C00	ADJSWTCH	19F6C				
FIOCS#	1A118	FIOCSBEP	1A11E				
ERRMON	1B398	IHCERRE	1B3B0				
IHCTRCH	1BC58	ERRTRA	1BC60				
ERRSET	1C0F0	ERRSAV	1C2EA	ERRSTR	1C310		
IBERH#	1C3F8						

ENTRY ADDRESS 41F8
TOTAL LENGTH 22CF8

****IKMSTATS DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET

APPENDIX III

SAMPLE OUTPUT

```
PARAMETER ANALYSIS

TITLE,  THE MC NEMAR TEST - DATA FROM CONOVER, PAGE 208

DATA,BIVAR

LABEL=BEFORE,AFTER

CODING,4C

TESTS,.05,MCNEMAR

END

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF A BIVARIATE SAMPLE
THE SAMPLE IS OF SIZE     0

IKMS161  END OF DATA WILL DETERMINE SAMPLE SIZES.

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE MC NEMAR TEST

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

VARIATE  1 IS CALLED BEFORE
VARIATE  2 IS CALLED AFTER

INPUT OF DATA

THERE ARE 4C DATA ITEMS PER READ
```

INPUT IS FROM UNIT 1

INPUT FORMAT STATEMENT
(4CF1.0)

END OF INPUT OF DATA

THE MC NEMAR TEST FOR SIGNIFICANCE OF CHANGE

HO: P(BEFORE = 0) .EQ. P(AFTER = 0)

VERSUS

H1: P(BEFORE = 0) .NE. P(AFTER = 0)

THE DATA IS DICHOTOMIZED AT 0.50000

THE SAMPLE SIZE IS 100

A = 63 B = 21 C = 4 D = 12

THE TEST STATISTIC IS 11.56

AT ALPHA = 0.050, HO IS REJECTED.

ALPHA HAT = 0.001

END OF THE MC NEMAR TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE,  TEST FOR TREND - DATA FROM CONOVER, PAGE 397

DATA,1RS

CODING,1

LABEL=RAIN

TESTS,.1C,TREND

END OF PARAMETERS


NO PARAMETER SYNTAX ERRORS DETECTED.


THE DATA SUPPLIED CONSISTS OF 1 RANDOM SAMPLE
THE SAMPLE IS OF SIZE     0


IKMS161  END OF DATA WILL DETERMINE SAMPLE SIZES.


THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

TEST OF TREND USING SPEARMAN'S RHO


THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

THE SAMPLE IS CALLED RAIN


INPUT OF DATA

THERE IS 1 DATA ITEM PER READ
INPUT IS FROM UNIT  1
```

INPUT FORMAT STATEMENT
(F4.2)

END OF INPUT OF DATA

TEST OF TREND USING SPEARMAN'S RANK CORRELATION COEFFICIENT

HO: THERE IS NO SIGNIFICANT TREND IN THE DATA

VERSUS

H1: THERE IS A DEFINITE TREND IN THE OBSERVED DATA

THE SAMPLE IS OF SIZE 19

THERE WAS A TOTAL OF 2 TIES IN 1 PLACES

THE CORRELATION COEFFICIENT (RHO) IS -0.247

AT ALPHA = 0.100, HO IS ACCEPTED.

ALPHA HAT = 0.295

IKMS901 ***** BECAUSE OF SAMPLE SIZES, ALPHA HAT MAY NOT BE CORRECT. VERIFY WITH TABLES. *****

END OF THE TEST OF TREND

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE,  MEDIAN TEST DATA FROM CONOVER, PAGE 268

DATA,KRS,4,9,10,7,8

LABEL=METHOD1,METHOD2,METHOD3,METHOD4

TEST,.10,MEDIAN

CODING,17,9,10,7,8

END

NO PARAMETER SYNTAX ERRORS DETECTED.


THE DATA SUPPLIED CONSISTS OF  4 RANDOM SAMPLES
THE SAMPLES ARE OF SIZES               9        10         7         8


THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE MEDIAN TEST


THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

SAMPLE  1 IS CALLED METHOD1
SAMPLE  2 IS CALLED METHOD2
SAMPLE  3 IS CALLED METHOD3
SAMPLE  4 IS CALLED METHOD4


INPUT OF DATA

THERE ARE 17 DATA ITEMS PER READ
INPUT IS FROM UNIT  1
```

INPUT FORMAT STATEMENT
(17F3.0)

END OF INPUT OF DATA

THE MEDIAN TEST

HO: THE MEDIANS OF ALL 4 SAMPLES ARE THE SAME

VERSUS

H1: THE MEDIANS OF AT LEAST 2 SAMPLES ARE NOT THE SAME

THERE IS A TOTAL OF 34 OBSERVATIONS WITH THE FOLLOWING BREAKDOWN:

SAMPLE	NUMBER	LESS THAN OR EQUAL	GREATER THAN	TOTAL SIZE
METHOD1	1	3	6	9
METHOD2	2	7	3	10
METHOD3	3	0	7	7
METHOD4	4	8	0	8

THE GRAND MEDIAN IS 89.000000

THE TEST STATISTIC IS 17.765 WITH 3 DEGREES OF FREEDOM

AT ALPHA = 0.100, HO IS REJECTED.

ALPHA HAT = 0.000

END OF THE MEDIAN TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE, EXTENDED MEDIAN TEST - DATA FROM CONOVER, PAGE 273

DATA,RCB,6,4,3

BLOCK=FIELDS,TREAT=FERTS,REP=SEEDS

CODING,4,BLOCK,REP

TEST,.20,MEDIAN

END

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF A RANDOMIZED COMPLETE BLOCK, WITH  6 BLOCKS,  4 TREATMENTS, AND  3 REPLICATIONS

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE MEDIAN TEST

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

BLOCKS ARE CALLED FIELDS
TREATMENTS ARE CALLED FERTS
REPLICATIONS ARE CALLED SEEDS

INPUT OF DATA

THERE ARE  4 DATA ITEMS PER READ
INPUT IS FROM UNIT  1

INPUT FORMAT STATEMENT
(2I1,4F3.1)
```

KANSAS STATE UNIVERSITY
DEPT. OF STATISTICS & COMPUTER SCIENCE
EXTENDED MEDIAN TEST - DATA FROM CONOVER, PAGE 273

DATA IS CODED BY BLOCKS REPLICATIONS

END OF INPUT OF DATA

THE MEDIAN TEST

THERE IS A TOTAL OF 72 OBSERVATIONS

HO: THE MEDIANS OF ALL 6 FIELDS ARE THE SAME

VERSUS

H1: THE MEDIANS OF AT LEAST 2 FIELDS ARE NOT THE SAME

THE TEST STATISTIC IS 5.3333 WITH 5 DEGREES OF FREEDOM

AT ALPHA = 0.200, HO IS ACCEPTED.

ALPHA HAT = 0.377

HO: THE MEDIANS OF ALL 4 FERTS ARE THE SAME

VERSUS

H1: THE MEDIANS OF AT LEAST 2 FERTS ARE NOT THE SAME

THE TEST STATISTIC IS 14.063 WITH 3 DEGREES OF FREEDOM

AT ALPHA = 0.200, HO IS REJECTED.

ALPHA HAT = 0.003

HO: THE MEDIANS OF ALL 3 SEEDS ARE THE SAME

VERSUS

H1: THE MEDIANS OF AT LEAST 2 SEEDS ARE NOT THE SAME

KANSAS STATE UNIVERSITY
DEPT. OF STATISTICS & COMPUTER SCIENCE
EXTENDED MEDIAN TEST - DATA FROM CONOVER, PAGE 273

THE TEST STATISTIC IS 34.875 WITH 2 DEGREES OF FREEDOM

AT ALPHA = 0.200, HO IS REJECTED.

ALPHA HAT = 0.000

END OF THE MEDIAN TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE, CHI-SQUARE TEST OF PROBABILITIES - 2 X 2 - CONOVER, PAGE 239, NO 5

LABEL=CITIZENS,ALIENS

DATA,2RS

CODING,2C,2C,2O

TESTS,,.05,CHIPROB

CLASSES,COLUMN,2,3.499

END

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF 2 RANDOM SAMPLES
THE SAMPLES ARE OF SIZES                0         0

IKMS15I  EQUAL SAMPLE SIZES ASSUMED.

IKMS16I  END OF DATA WILL DETERMINE SAMPLE SIZES.

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE CHI SQUARE TEST OF DIFFERENCES -   2 BY  2

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

SAMPLE  1 IS CALLED CITIZENS
SAMPLE  2 IS CALLED ALIENS
```

INPUT OF DATA

THERE ARE 20 DATA ITEMS PER READ
INPUT IS FROM UNIT 1

INPUT FORMAT STATEMENT
(20F3.2)

END OF INPUT OF DATA

THE CHI-SQUARE TEST FOR DIFFERENCES IN PROBABILITIES - 2 BY 2

HO: THE PROBABILITY OF APPEARRING IN ANY COLUMN IS THE SAME FOR ALL SAMPLES

VERSUS

H1: THE PROBABILITY OF APPEARRING IN ANY COLUMN IS DIFFERENT FOR AT LEAST 2 SAMPLES

THE SAMPLES ARE TO BE DICHOTOMIZED INTO 2 CLASSES AS FOLLOWS:

-.1000E 76 TO 3.499 3.499 TO 0.1000E 76

THE SAMPLES WERE DISTRIBUTED AS FOLLOWS:

SAMPLE	NUMBER	SIZE	CLASSES	1	2
CITIZENS	1	20		11	9
ALIENS	2	20		6	14

THE TEST STATISTIC IS 2.5575 WITH 1 DEGREES OF FREEDOM

AT ALPHA = 0.050, HO IS ACCEPTED.

ALPHA HAT = 0.110

END OF THE CHI-SQUARE TEST FOR PROBABILITY DIFFERENCES

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE,  GOODNESS OF FIT TEST - NORMAL - FROM CONOVER, PAGE 296

DATA,IRS

SOURCE,1,EXTERNAL,15,RAW

CODING,2C,2C

TEST,.05,CHIGOF

DISTRIB,NORMAL,30,100

CLASSES,COLUMNS,4,23.255,30,36.745

END

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF 1 RANDOM SAMPLE
THE SAMPLE IS OF SIZE     0

IKMS161  END OF DATA WILL DETERMINE SAMPLE SIZES.

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE CHI-SQUARE GOODNESS OF FIT TEST

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

NO LABELS PROVIDED

INPUT OF DATA
```

THERE ARE 20 DATA ITEMS PER READ
INPUT IS FROM UNIT 1

INPUT FORMAT STATEMENT
(20F3.1)

THE DATA IS WRITTEN OUT ON UNIT 15 AS IT COMES IN

END OF INPUT OF DATA

THE CHI-SQUARE GOODNESS OF FIT TEST

HO: THE SAMPLE COMES FROM A NORMAL POPULATION WITH MEAN 30.00 AND VARIANCE 100.0

VERSUS

H1: THE SAMPLE DOES NOT COME FROM A POPULATION AS DESCRIBED IN HO

THE SAMPLE WAS BROKEN DOWN AS FOLLOWS:

CLASS	CLASS INTERVAL	OBSERVED	EXPECTED
1	-.10000E 76 TO 23.26	8	5.000
2	23.26 TO 30.00	4	5.000
3	30.00 TO 36.74	3	5.000
4	36.74 TO 0.10000E 76	5	5.000
TOTAL		20	

THE TEST STATISTIC IS 2.8002 WITH 3 DEGREES OF FREEDOM

AT ALPHA = 0.050, HO IS ACCEPTED.

ALPHA HAT = 0.423

END OF THE CHI-SQUARE GOODNESS OF FIT TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE, GOODNESS OF FIT - UNIFORM - SAME DATA AS PREVIOUS TEST

DATA,1RS

CODING,2C,2C

SOURCE,15

TEST,.10,GOODFIT

DISTRIB,UNIFORM

CLASSES,COLUMNS,4,20,30,40

END

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF 1 RANDOM SAMPLE
THE SAMPLE IS OF SIZE    0

[KMS16]  END OF DATA WILL DETERMINE SAMPLE SIZES.

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE CHI-SQUARE GOODNESS OF FIT TEST

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

NO LABELS PROVIDED

INPUT OF DATA
```

KANSAS STATE UNIVERSITY
DEPT. OF STATISTICS & COMPUTER SCIENCE
GOODNESS OF FIT - UNIFORM - SAME DATA AS PREVIOUS TEST

THERE ARE 2C DATA ITEMS PER READ
INPUT IS FROM UNIT 15

DATA IS UNFORMATTED

END OF INPUT OF DATA

THE CHI-SQUARE GOODNESS OF FIT TEST

HO: THE SAMPLE COMES FROM A UNIFORMLY DISTRIBUTED POPULATION

VERSUS

H1: THE SAMPLE DOES NOT COME FROM A POPULATION AS DESCRIBED IN HO

THE SAMPLE WAS BROKEN DOWN AS FOLLOWS:

CLASS	CLASS INTERVAL	OBSERVED	EXPECTED
1	-.1000E 76 TO 20.00	6	5.000
2	20.00 TO 30.00	6	5.000
3	30.00 TO 40.00	5	5.000
4	40.00 TO 0.1000E 76	3	5.000
TOTAL		20	

THE TEST STATISTIC IS 1.2000 WITH 3 DEGREES OF FREEDOM

AT ALPHA = 0.100, HO IS ACCEPTED.

ALPHA HAT = 0.753

END OF THE CHI-SQUARE GOODNESS OF FIT TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

PARAMETER ANALYSIS

TITLE, TEST OF THE DATA CONVERSION ERROR ROUTINE

DATA,BIVAR,2,4

LABEL=OK,BAC

CODING=2

TESTS,.1C,ALL

SOURCE,1,EXTERNAL,3,RAW

END OF PARAMETERS

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF A BIVARIATE SAMPLE
THE SAMPLE IS OF SIZE 4

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE SIGN TEST

THE MC NEMAR TEST

THE WILCOXON SIGNED RANK TEST

CORRELATION USING SPEARMAN'S RHO

CORRELATION USING KENDALL'S TAU

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

```
VARIATE  1 IS CALLED OK
VARIATE  2 IS CALLED BAD


INPUT OF DATA

THERE ARE  2 DATA ITEMS PER READ
INPUT IS FROM UNIT  1

INPUT FORMAT STATEMENT
(2F3.0)

THE DATA IS WRITTEN OUT ON UNIT  3 AS IT COMES IN

   123.00000        456.00000
   456.00000        123.00000
   654.00000        321.00000
  0.0               111.00000


IKM5201  CONVERSION ERROR OCCURRED IN COLUMN  5
1472K8
     $


END OF INPUT OF DATA


IKM5191  INPUT BYPASSED UNTIL END OF DATA
```

```
PARAMETER ANALYSIS

TITLE,  SOME TESTS PERFORMED ON TWO RANDOM SAMPLES

DATA=2RS,,1C,15

LABEL=FIRST,SECOND

CODING,25,1C,15

CLASSES,COLUMNS,10,100,200,300,400,500,600,700,800,900

TESTS,.10,2-TAIL,M-W,MOOD,K-W

TRANS=SAMPLE

END

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF 2 RANDOM SAMPLES
THE SAMPLES ARE OF SIZES              10        15

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE MANN-WHITNEY TEST

MOOD'S TEST OF VARIANCE

THE KRUSKAL-WALLIS TEST

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

SAMPLE  1 IS CALLED FIRST
SAMPLE  2 IS CALLED SECOND
```

INPUT OF DATA

THERE ARE 25 DATA ITEMS PER READ
INPUT IS FROM UNIT 1

INPUT FORMAT STATEMENT
(25F3.0)

USER TRANSFORMATION IS SPECIFIED AFTER SEPARATION OF DATA

END OF INPUT OF DATA

THE MANN-WHITNEY TEST

HO: E(FIRST) .EQ. E(SECOND)

VERSUS

H1: E(FIRST) .NE. E(SECOND)

FIRST HAS 10 OBSERVATIONS

SECOND HAS 15 OBSERVATIONS

THERE WAS A TOTAL OF 2 TIED OBSERVATIONS IN 1 PLACES

THE TEST STATISTIC IS 72.500

AT ALPHA = 0.100, HO IS ACCEPTED.

ALPHA HAT = 0.890

IKMS901 ***** BECAUSE OF SAMPLE SIZES, ALPHA HAT MAY NOT BE CORRECT. VERIFY WITH TABLES. *****

END OF THE MANN-WHITNEY TEST

MOOD'S TEST OF VARIANCES

HO: VAR(FIRST) .EQ. VAR(SECOND)

VERSUS

H1: VAR(FIRST) .NE. VAR(SECOND)

FIRST HAS 10 OBSERVATIONS

SECOND HAS 15 OBSERVATIONS

THERE WAS A TOTAL OF 2 TIED OBSERVATIONS IN 1 PLACES

THE TEST STATISTIC IS 402.25

AT ALPHA = 0.100, HO IS ACCEPTED.

ALPHA HAT = 0.310

IKMS901 ***** BECAUSE OF SAMPLE SIZES, ALPHA HAT MAY NOT BE CORRECT. VERIFY WITH TABLES. *****

END OF MOOD'S TEST OF VARIANCES

THE KRUSKAL-WALLIS TEST

HO: ALL 2 SAMPLES HAVE THE SAME DISTRIBUTION

VERSUS

H1: AT LEAST 1 SAMPLE YIELDS LARGER OBSERVATIONS THAN THE OTHERS

THERE ARE 2 SAMPLES WITH A TOTAL OF 25 OBSERVATIONS

SAMPLE NUMBER		SIZE	RANK SUM
FIRST	1	10	127.5
SECOND	2	15	197.5

THERE WERE 2 TIED OBSERVATIONS IN 1 PLACES

THE TEST STATISTIC IS 0.19180E-01 WITH 1 DEGREES OF FREEDOM

AT ALPHA = C.100, HO IS ACCEPTED.

ALPHA HAT = 0.890

END OF THE KRUSKAL-WALLIS TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE    SOME TESTS USED ON SEVERAL RANDOM SAMPLES

DATA,KRS,5,10,10,10,10,10

LABEL(1)=FIRST,SECOND,THIRD

LABEL(4)=FOURTH,FIFTH

CODING=5

CLASSES,COLUMNS,6,1,2,4,6,8

TESTS,.15,CHIPROB,MEDIAN,KRUSKAL-WALLIS

TRANS=NONE

END OF P

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF  5 RANDOM SAMPLES
THE SAMPLES ARE OF SIZES              10        10        10        10        10

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE CHI SQUARE TEST OF DIFFERENCES -   5 BY  6

THE MEDIAN TEST

THE KRUSKAL-WALLIS TEST

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:
```

```
SAMPLE  1 IS CALLED FIRST
SAMPLE  2 IS CALLED SECOND
SAMPLE  3 IS CALLED THIRD
SAMPLE  4 IS CALLED FOURTH
SAMPLE  5 IS CALLED FIFTH

INPUT OF DATA

THERE ARE  5 DATA ITEMS PER READ
INPUT IS FROM UNIT  1

INPUT FORMAT STATEMENT
(5F1.0)


IKMS211I  END OF DATA CARD MISSING.  ASSUMED PRESENT.


IKMS251I  INPUT SAMPLE SIZES DON'T MATCH SPECIFIED SIZES.  INPUT SIZES USED.


END OF INPUT OF DATA
```

THE CHI-SQUARE TEST FOR DIFFERENCES IN PROBABILITIES - 5 BY 6

HO: THE PROBABILITY OF APPEARRING IN ANY COLUMN IS THE SAME FOR ALL SAMPLES

VERSUS

H1: THE PROBABILITY OF APPEARRING IN ANY COLUMN IS DIFFERENT FOR AT LEAST 2 SAMPLES

THE SAMPLES ARE TO BE DICHOTOMIZED INTO 6 CLASSES AS FOLLOWS:

-.1000E 76 TO 1.000 1.000 TO 2.000 2.000 TO 4.000 4.000 TO 6.000
6.000 TO 8.000 8.000 TO 0.1000E 76

THE SAMPLES WERE DISTRIBUTED AS FOLLOWS:

SAMPLE	NUMBER	SIZE	CLASSES	1	2	3	4	5	6
FIRST	1	14		6	3	3	2	0	0
SECOND	2	14		2	1	4	1	5	1
THIRD	3	14		2	1	6	4	1	0
FOURTH	4	14		2	2	2	1	5	2
FIFTH	5	14		5	1	1	3	2	2

THE TEST STATISTIC IS 26.340 WITH 20 DEGREES OF FREEDOM

AT ALPHA = 0.150, HO IS ACCEPTED.

ALPHA HAT = 0.155

END OF THE CHI-SQUARE TEST FOR PROBABILITY DIFFERENCES

THE MEDIAN TEST

HO: THE MEDIANS OF ALL 5 SAMPLES ARE THE SAME

VERSUS

H1: THE MEDIANS OF AT LEAST 2 SAMPLES ARE NOT THE SAME

THERE IS A TOTAL OF 70 OBSERVATIONS WITH THE FOLLOWING BREAKDOWN:

SAMPLE	NUMBER	LESS THAN OR EQUAL	GREATER THAN	TOTAL SIZE
FIRST	1	12	2	14
SECOND	2	7	7	14
THIRD	3	9	5	14
FOURTH	4	6	8	14
FIFTH	5	7	7	14

THE GRAND MEDIAN IS 4.000000

THE TEST STATISTIC IS 7.2237 WITH 4 DEGREES OF FREEDOM

AT ALPHA = 0.150, HO IS REJECTED.

ALPHA HAT = 0.125

END OF THE MEDIAN TEST

THE KRUSKAL-WALLIS TEST

HO: ALL 5 SAMPLES HAVE THE SAME DISTRIBUTION

VERSUS

H1: AT LEAST 1 SAMPLE YIELDS LARGER OBSERVATIONS THAN THE OTHERS

THERE ARE 5 SAMPLES WITH A TOTAL OF 70 OBSERVATIONS

SAMPLE	NUMBER	SIZE	RANK SUM
FIRST	1	14	317.5
SECOND	2	14	581.5
THIRD	3	14	495.0
FOURTH	4	14	618.0
FIFTH	5	14	473.0

THERE WERE 70 TIED OBSERVATIONS IN 10 PLACES

THE TEST STATISTIC IS 9.4131 WITH 4 DEGREES OF FREEDOM

AT ALPHA = 0.150, HO IS REJECTED.

ALPHA HAT = 0.052

END OF THE KRUSKAL-WALLIS TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE, 2 MATCHED SAMPLES - THIS TITLE CARD WAS 'END OF DATA' CARD FOR LAST TEST

DATA=2MS,2,16

LABEL=FIRST,LAST

CODING=16,8,8

TEST,.20,SIGN,MCNEMAR,SIGNRANK

DISTRIB,LEVEL,50.

END OF PARMS


NO PARAMETER SYNTAX ERRORS DETECTED.


THE DATA SUPPLIED CONSISTS OF 2 MATCHED SAMPLES
THE SAMPLE IS OF SIZE   16


THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE SIGN TEST

THE MC NEMAR TEST

THE WILCOXON SIGNED RANK TEST


THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

SAMPLE  1 IS CALLED FIRST
SAMPLE  2 IS CALLED LAST


INPUT OF DATA
```

THERE ARE 16 DATA ITEMS PER READ
INPUT IS FROM UNIT 1

INPUT FORMAT STATEMENT
(16F2.0)

END OF INPUT OF DATA

THE SIGN TEST

HO: E(FIRST) .EQ. E(LAST)

VERSUS

H1: E(FIRST) .NE. E(LAST)

THE SAMPLE SIZE IS 16

THE TEST STATISTIC IS 7

AT ALPHA = 0.200, HO IS ACCEPTED.

ALPHA HAT = 0.804

END OF THE SIGN TEST

2 MATCHED SAMPLES - THIS TITLE CARD WAS 'END OF DATA' CARD FOR LAST TEST

THE MC NEMAR TEST FOR SIGNIFICANCE OF CHANGE

HO: P(FIRST = 0) .EQ. P(LAST = 0)

VERSUS

H1: P(FIRST = 0) .NE. P(LAST = 0)

THE DATA IS DICHOTOMIZED AT 50.000

THE SAMPLE SIZE IS 16

A = 4 B = 3 C = 4 D = 5

THE TEST STATISTIC IS 3.00

AT ALPHA = 0.200, HO IS ACCEPTED.

ALPHA HAT = 1.000

END OF THE MC NEMAR TEST

THE WILCOXON SIGNED RANKS TEST

HO: E(FIRST) .EQ. E(LAST)

VERSUS

H1: E(FIRST) .NE. E(LAST)

THE SAMPLE SIZE IS 16

THERE WAS A TOTAL OF 2 TIED DIFFERENCES IN 1 PLACES

THE TEST STATISTIC IS 72.000

AT ALPHA = 0.200, HO IS ACCEPTED.

ALPHA HAT = 0.509

IKMS901 ***** BECAUSE OF SAMPLE SIZES, ALPHA HAT MAY NOT BE CORRECT. VERIFY WITH TABLES. *****

END OF THE WILCOXON SIGNED RANK TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

PARAMETER ANALYSIS

TITLE, BALANCED INCOMPLETE BLOCK DESIGN (BIBD) DATA FOR THIS TEST

DATA,BIBD,4,6,2,3

BLOCK=JUDGE,TREAT=SAMPLE

CODING,1,BLOCK,TREAT

TEST,,10,ALL

END OF PARMS

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF AN INCOMPLETE BLOCK, WITH 4 BLOCKS AND 6 TREATMENTS
THERE ARE 2 BLOCKS PER TREATMENT AND 3 TREATMENTS PER BLOCK

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE DURBIN TEST

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

BLOCKS ARE CALLED JUDGE
TREATMENTS ARE CALLED SAMPLE

INPUT OF DATA

THERE IS 1 DATA ITEM PER READ
INPUT IS FROM UNIT 1

INPUT FORMAT STATEMENT
(2I1,8X,F1.0)
DATA IS CODED BY BLOCKS TREATMENTS

END OF INPUT OF DATA

THE DURBIN TEST

HO: THE 6 TREATMENTS, CALLED SAMPLE , ARE EQUALLY EFFECTIVE ACROSS BLOCKS, CALLED JUDGE

VERSUS

H1: AT LEAST 1 TREATMENT YIELDS LARGER OBSERVATIONS THAN THE OTHER TREATMENTS

THE OBSERVATIONS ARE TAKEN ACROSS 4 BLOCKS
WITH 3 TREATMENTS IN A BLOCK
AND 2 BLOCKS IN A TREATMENT

THE TEST STATISTIC IS 5.0000 WITH 5 DEGREES OF FREEDOM

AT ALPHA = 0.100, HO IS ACCEPTED.

ALPHA HAT = 0.416

END OF THE DURBIN TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

```
PARAMETER ANALYSIS

TITLE, TELEPHONE WRONG NUMBERS, FELLER PAGE 161

LABEL=WRONGNUM

DATA,IRS,1,267

CLASS,COLUMN,13,3,4,5,6,7,8,9,10,11,12,13,14

DISTRIB,POISSON,8.74

CODING,3,3

TESTS,.05,GOODFIT

END

NO PARAMETER SYNTAX ERRORS DETECTED.

THE DATA SUPPLIED CONSISTS OF 1 RANDOM SAMPLE
THE SAMPLE IS OF SIZE  267

THE FOLLOWING TESTS WILL BE PERFORMED ON THE SUPPLIED DATA

THE CHI-SQUARE GOODNESS OF FIT TEST

THE DATA FOR THIS EXPERIMENT IS LABELLED AS FOLLOWS:

THE SAMPLE IS CALLED WRONGNUM

INPUT OF DATA

THERE ARE  3 DATA ITEMS PER READ
```

INPUT IS FROM UNIT 1

INPUT FORMAT STATEMENT
(3F2.0)

END OF INPUT OF DATA

THE CHI-SQUARE GOODNESS OF FIT TEST

HO: THE SAMPLE COMES FROM A POPULATION WITH A POISSON DISTRIBUTION OF MEAN 8.740

VERSUS

H1: THE SAMPLE DOES NOT COME FROM A POPULATION AS DESCRIBED IN HO

THE SAMPLE WAS BROKEN DOWN AS FOLLOWS:

CLASS	CLASS INTERVAL	OBSERVED	EXPECTED
1	-.1000E 76 TO 3.000	6	6.804
2	3.000 TO 4.000	11	10.390
3	4.000 TO 5.000	14	18.162
4	5.000 TO 6.000	22	26.455
5	6.000 TO 7.000	43	33.031
6	7.000 TO 8.000	31	36.087
7	8.000 TO 9.000	40	35.044
8	9.000 TO 10.00	35	30.629
9	10.00 TO 11.00	20	24.336
10	11.00 TO 12.00	18	17.725
11	12.00 TO 13.00	12	11.916
12	13.00 TO 14.00	7	7.439
13	14.00 TO 0.1000E 76	8	8.982
TOTAL		267	

THE TEST STATISTIC IS 7.7952 WITH 12 DEGREES OF FREEDOM

AT ALPHA = C.05C, HO IS ACCEPTED.

ALPHA HAT = C.801

END OF THE CHI-SQUARE GOODNESS OF FIT TEST

REQUESTED TESTING OF THIS DATA IS COMPLETE

IKM500I END OF IKMSTAT RUN

IKM501I TOTAL EXECUTION TIME WAS 106.01 SECONDS

APPENDIX IV

PRELIMINARY USER'S GUIDE

DISCLAIMER

Although this program has been tested by its contributor, no warranty, express or implied, is made by the contributor or the Kansas State University Department of Statistics and Computer Science or the Statistical Laboratory as to the accuracy and functioning of the program and related program material. Any questions concerning the technical details of the programming for this routine should be directed to the Statistical Laboratory, Kansas State University, Manhattan, Kansas.

I. INTRODUCTION

This manual is a user's guide to the nonparametric statistics package, NONPAR. To avoid lengthy explanations it assumes the user has some knowledge of both FORTRAN and statistics (though not necessarily nonparametric statistics). A user who has no knowledge of FORTRAN should be able to use this manual to set up test packets with perhaps, the aid of members of the statistical laboratory staff when needed.

The nonparametric tests included in this package cover a very broad area, ranging from the simple Sign Test on bivariate data to a complex rank test analysis of a Randomized Complete Block Design. The user must be able to determine the type of data he has and, of course, what he is trying to test. In many cases the user can select more than one method of analysis, but is cautioned not to use several methods and then select the answers he likes best.

One test that may be of interest to most users is the Goodness of Fit Test, which can be used to test data for normality. This can be used as a preliminary step to a parametric analysis. If the data is unsuited for such analysis, the nonparametric methods may be used and the user will have no doubts that his results are statistically sound.

For information concerning FORTRAN the user is directed to IBM's FORTRAN IV Language Specifications and Programmer's Guide (forms GC28-6515 and GC28-6817 respectively). For information on nonparametric statistics the user may consult Practical Nonparametric Statistics, by W. J. Conover, 1970. Other nonparametric statistics texts or references may also be consulted, but this user's guide follows the lines of development used in the text by Conover.

II. PARAMETER CARDS

Parameter cards are used to provide information to the program about the data, which tests are to be performed, contingency table limits, and other information. With the exception of the 'END OF PARAMETERS' card (which must be last) they may be supplied in any order. However, it is recommended that the TITLE card be first. If more than one card of a particular type is supplied, the information supplied in the card appearing last is used with no error indication.

General Format and Syntax

A parameter card usually consists of a keyword, starting in column 1, followed by either an equal sign (=) or a comma (,), followed by one or more parameters, each separated by commas. (Two exceptions to this will be pointed out later.) A parameter card might look like this:

```
KEYWORD,a,b,c
KEYWORD=x,y
```

In this and all succeeding examples, UPPER case letters will represent information which must be coded as shown. Lower case letters will represent variable information that the user will supply. The parameter information on a card is terminated by the remaining columns of the card, following the last parameter, being blank. Therefore the user MUST NOT put comments on parameter cards.

If a parameter is shown in examples in UPPER case letters, it must be coded as shown. (If a parameter must be a numerical value, the parameter may be any number capable of being converted to a real number.) Numbers on parameter cards may include a sign, decimal point, and exponent, but must follow the FORTRAN rules for number representation. If a parameter is spe-

cified as having to be an integer, it must be a positive, whole number with no decimal point or exponent.

All parameters must be 8 or less characters, including any blanks that the user wishes to insert. A parameter is considered terminated by a comma, by the rest of the card being blank, or upon reaching the 8th character (scanning left to right) without finding either of the other two conditions. If a parameter is longer than 8 characters the 9th and succeeding characters are considered the next parameter WITH NO ERROR INDICATION GIVEN.

Explanations of each individual parameter card follow.

A. The TITLE Card

The TITLE card is used to supply a title which is printed at the top of every page of output. It's use is optional, but is recommended because parameter cards are not listed until a TITLE card is read (unless an error is detected). If no TITLE card is supplied, the user will have no listing of his parameter cards. If the TITLE card is not the first parameter card, those parameter cards coming before the TITLE card will not be listed. If a second TITLE card is supplied the program will start a new page with the new title and continue.

The contents of columns 7-80 of the TITLE card are used as the heading of each page. In order to enhance the appearance of the output, the title is centered on the page, regardless of where the title is located on the TITLE card. Examples of TITLE cards are:

1) TITLE, THIS IS A TITLE CARD
2) TITLE=THIS IS A SECOND TITLE
3) TITLE= THIS IS A TITLE CARD

Cards 1) and 3) above would produce the EXACT SAME title on the output.

B. The DATA Card

The DATA parameter card is used to define the type of data being used. This parameter card is MANDATORY. Its general format is:

DATA,t,k,s_1,s_2,s_3,.....

where "t" is the data type and "k" and the "s"'s are dependent of the type specified.

The values for the parameters are summarized as follows:

"t"	"k"	number of "s" parameters
1RS	1 or omitted	1 or omitted
2RS	2 or omitted	2 or omitted
KRS	number of samples	"k" or omitted
BIVAR	2 or omitted	1 or omitted
MULVAR	multiplicity	1 or omitted
2MS	2 or omitted	1 or omitted
RCB	See below	1 or 2, see below
BIBD	See below	3, see below

A "t" of 1RS means that the data consist of ONE RANDOM SAMPLE of size "s_1".

A "t" of 2RS means that the data consist of TWO RANDOM SAMPLES of sizes "s_1" and "s_2", respectively.

A "t" of KRS means that the data consist of "k" RANDOM SAMPLES of sizes "s_1", "s_2", ..., "s_k", respectively.

A "t" of BIVAR means that the data consist of a BIVARIATE SAMPLE of size "s_1".

A "t" of MULVAR means that the data consist of a MULTIVARIATE SAMPLE of multiplicity "k" and sample size of "s_1".

A "t" of 2MS means that the data consist of TWO MATCHED SAMPLES of size "s_1" each.

The "s"'s above represent the sample size(s). If the sample sizes are the same for each sample, these may be omitted. They must be specified for

the case of 2 or more random samples (2RS and KRS) with unequal sizes, since the program would otherwise be unable to establish the number of observations in each sample.

For both RCB (RANDOMIZED COMPLETE BLOCK) and BIBD (BALANCED INCOMPLETE BLOCK DESIGN) "t" specifications, the "k" and "s_1" parameters specify the number of BLOCKS and TREATMENTS, respectively. These two parameters are required for both data types. For RCB data "s_2" specifies the number of replications. If omitted one is assumed. For BIBD both "s_1" and "s_2" are mandatory and specify, respectively, the number of BLOCKS per TREATMENT and the number of TREATMENTS per BLOCK. As a check of valid specification of the BIBD parameters, "k" times "s_3" must equal "s_1" times "s_2". (See the discussion of BIBD data in Conover for more details on these parameters.)

Some examples of DATA cards are:

DATA,1RS,,5	1 Random Sample of 5 observations
DATA,2RS,2,10,15	2 Random Samples of 10 & 15 obs., resp.
DATA,KRS,6	6 Random Samples of equal, unspecified sizes
DATA,BIVAR,,40	A Bivariate Sample of 40 observations
DATA,MULVAR,3	A Trivariate Sample of unspecified size
DATA,2MS,2,25	2 Matched Samples of 25 obs. each
DATA,RCB,4,3,4	An RCB Sample - 4 Blocks, 3 Treatments, 4 Replications
DATA,BIBD,6,8,3,4	A BIBD Sample - 6 Blocks, 8 Treatments, 3 Blocks/Treatments, 4 Treatments/Blocks

C. The BLOCK, TREAT, and REP Card(s)

When RCB and BIBD data is specified as input the user may optionally provide names for the Block, Treatment, and Replication. These are NOT names for EACH Block, Treatment, or Replication category, but one for each

of the three sample breakdowns (Blocks, Treatments, Replications). For example, the Blocks may be age groups, the Treatments income groups, and the Replications ethnic background.

Either one, two, or all three labels may be provided on one card. The examples below show how this is accomplished:

```
BLOCK=AGE,TREAT=INCOME,REP=ETHNIC

BLOCK=DAYS

TREAT=TIME,REP=SEX

REP=OBS,BLOCK=SIRE

TREAT=DAM
```

The BLOCK, TREAT, and REP keywords MUST be followed by an equal sign (=), rather than a comma. Specifications of names longer than 8 characters will cause a syntax error since the program will consider the remaining characters part of the next keyword.

D. The LABEL Card(s)

The LABEL parameter card is used to provide labels for the variables in the data, so that they can be referred to by name in the output. If the data consist of random samples each name supplied refers to a sample. If Bivariate or Multivariate observations are specified each name refers to a variate. The general format of the LABEL card is:

$$\text{LABEL}(n)=l_1,l_2,l_3,\ldots$$

The "(n)" is optional. If omitted, n=1 is assumed. The "n" represents which sample or variate the first label is for, i.e., the first label after the equal sign (l_1) is assigned to the nth sample or variate. The second label (l_2) is assigned to the $n+1^{th}$ sample or variate. This continues for

as many labels as are on a card.

Since there may not be enough room on one card to label all variables, more than one LABEL card can be provided. The "(n)" specification is used to tell the program where to begin assigning labels, so that previously specified labels will not be overridden, unless the "(n)" specifications overlap.

The LABEL card is one of the two exceptions to the general parameter format in that the character following LABEL or LABEL(n) must be an equal sign (=). The other exception is the BLOCK, TREAT, and REP labels.

Examples of labelling are:

LABEL=FIRST,SECOND	labels for variables 1 & 2
LABEL(1)=X,Y,Z	labels for variables 1, 2, & 3
LABEL(4)=A,B,C	labels for variables 4, 5, & 6
LABEL(17)=OUTPUT,AGE,SEX	labels for variables 17, 18, & 19

If any label is provided that has more than 8 characters, the LABEL card will be flagged as having invalid syntax.

E. The SOURCE Card

The SOURCE parameter card specifies the location (logical unit number) and form of the input data, and an optional unit on which to write out the data. If all of the defaults are suitable the SOURCE card may be omitted entirely. The format of the SOURCE card is:

SOURCE,a,b,c,d

"a" is the input logical unit number. If this parameter is omitted (by coding sequential commas), the program assumes that the data are on the same unit as are the parameters, generally unit 1 (see the section on FORTRAN Unit Numbers).

"b" is the form of the input data and may be the words EXTERNAL or INTERNAL. EXTERNAL means that the user has supplied a FORMAT statement (immediately following the parameter cards) which will be used to read the data. INTERNAL means that the data exists in internal form (unformatted) and consists of data previously written out by either this or another program. If "b" is omitted, the default depends on "a". If "a" specifies the same unit as the parameters were read on (or is omitted), the default is EXTERNAL. If "a" is not equal to the parameter input unit, the default is INTERNAL.

The "c" and "d" parameters specify the writing out of the data by the program. If no data output is desired, they should be omitted. "c" specifies the output unit number. The data are written out on this unit unformatted and the number of items written in each record depends on other specifications and can be determined from the following table.

Table 1. Items Written In Each Output Record

"d" parameter	1RS,2RS,KRS,2MS	BIVAR	MULVAR	RCB,BIBD
RAW	i	i	i	i+1
TRANSFORMED	t	t	t	t+1
SAMPLE	1 or t	2	x	x
RANKED	1	x	x	x

i: number of items on each read (see CODING card)
t: number of items after user transformation
1: number of user location parameters (see CODING card)
x: invalid "d" specification

The "d" parameter specifies when during the processing the data should be written out. This parameter may be either RAW, TRANSFORMED, SAMPLE, or RANKED. RAW means the data are written out immediately after reading it. TRANSFORMED means to output the data after the user's RAW data transformation, if any. SAMPLE means the output is written on an observation by observation basis for each sample, taking any user SAMPLE transformation into account. RANKED is the same as SAMPLE, except after the data has been sorted into ascending order. Note in Table 1 that some options are not valid for all types of data.

For the Random Sample case (1RS,2RS,KRS,2MS), the output by SAMPLE will consist of all observations from sample 1, followed by all observations of sample 2, etc. For bivariate data (BIVAR), the output by SAMPLE will be by pairs of observations.

If the "c" parameter that is specified is the same unit number as that of the printed output unit (usually 3; see the section on FORTRAN Unit Numbers), the data will be printed, forming a listing of the data. In this case, the data will be printed using a FORMAT specification of 7G16.8. If more than 7 items are to be written at one time, as many lines as needed will be printed.

F. The TESTS Card

The TESTS parameter card contains the information required to perform the desired tests. It provides the significance level, the tail(s) to be tested, and the tests to be performed. The general format of the TESTS card is:

TESTS,a,t_0,t_1,t_2,...

The "a" parameter is the significance level (alpha) at which the tests are to be performed. It must be a number larger than zero and smaller than one. Currently this value MUST be specified.

The "t_0" parameter is the specification of the tail to be tested. The valid values of this parameter are UPPER, LOWER, and 2-TAIL. If this parameter is omitted a two tailed test is performed.

The "t_1", "t_2", ..., etc., parameters specify the tests to be performed. If the tail specification is omitted, "t_0" may also be used to specify tests. The tests are specified by keyword abbreviations which stand for the desired tests. These keywords are summarized in Table 2 and are also given in the descriptions of each test.

Table 2. Keyword Abbreviations For The TESTS Parameter Card

KEYWORD	TEST NAME
SIGN	Sign Test
MCNEMAR	McNemar Test
CHIPROB	Chi-Square Test for Probability Differences
CHINDEP	Chi-Square Test for Independence
MEDIAN	Median Test
SIGNRANK	Wilcoxon Signed Ranks Test
MANN-WHITNEY or	
M-W	Mann-Whitney Test
MOOD	Mood's Test of Variances
RHO	Correlation Using Spearman's Rho
TREND	Test of Trend Using Spearman's Rho
TAU	Correlation Using Kendall's Tau
KRUSKAL-WALLIS or	
K-W	Kruskal-Wallis Test
FRIEDMAN	Friedman Test
CHIGOF or	
GOODFIT	Chi-Square Goodness of Fit Test
DURBIN	Durbin Test

Instead of listing all the tests applicable in a given situation the user may use the keyword ALL in place of the tests keywords. The program will perform all appropriate tests for the supplied data and other parameters, such as contingency table limits or a distribution function.

Examples of some typical TESTS cards are:

TESTS,.05,LOWER,SIGN,RHO,TAU

TESTS,.10,MEDIAN,K-W

TESTS,.01,2-TAIL,ALL

TESTS,.05,ALL

TESTS,.05,UPPER,ALL

G. The CLASSES Cards

The CLASSES parameter card is used to specify information which the program can use to build a contingency table. It may be used twice in any one set of parameters; once for row specifications and once for column specifications. The general format of the CLASSES card is:

CLASSES,a,s,l_1,l_2,l_3,...,l_{s-1}

The "a" parameter is either ROWS or COLUMNS, signifying that the remaining parameters on the card are for rows or columns, respectively. The "s" parameter is an integer less than or equal to 20, and specifies the number of rows or columns in the contingency table.

The "l_i" parameters specify upper limit values for each of the "s" rows or columns except the last one. When categorizing an item into a row or column the item is placed in row or column 1 if it is greater than -10^{76} and less than or equal to "l_1". It is placed in the second row or column if it is greater than "l_1" and less than or equal to "l_2". The "l_s" for

the last classification is always ignored. The upper limit of the last row or column is always 10^{76}. Hence, the user should only specify "s-1" upper limits. Note that the "1_i" values must be in ascending sequence.

Examples of CLASSES specifications are:

CLASSES,ROWS,5,-5,-2,2,5

CLASSES,COLUMNS,4,2.5,7.5,12.5

CLASSES,ROWS,2,4

CLASSES,COLUMNS,6,-7,0,4,10,15

H. The TRANS Card

The TRANS parameter card supplies information concerning user transformation of the data to be processes. Its general format is:

TRANS,a

"a" equals RAW, SAMPLE, or NONE.

If none is the requested option the card may be omitted entirely. The RAW option specifies that the user wishes to transform the data exactly as it is read in before the program attempts to process it in any manner. The SAMPLE option, which is invalid for MULVAR, RCB, and BIBD data, specifies that the user wishes to transform the data on a sample by sample basis. See the section entitled "Transformation of Data" for details on how data may be transformed with each specification.

I. The DISTRIB Card

The DISTRIB card has three different uses. It can be used to specify a dichotomization level, a median value, or a distribution function. Its use depends on the data being tested and the tests specified by the user. The general format of all three cases is:

DISTRIB,a,b,c

The use of the DISTRIB card to specify a dichotomization level is valid only when the data consist of a bivariate sample and the McNemar Test is requested. For this case "a" is either the keyword LEVEL or omitted (by coding two consecutive commas). The "b" parameter specifies the dichotomization level below which an observation is considered a zero and above which an observation is considered a one. The "c" parameter is ignored, but if supplied, it must be numeric. Examples of this use of the DISTRIB card are:

DISTRIB,LEVEL,25

DISTRIB,,-2.5

The second use of the DISTRIB card is when the data consist of a single random sample and the Wilcoxon Signed Ranks Test is specified. The DISTRIB card specifies a median value to be used in the test. The "a" parameter is either the keyword MEDIAN, or omitted. The "b" parameter specifies the median value and the "c" parameter is ignored. Examples of this use are:

DISTRIB,MEDIAN,7.5

DISTRIB,,0

The third use of the DISTRIB card is for the case of a single random sample and the Chi-Square Goodness of Fit Test. Both the Wilcoxon Signed Ranks Test and the Goodness of Fit Test CANNOT be specified in the same set of parameters, even though they both work on a single random sample. For the case of the Goodness of Fit Test, the "a" parameter specifies the type of distribution and the "b" and "c" parameters are parameters of the distribution function. These are summarized as follows:

"a"	"b"	"c"
NORMAL	mean	variance
POISSON	mean	ignored
UNIFORM	ignored	ignored
BINOMIAL	population size	probability

Examples of this case are:

```
DISTRIB,NORMAL,2.5,7
DISTRIB,POISSON,8.74
DISTRIB,UNIFORM
DISTRIB,BINOMIAL,10,.5
```

J. The CODING Card

The CODING card is one of the most important parameter cards, even though its use is optional. With the CODING card the user describes how the data are to be read and distributed into samples, variates, blocks, treatments, or replications. The CODING card has two formats: One for random sample, bivariate, multivariate, or matched sample data; the other for RCB or BIBD data. In the ensueing discussion, the term "sample" is considered to mean either sample or variate, depending on the type of data.

The format of the CODING card for data classes as samples or variates is:

CODING,r,s_1,s_2,s_3, ...,s_n

The "r" parameter specifies the number of data items that will be read by each read command in the program. This would normally be the same as the number of E and F fields in the user's FORMAT or the number of words

in an internal data record. If omitted one item is assumed and "r" has an upper limit of 40.

The "s" parameters specify how the input data should be broken down into samples (variates). One "s" specification should be provided for each sample or variate that the user has specified on the DATA card. If any "s" specifications are missing or omitted, one is assumed for each missing one. The "s" parameters must be integers and are used by the program in the following way: The first "s_1" items read (or provided by user transformation) are placed into sample 1, the next "s_2" items into sample 2, and so on until the end of the data is reached. Hence, if the user had 2 random samples of unequal sizes, say 10 and 15 observations, respectively, an "s_1" of 10 and "s_2" of 15 would place the first 10 items read in into the first sample and the next 15 items into the second sample. If the input contained more than 25 items the program would start placing items into sample 1 again.

The sample breakdown bears NO relation to "r", the number of items per read. If, as in the above mentioned example, "s_1" and "s_2" were 10 and 15, and "r" was 5, the items from the first two reads would be placed into sample 1, the items from the next three reads into sample 2 and then back to sample 1 if more data existed. If "r" had been 25 the first 10 items would go into sample 1 and the remaining 15 into sample 2.

Although the user may specify the total sample sizes on the DATA parameter card, the program does not stop placing items into a particular sample when this size has been reached. For example, in the samples of 10 and 15 items, if "r" were specified as 15, 10 items from the first read would go into sample 1, 5 items from the first read and 10 items from the second read would go into sample 2. The remaining 5 items would be placed into sample 1, even though no more data were present after the second read.

The above explanation used the "items per read" as the number of items that were broken down into samples. This is not exactly true when the user has specified RAW data transformation. If RAW transformation is done the number of items returned by the transformation subroutine are broken into samples.

Some examples of CODING cards for the above explanation might be:

CODING,5,10,15

CODING,1,1,1,1,1

CODING,3,1,1,1

CODING,2,3,1,2

The second type of CODING card specification is for RCB or BIBD data. The general format for this type of CODING card is:

CODING,r,c_1,c_2,c_3

The "r" parameter is the same as the "r" parameter of the other form of the CODING card. The "r" value DOES NOT include the number of location parameters (explained below) present on each read.

The "c_i" parameters specify how the data have been coded to identify which block, treatment, and replication the item(s) belong in. None, one, two, or three "c" parameters may be specified, indicating how many locations they indicate. Specifying "c_1" as the keyword NONE has the same effect as omitting all of the "c_i" parameters.

The "c_i" parameters can be the keywords BLOCK, TREAT, and REP, indicating that location information (subscript) is present on every read (not every item) for blocks, treatments and replications, respectively. The number of location parameters present on each read is equal to the number of keywords specified on the CODING card and must also equal the number of

I-fields in the user's FORMAT statement.

If all this sounds confusing, don't worry. It is! The following explanations and examples should clarify the use of coding parameters. In each of the following examples, the user is assumed to have specified "B" blocks, "T" treatments, and "R" replications (possibly 1).

If NONE is specified, the first item read is placed into block 1, treatment 1, replication 1. The second item goes into block 1, treatment 1, replication 2 (unless only 1 replication was specified). The (R+1)th item goes into block 1, treatment 2, replication R. The specification of NONE will not work for BIBD data since it would imply that every block-treatment combination is present.

If all three keywords (BLOCK,TREAT,REP) are specified, every read specifies an explicit location for every item read. The same holds for the case of only BLOCK,TREAT specified and only one replication.

If only two location parameters are specified (BLOCK,TREAT,BLOCK, REP, or TREAT,REP) the unspecified parameter is provided by the program. The unspecified parameter starts at 1, goes to its limit (B, T, or R), then starts back at 1, regardless of what the locations provided are. If BLOCK,TREAT was specified and the first two items were labelled (1,1) and (1,2) the resulting locations would be (1,1,1) and (1,2,2), again assuming more than one replication was specified. Hence, the program assumes that the unspecified parameter is in minor sort order. In the case cited, it does not matter what replication number the program assigns to a particular block-treatment combination.

If only one location parameter is specified (BLOCK,TREAT, or REP), the other two locations are supplied by the program. These two unspecified parameters are varied from 1 to their limits, varying replications first,

then treatments, then blocks, not varying the supplied parameter. For example, if TREAT was specified, the block and replication numbers would be supplied in the following order: (1,1), (1,2), ..., (1,R), (2,1), (2,2), ..., (2,R), ..., (B,1), (B,2), ..., (B,R).

A word of warning. The program will consider it an error if the user provides more than one item for a given block-treatment-replication combination. The automatic assignment of location parameters can cause this if, for example, all the replications for a specified block-treatment combination are not placed together in the input. It is generally advisable to supply both the block and treatment location parameters whenever possible, and to arrange the input so that all the replications for each block-treatment combination are together.

If "r", the number of items read, is greater than 1, the program assigns the second through "r"th items to block-treatment-replication combinations by varying the UNSPECIFIED location parameter(s). Using this feature, the user could, for example, specify his data locations by block-treatment and have all the replications on one card. With the numerous combinations of specified and unspecified parameters and items per read, the user has tremendous flexibility, but also has himself a potful of trouble if he is not very careful.

Another note of warning. Regardless of the order in which the BLOCK, TREAT, and REP keywords are specified, in reading the data the program ALWAYS USES THE BLOCK, TREATMENT, REPLICATION ORDER IN PICKING UP THE LOCATIONS FROM THE INPUT DATA. The location parameters must be read in using I specifications in the user's FORMAT and must be the first field specification(s) in the FORMAT. If the data are coded internally, the

locations must be integer values and must be the first word(s) of each record. Also, specification of the same keyword twice (such as BLOCK, BLOCK,REP) on the CODING card may not be caught as an error (due to the scheme used to analyze these parameters), but might cause the data to be read improperly or cause the program to abnormally terminate without any indication of the real source of the error.

Some examples of the RCB-BIBD type of CODING cards are:

```
CODING,1,BLOCK,TREAT

CODING,4,TREAT,REP

CODING,1,REP,BLOCK,TREAT

CODING,1,NONE

CODING,12,BLOCK
```

K. The END OF PARAMETERS Card

The END OF PARAMETERS must be the last parameter card provided for a set of parameter statements. It must have a single blank between each of the words and has no user supplied information on it. There are, however, several optional forms of this statement. Each of the following examples are acceptable to the program.

```
END OF PARAMETERS

END

END OF PARMS
```

All that is actually required is the word END. Anything following the P is not examined.

III. TYPES OF DATA

For the sake of simplicity, all data used by this program has been classified into one of eight types. While some types are special cases of more general types, they were separated because analysis techniques were considered different.

The first type of data is ONE RANDOM SAMPLE (1RS). It consists of one or more observations from a single statistical population.

The second type of data is TWO RANDOM SAMPLES (2RS). It consists of observations taken from two (distinct) populations and do not have to have the same number of observations in each sample. This is unlike most parametric analysis work which requires equal sample sizes. This type can be considered a special case of the next type of data, but is coded as a distinct type.

The third type of data is "K" RANDOM SAMPLES (KRS), where "K" is some number greater than 1 (and less than 41). It consists of observations taken from "K" different populations. Like the case of two random samples the sample sizes do not have to be the same for each sample.

The fourth type of data is a BIVARIATE SAMPLE (BIVAR), where two observations can be paired together with something in common, such as before and after on individual experimental units. This means that two distinct items are put together to form an "observation" for purposes of analysis.

The fifth type of data is a generalization of the bivariate sample. It is a MULTIVARIATE SAMPLE (MULVAR) and consists of "N" observations which can be placed together for some reason, such as a single experimental unit under several different conditions. "N" must be greater than (NOT equal to) 2 and less than 41. There must always be "N" items per "observation".

The sixth type of data is a special case of two random samples. It is called TWO MATCHED SAMPLES (2MS) and consists of observations from two different populations and each sample must be the same size. The observations are paired (one from each sample) for some reason which the user must be aware of. This is the method by which two random samples can be analyzed as a bivariate sample. The user is cautioned that there must be some reason for the pairing or the subsequent analysis will be meaningless.

The seventh type of data is data for the RANDOMIZED COMPLETE BLOCK (RCB) experiment. The observations are assigned to certain blocks, treatments, and replications, either by the user, the program, or a combination of the two. There must be at least two block and treatment catagories, but multiple replications of these block-treatment combinations are optional. All data for all block-treatment-replication-combinations must be provided. The program has no facilities to handle missing data other than the user transformation subroutine.

The eighth, and last, data type is a special case of the Randomized Complete Block type data and is called BALANCED INCOMPLETE BLOCK DESIGN (BIBD) data. This type of data consists of block-treatment combinations with not every block paired with every treatment. Multiple replications are NOT allowed, an equal number of treatments must appear in every block, and an equal number of blocks must be used for each treatment.

Examples of specifying each type of data are given in the section on the DATA parameter card.

IV. INPUT OF DATA

After analyzing the parameter cards the user has supplied, the program reads the user's data according to his specifications. If the data were specified as being in INTERNAL form (Unformatted) the data are read as an array of SINGLE PRECISION items, location parameters first, then the data values. If the data were specified as being in EXTERNAL form the user's FORMAT statement is read, then the data are read according to the specifications of the FORMAT statement. Again, the data are read in as an array of SINGLE PRECISION items or location parameters first.

A. The User's FORMAT Statement

If the data supplied are to be read in EXTERNAL form, such as punched cards, a FORMAT statement must be supplied so that the program can properly read the data. This FORMAT statement MUST immediately follow the END OF PARAMETERS card. This FORMAT is coded according to the rules of FORTRAN and can contain E, F, X, T, I (for location parameters only), and G specifications, can have slashes (/), repetitive specifications (7F3.2), and can have group specifications (4(F2.0,F3.0)). Since the FORMAT supplied could conceivably be longer than the 80 characters available on one card, the FORMAT statement can occupy from one to ten cards.

The FORMAT supplied must conform to the following rules:

1) The word FORMAT is never coded at the beginning of the FORMAT statement as it would be in a program.
2) The FORMAT must start with its opening (left) parenthesis in column 1 of the first card.
3) The FORMAT statement should be coded with few or no embedded blanks between specifications.

4) The closing (right) parenthesis must be placed immediately following the last specification and the remainder of the card MUST be blank.

5) The FORMAT is coded from card to card without regard to card boundaries (column 80 of card 1 is followed immediately by column 1 of card 2).

Some typical FORMAT statements are:

```
(4F3.2)

(I2,5X,F6.4)

(2I3,T21,6(2F2.0,F1.0,F6.3),1X,E14.4)
```

Note that the I specifications are used to read in location parameters, while the F, E, and G specifications are used to read in the values of the variables in the data. X and T are used for card column skipping and positioning.

B. The END OF DATA Card

When the data being read are in EXTERNAL form (requiring a FORMAT statement), a special END OF DATA card is used to signal when no more data are present. The use of this card allows the program to have multiple sets of parameters and data in a single program run.

The END OF DATA card is coded with the words END OF DATA in columns 1 and 12 of the card following the last actual data card, with <u>one</u> blank between each word and at least one blank after DATA. When this card is not present following the data, the program attempts to read whatever follows the data, as data. If there happens to be another set of parameters following the data, a data conversion error will result which will terminate processing of the set of data being read. An exception to this exists and is discussed below.

If the user supplies data such that he does not read anything from columns 1-11, the END OF DATA card will not be recognized unless some non-blank, nonnumeric character (such as *) is placed in at least one of the columns that is read. The reason for this is: The END OF DATA card causes a data conversion error to occur which the program intercepts. If the card in error is an END OF DATA card no error is noted and the program assumes that the end of the supplied data has been reached, regardless of the card column in which the error occurred. Note that columns 4, 7, and 12 must be blank and if these are the only locations of the user data, processing by this program would be impossible except as noted below.

If the END OF DATA card is omitted and the actual end of file (/*) is read, or if the END OF DATA is omitted and card that causes the resulting conversion error is a TITLE card (word TITLE in columns 1-5), the user is forgiven with a warning message. (This is another good reason for always making the first parameter card a TITLE card.) These two methods can be used to process data that appears only in columns 4, 7, or 12. In order for the TITLE card to be intercepted the user must make sure that there is some nonblank, nonnumeric character in at least one of the columns that are read. In the event of a TITLE card being intercepted the contents of the card are saved for use as the TITLE card for the next set of parameters.

Note that the END OF JOB and END OF RUN cards will NOT signal end of data, but will give a conversion error.

C. The END OF JOB Card

The END OF JOB card is the last card in the ENTIRE job submitted. If used, it would follow the END OF DATA card of the last set of data or the END OF PARAMETERS card if no data are present. Upon reading the END OF JOB card all further processing is terminated and control is passed back to the operating system. Its use is entirely optional. End of file on the input

unit will produce the same result. The use of this card was provided because of the batching facilities of the program. Like the END OF PARAMETERS card, this card also has optional formats. They are:

END OF JOB

END OF RUN

V. USER PROVIDED DATA TRANSFORMATION

This package has the facilities to allow the user to alter (transform), add, or delete data items. This is done by the user providing the proper specification on the TRANS parameter card and by writing, compiling, and linking into the program, a FORTRAN subroutine with the name TRANS. This is actually a lot easier than it may sound.

The TRANS parameter card tells the program when to call the TRANS subroutine to allow the subroutine to change data values. The TRANS subroutine is compiled and linked into the program by use of a special Job Control Procedure designed for user transformation, as opposed to the normal procedure (see the section on Job Control Cards). Should the user forget to provide a TRANS subroutine, the program would not be aware of the fact and would run as if no transformations had been specified. This is because the user supplied TRANS subroutine actually replaces a dummy TRANS subroutine (that does not alter the data in any way) that was made part of the program specifically to prevent accidental omission of TRANS by the user.

The TRANS subroutine is written according to the standard rules for FORTRAN subroutines. When called no parameters are passed. Instead, all of the information the user should need is provided in a COMMON block named /USER/. This COMMON block should be declared as follows:

```
COMMON /USER/ ITEMS,X(40),LOCAT(4),RDNO
INTEGER*4 ITEMS,LOCAT,RDNO
REAL*4 X
```

Integer and Real declarations are not needed for ITEMS, LOCAT, and X, but were provided above to explicitly show their types. The fields will

have the following contents upon ENTRY to the subroutine:

ITEMS will contain the number of data values in the X array.

X will contain the data to be transformed.

LOCAT(1)(2)(3) will contain the location parameters THAT WERE READ.

LOCAT(4) will contain the number of location parameters in LOCAT(1)(2)(3). If this is one, LOCAT(1) is a location value and LOCAT(2) and (3) contain no valid information. If it is two, only LOCAT(1) and (2) contain valid information. If LOCAT(4) is three, LOCAT(1), (2), and (3) all contain valid information.

RDNO will contain the number of the read the data came from (first, second, etc.). This value is provided ONLY for RAW transformation and will contain a meaningless value when transforming by SAMPLE.

The "ITEMS" values in the X array are stored in consecutive elements starting with (1).

Upon returning to the program these fields should contain the following:

ITEMS should be the number of values presently in the X array. If ITEMS is larger at return than upon entry, values should have been added. If smaller, values should have been deleted. If the same size, the values MAY have been altered. If ITEMS is zero (or less) the program assumes that ALL of the X values have been deleted and are not to participate in subsequent processing, hence the observation is dropped.

X should contain the data that the program is to process. These values MUST occupy the first through "ITEMS"th elements of X. This means that if an X value is deleted, those following it in the array that

are not to be deleted will have to be moved down to occupy the space of the deleted value. ITEMS would have to be changed correspondingly.

LOCAT(1)(2)(3) should contain either the original location parameter values or location parameter values that were provided or modified by the TRANS subroutine.

LOCAT(4) MUST NOT BE ALTERED IN ANY WAY.

RDNO MUST NOT BE ALTERED IN ANY WAY.

Upon entry the value of ITEMS depends upon the type of transformation (RAW or SAMPLE) and the type of data specified. If RAW transformation is specified the value of ITEMS will be the number of items per read specified on the CODING parameter card. If SAMPLE transformation is specified the value of ITEMS will be 1 for 1RS, 2RS, KRS, and 2MS data, and 2 for BIVAR data. SAMPLE transformation of MULVAR, RCB, and BIBD data is not allowed.

For SAMPLE transformation of one or more random samples or two matched samples, the TRANS subroutine will be called once for each observation. All observations from sample 1 will be passed to TRANS, then all observations from sample 2, and so on through the last sample. No indication of the beginning or ending of each sample is given. All observations returned in X (the number being determined by ITEMS) will be considered part of the sample from which the original value came.

For SAMPLE transformation of bivariate data the TRANS subroutine will be called once for each bivariate observation, each observation consisting of two data items. The user may alter or delete the observation but cannot add items. The reason is because the observation is bivariate and MUST consist of two items. If any number of items except two is returned the

observation is deleted. If the user wishes to add pairs of items to form additional observations he must do so by RAW transformation.

The values of ITEMS and X have a special meaning when used in conjunction with the Chi-Square Test for Independence, a single random sample (1RS), and SAMPLE transformation. If ITEMS is zero the observation is discarded from the analysis, as usual. If ITEMS is 2 or greater the X(1) value is used to classify the observation into a particular row and the X(2) value is used to classify the observation into a particular column. X(3) through X(40) are ignored. If ITEMS is 1 the value in X(1) is used to classify the observation into both row and column. Since the test is a test on data classified according to two different criteria TRANS could examine the data and provide a second data value for classification purposes.

For RAW transformation the number of items returned is the number of items that will be divided among samples or variates according to the specifications on the CODING card. This number of items returned will also be the number of items written if TRANSFORMED output of the data has been requested.

Since the program assumes that the contingency table limits are, effectively, minus infinity to plus infinity, the TRANS subroutine can delete items that fall out of the range that the user wishes to consider.

Two examples of TRANS subroutines follow as a guide to the proper use of this facility. These examples are, by no means, meant to be representative of all the different ways that data can be altered.

```
      SUBROUTINE  TRANS
C
C     WORKS ON RAW DATA
C     GENERATES 2 NEW "OBSERVATIONS" FROM THOSE READ IN
C
      COMMON /USER/ ITEMS,X(40),LOCAT(4),RDNO
      INTEGER*4 RDNO
C     ITEMS READ IN WAS SPECIFIED AS 4
      X(5)=X(1)**2
      X(6)=X(5)-X(2)+X(3)
C     X(5) AND X(6) WERE GENERATED BY THIS ROUTINE
C     NUMBER OF ITEMS MUST NOW BE SET TO 6
      ITEMS=6
      RETURN
      END


      SUBROUTINE TRANS
C     WORKS ON DATA SAMPLE BY SAMPLE
      COMMON /USER/ ITEMS,X(40),LOCAT(4),RDNO
      INTEGER*4 RDNO
C     SINCE THIS IS SAMPLE TRANS, ITEMS WILL BE 1 ON ENTRY
C     CHECK FOR NEGATIVE X(1) AND DELETE IF SO
      IF(X(1).LT. 0.) GO TO 1
C     X(1) IS NOT NEGATIVE, TAKE SQUARE ROOT
      X(1)=SQRT(X(1))
      GO TO 2
C     SET ITEMS TO ZERO FOR DELETE
    1 ITEMS=0
    2 RETURN
      END
```

VI. TYPES OF TESTS

There are 15 nonparametric tests available. The criteria for each one are presented here. Other sections of this guide explain the meaning of the codes and abbreviations used. The tests are described in the following format:

TEST NAME

Keyword: TESTCODE, Data Types: DATACODE(S)

Other Required Specs: PARMSNEEDED

Optional Specs: PARMSOPTIONAL

A brief description of the test.

"TEST NAME" is the name of the test.

"TESTCODE" is the keyword abbreviation for the test.

"DATACODE(S)" is the keyword abbreviation(s) for the type(s) of data used by the test.

"PARMSNEEDED" is parameter card(s) that must be provided to perform the test.

"PARMSOPTIONAL" is optional parameter card(s) or keyword(s) that may be specified.

A. THE SIGN TEST

Keyword: SIGN, Data Types: BIVAR, 2MS

Other Required Specs: NONE

Optional Specs: UPPER, LOWER, 2-TAIL

The Sign Test is a simple method of testing whether one variate is less than, equal to, or greater than another variate. This is done by counting the number of "first" variate observations greater than the

corresponding "second" variate observations and determining the probability of a true null hypothesis from this count based on the binomial distribution with probability of 0.5.

B. THE McNEMAR TEST

Keyword: MCNEMAR, Data Types: BIVAR, 2MS

Other Required Specs: DISTRIB,LEVEL

Optional Specs: NONE

The McNemar Test for Significance of Change takes a bivariate sample (or 2 matched samples) and assumes that the first variate or sample represents a before condition and the second an after condition. Each observation less than the LEVEL specified is made 0 (zero), those greater than the LEVEL are made 1 (one). The LEVEL value is specified on the DISTRIB card. The test defaults to a LEVEL of 0.5 if the LEVEL is not provided by the user. The LEVEL represents the dividing line of the categories that observations are classified into. For example, if a questionaire with YES and NO answers was used as a source of data and YES was coded in the input as a 1 and NO was coded as a 2, a LEVEL specification of 1.5 would separate these two answers. The default level of 0.5 would place all of the observations into the 1 category. Values that are equal to the LEVEL value are discarded and do not participate in the test. The observations are counted into the classes (0,0), (0,1), (1,0), and (1,1) based on the before and after values. Those observations that represent changes from before to after, (0,1) and (1,0) are used to form a test statistic which is based on a binomial distribution with probability 0.5.

C. THE CHI-SQUARE TEST FOR PROBABILITY DIFFERENCES

Keyword: CHIPROB, Data Types: 2RS, KRS

Other Required Specs: CLASSES,COLUMNS

Optional Specs: NONE

The Chi-Square Test for Probability Differences forms an R by C contingency table by classifying the R different samples (rows) into C columns and calculating a test statistic with (R-1)x(C-1) degrees of freedom. The null hypothesis is that the probability of an observation falling into any particular column is the same for all samples.

D. THE CHI-SQUARE TEST FOR INDEPENDENCE

Keyword: CHINDEP, Data Types: 1RS, BIVAR

Other Required Specs: CLASSES,ROWS;CLASSES,COLUMNS

Optional Specs: SAMPLE transformation

The Chi-Square Test for Independence forms an R by C contingency table by classifying the supplied sample into one of R rows and one of C columns. The method of classification depends on the type of data. If a bivariate sample is provided the first item of each observation is used for the row classification and the second item is used for the column classification. If one random sample is supplied each item is used to classify each observation in both a row and a column, with one exception. If the user specifies SAMPLE transformation and two or more items are returned by TRANS, the first item is used for row classification, the second item is used for column classification, and any additional items are ignored. From this contingency table a test statistic with (R-1)x(C-1) degrees of freedom is calculated. The test determines the probability of

a sample classified by some scheme (rows) being independent of a second classification scheme (columns).

E. THE CHI-SQUARE GOODNESS OF FIT TEST

Keyword: CHIGOF or GOODFIT, Data Type: 1RS

Other Required Specs: CLASSES,COLUMNS;DISTRIB

Optional Specs: NONE

The Chi-Square Goodness of Fit Test is used to test if the sample data may be considered to be a sample from a population as specified on the DISTRIB card. The sample is divided into "C" different classes, where "C" is the number of columns specified on the CLASSES card, and the expected number of observation in each column is computed, assuming the sample had come from a population with the specified distribution. The distribution functions currently implemented are: the NORMAL distribution with specified mean and variance; the UNIFORM distribution with equal probability assigned to each column; the POISSON distribution with specified mean; and the BINOMIAL distribution with specified size and probability, P. A test statistic is calculated using the observed and expected numbers in each column and the probability of a true null hypothesis is determined from the Chi-Square distribution with (C-1) degrees of freedom. Poor selection of the classes in which the sample is divided can result in a poor test. The choice of these classes should be made with care, although making each cell the same size (lower to upper limit) will usually work reasonably well. If any expected number in any column is less than 4 a warning is printed because the test statistic may be biased. This would indicate that the test perhaps should be run with a different column breakdown specified.

F. THE MEDIAN TEST

Keyword: MEDIAN, Data Types: KRS, RCB

Other Required Specs: NONE

Optional Specs: Sample sizes for unequal sample sizes (KRS)

The Median Test is a test on "K" random samples with the null hypothesis being that all sample medians are the same. For an RCB experiment the data are tested for equal block medians, equal treatment medians, and equal replication medians. For "K" random samples the number of observations greater than the GRAND median of all samples is counted and a test statistic, which has a Chi-Square distribution with (K-1) degrees of freedom, is computed. The Median Test on RCB data is somewhat more difficult to describe and the user is directed to the section on the Median Test Extended in Conover.

G. THE WILCOXON SIGNED RANKS TEST

Keyword: SIGNRANK, Data Types: 1RS, BIVAR, 2MS

Other Required Specs: DISTRIB,MEDIAN for 1RS

Optional Specs: UPPER, LOWER, 2-TAIL

The Wilcoxon Signed Ranks Test can be used to test the null hypothesis that the expected value (median) of a random sample is greater than, less than, or equal to the MEDIAN value specified. Alternatively, the test can be used on two variates of a bivariate sample or the 2 samples of 2 matched samples, rather than a single random sample and a MEDIAN value. The differences between the sample values and the MEDIAN value, between variates, or between samples are calculated. The absolute values of these differences are taken and assigned ranks from smallest to largest absolute

difference. The sum of the ranks for those observations where the difference is negative is taken. The probability of a true null hypothesis is determined from this rank sum using a large sample normal approximation. This probability (called alpha hat) and the decision to accept or reject the null hypothesis may be invalid for samples smaller than 20 observations. If this is the case, a warning message is printed after the test is performed. If there are too many tied values, the alpha hat values may also be invalid. Since what constitutes "too many ties" is debatable, the program will not print a warning, but will report how many ties were encountered.

H. THE MANN-WHITNEY TEST

Keyword: MANN-WHITNEY or M-W, Data Type: 2RS

Other Required Specs: Sample sizes for unequal sample sizes

Optional Specs: UPPER, LOWER, 2-TAIL

The Mann-Whitney Test may be performed when the null hypothesis is that the Expected Value (Median) of the first sample is greater than, less than, or equal to the Expected Value of the second sample. All the observations are sorted into ascending order and assigned ranks regardless of sample. The test statistic is formed from the sum of the ranks of the first sample and the probability of a true null hypothesis is computed using a large sample normal approximation. A warning message is printed if either sample is smaller than 20 observations since the probability computed may not be correct and exact probability tables should be referenced.

I. MOOD'S TEST OF VARIANCES

Keyword: MOOD, Data Type: 2RS

Other Required Specs: Sample sizes for unequal sample sizes

Optional Specs: UPPER, LOWER, 2-TAIL

Mood's Test of Variances tests the null hypothesis that the variance of the first sample is greater than, less than, or equal to the variance of the second sample. The two samples are sorted together in ascending sequence, ranks are assigned to the observations without regard to samples, and a test statistic is computed from the ranks of the first sample. A large sample normal approximation is used to determine the probability of a true null hypothesis and this may be invalid if either sample has fewer than 20 observations. If this is the case a warning is printed. To the author's knowledge no exact tables for small samples exist, so if the samples have fewer than 20 observations and the probability given is close to the alpha level specified, a statistician should be consulted.

J. SPEARMAN'S RANK CORRELATION COEFFICIENT, RHO

Keyword: RHO, Data Type: BIVAR

Other Required Specs: NONE

Optional Specs: UPPER, LOWER, 2-TAIL

Spearman's Rank Correlation Coefficient, Rho, is a measure of correlation between the two variates of a bivariate sample. The null hypothesis tested is no correlation versus the alternative of positive, negative, or either direction correlation between the two variates. The two variates (hereafter referred to X and Y) are sorted into ascending sequence and ranked independently. The differences between the ranks of the X's and their corresponding Y's is used to calculate Rho. The probability of no

correlation is determined from a large sample normal approximation which may be invalid for 30 or fewer observations. In that case a warning message is printed and exact tables should be consulted to determine the correct probability.

K. KENDALL'S RANK CORRELATION COEFFICIENT, TAU

Keyword: TAU, Data Type: BIVAR

Other Required Specs: NONE

Optional Specs: UPPER, LOWER, 2-TAIL

Kendall's Tau is a measure of rank correlation, much like Spearman's Rho. The correlation coefficient, Tau, however, is computed differently and will, in general, be different from Rho. All of the bivariate observations are paired (forming a set of 2 observations, each observation consisting of 2 items) and the number of concordant and discordant pairs is calculated. A concordant pair is one in which both items in one observation are greater than their corresponding items in the other observation. A discordant pair is one in which one item in one observation is greater than its corresponding item in the other observation and the other item in this one observation is smaller than its corresponding item in the other observation. While Tau is computed as a measure of correlation, the number of concordant pairs minus the number of discordant pairs is used as the test statistic in a test of no correlation versus positive, negative, or any direction correlation. The large sample normal approximation of the probability of a true null hypothesis requires more than 40 observations to be considered valid.

L. THE TEST OF TREND USING SPEARMAN'S RHO

Keyword: TREND, Data Type: 1RS

Other Required Specs: NONE

Optional Specs: UPPER, LOWER, 2-TAIL

The Test of Trend using Spearman's Rho takes a single random sample and converts it into a bivariate sample by assigning the observation number (1,2,3,...) as the second variate of each observation. It then ranks this bivariate population on the supplied observations and computes Rho. A test of no trend versus an alternative of upward, downward, or either direction trend can be specified. Since Spearman's Rho is used as the measure of trend, the same large sample normal approximation is used and the probability of a true null hypothesis may not be correct for 30 or fewer observations.

M. THE KRUSKAL-WALLIS TEST

Keyword: KRUSKAL-WALLIS or K-W, Data Types: 2RS, KRS

Other Required Specs: Sample sizes for unequal sample sizes

Optional Specs: NONE

The Kruskal-Wallis Test is essentially a "K" sample Mann-Whitney Test and will give similar results if performed on only two samples. The null hypothesis tests if all samples can be considered to have come from populations with the same distributions versus a suitable alternative. All of the data are ranked without regard to the sample from which any particular observation comes. The sums of the ranks assigned to each sample are used to calculate a test statistic. The probability of a true null hypothesis is calculated from a Chi-Square distribution with (K-1) degrees of freedom, where "K" is the number of samples tested.

N. THE FRIEDMAN TEST

Keyword: FRIEDMAN, Data Types: MULVAR, RCB

Other Required Specs: NONE

Optional Specs: NONE

The Friedman Test is used to test the null hypothesis of equal treatment effects versus the alternative that at least two treatments have different effects. If a multivariate sample was specified, each observation is considered a block and each variate is considered a treatment, thereby converting the sample into an RCB sample with one replication per block-treatment combination. The sample is ranked within blocks, each block being ranked separately, and the sum of the ranks assigned to each treatment is computed. A test statistic with (K-1) degrees of freedom is calculated, where "K" is the number of treatments. The probability of a true null hypothesis is determined from the Chi-Square distribution.

O. THE DURBIN TEST

Keyword: DURBIN, Data Type: BIBD

Other Required Specs: NONE

Optional Specs: NONE

The Durbin Test is essentially a Friedman Test extended to incomplete blocks but restricted to only one replication. The observations in each block are ranked and the sum of the ranks in each treatment is computed. The test statistic calculated from the rank sums has (T-1) degrees of freedom, where "T" is the number of treatments. The probability of a true null hypothesis of equal treatment effects is computed from the Chi-Square distribution.

VII. JOB CONTROL CARDS

Two Job Control Language procedures have been provided to relieve the user of the necessity of coding the large number of cards needed to use this package. The user chooses the procedure to use depending on whether or not a transformation subroutine is being used. Although the procedure for transformations could be used for non-transformation runs, there is an overhead associated with the compiler and linkage editor which is necessary for transformation runs, but unnecessary otherwise.

The following control cards are used for the NO TRANSFORMATION case.

```
//jobname  JOB (standard job information...
//stepname EXEC STATPROG,ROUTINE=NONPAR
//STATPROG.SYSIN DD *
--  parameter and data cards  --
/*
```

To run WITH TRANSFORMATION, the following control cards should be used.

```
//jobname  JOB (standard job information...
//stepname EXEC STATRANS,ROUTINE=NONPAR
//FORT.SYSIN DD *
--  user's TRANS subroutine source deck --
/*
//LKED.SYSIN DD *
          INCLUDE SYSLIB(NONPAR)
          ENTRY MAIN
/*
//STATPROG.SYSIN DD *
--  parameter and data cards  --
/*
```

If the user desires to write data out on an intermediate unit for processing the data more than once by this program, or for processing by another program, he may take advantage of the supplied control cards for unit 10. Data written out on unit 10 may be reread any number of times or read by another program. However, any data written on unit 10 will be erased at end of JOB. It cannot be saved for processing at a future time. If saving of the data for an extended period is desired or necessary the user will have to supply the proper control cards for a unit other than 10 (or that will override the procedure specifications for 10). Since it is not expected that every user will be able to set up control cards, members of the statistical laboratory staff will help the user with any special needs.

VIII. FORTRAN UNIT NUMBERS

This program makes use of the following FORTRAN Input/Output Units and their use by the user in his TRANS subroutine or as data input/output units is prohibited, restricted, discouraged, or incouraged.

UNIT NUMBER	USE-RESTRICTION
1	Parameter card input-May be used for input of FORMAT data
2	Punched output-Prohibited for use as data input or output; use in TRANS is discouraged
3	Printed output-Prohibited for use as data input; may be specified as output unit to obtain listing of data; use in TRANS is discouraged, but may be used if necessary.
10	Data input or output-Use is entirely optional and depends on user parameter cards. If data input or output on this unit is specified, TRANS must not use it.
11 12 13 14	Intermediate work units-Use by user is completely prohibited, since such use would destroy their contents
20	Use count data set-Must not be used at all

Of all the above unit numbers only units 1 and 10 can be specified for inputting of data and if unit 1 is specified, the data must be in EXTERNAL (FORMATTED) form. Only units 3 and 10 can be specified for outputting of data and if unit 3 is specified the data are printed in a suitable format.

IX. ERROR MESSAGES

All error messages (and warnings) produced by this program have a message number and severity code associated with them. The message number is 7 characters long and is of the form, IKMSxxI, where xx is the numerical designation of the error (00-99). This 7 character message number is included as part of the text of the error message when printed. The severity code, which is not printed with the message, represents how severe the error is. The severity codes are summarized as follows:

Severity Code	Meaning
0	Strictly informational or warning message. Program continues.
8	Parameter specification error. No processing is done on the data for the set of parameters in error.
12	Error in the input data or in the performing of a test on the data. Further processing on the set of data is bypassed.
16	Program error. Program either bypasses processing on the current set of data or terminates all processing completely.
-	No severity assigned. Message is usually associated with one or more other error messages.

The highest severity code generated for the entire run is used as the step return code.

NUMBER	SEVERITY	MESSAGE TEXT/EXPLANATION
IKMS00I	0	END OF IKMSTAT RUN
IKMS00I	0	TOTAL EXECUTION TIME WAS xxx.xx SECONDS End of run messages. IKMSTAT is system name for program.
IKMS01I	8	INVALID PARAMETER CARD. Preceeded by card in error. Parameter card is completely unrecognizable.
IKMS02I	8	INVALID SPECIFICATION IN PARAMETER xx Parameter card has illegal specification in positional parameter xx (not counting the keyword). If xx=0, syntax of card is in error (such as a label of 9 characters).
IKMS03I	-	END OF FILE ENCOUNTERED WHILE READING PARAMETERS. End of File (/*) card may be misplaced or parameter cards may be missing. Job is terminated since no more input is present.
IKMS04I	-	1 PARAMETER SYNTAX ERROR DETECTED.
IKMS05I	-	nn PARAMETER SYNTAX ERRORS DETECTED. All parameter cards for this set have been read and the number of errors noted have been found.
IKMS06I	8	EXCESSIVE RIGHT PARENTHESES
IKMS07I	8	EXCESSIVE LEFT PARENTHESES The supplied FORMAT specification was in error for the reason noted.
IKMS08I	8	INVALID NO. OF SAMPLES FOR SAMPLE TYPE The number of samples specified is invalid, such as specifying "k" Random Samples and only 1 sample.
IKMS09I	8	INVALID TEST FOR TYPE OF DATA A test was specified that does not work on the given type of data.
IKMS10I	8	MISSING OR INVALID RCB/BIBD PARAMETERS Insufficient information about the data was specified, i.e., no block or treatment size.
IKMS11I	8	NO SAMPLE SIZES PROVIDED WHERE REQUIRED The size of 1 or more samples was given, but sizes were not given for ALL samples.

NUMBER	SEVERITY	MESSAGE TEXT/EXPLANATION
IKMS12I	8	MISSING MEDIAN/LEVEL SPECIFICATION The median/level parameter on the DISTRIB statement was not specified, but is required for the desired tests.
IKMS13I	8	CLASS INFORMATION IMPROPERLY SPECIFIED Too few, none, or non-increasing contingency table classes (either row or column) were specified.
IKMS14I	8	DISTRIBUTION PARAMETERS INVALID FOR SPECIFIED TYPE OF DISTRIBUTION Inconsistant parameters were supplied, such as a negative variance.
IKMS15I	0	EQUAL SAMPLE SIZES ASSUMED. No sample sizes were given for multiple random samples. Program assumes that all samples will be the same size.
IKMS16I	0	END OF DATA WILL DETERMINE SAMPLE SIZES. No sample sizes were given. Program will determine sample sizes when end of data is reached.
IKMS17I	8	TRANSFORMATION INVALID FOR THIS DATA SAMPLE, rather than RAW transformation specified for RCB or BIBD type data.
IKMS18I	8	NO TESTS SPECIFIED No tests to be performed were specified (TESTS card may have been omitted).
IKMS19I	-	INPUT BYPASSED UNTIL END OF DATA Because of some error, the input is bypassed until the END OF DATA for this test is encountered, or until the TITLE card for the next set of parameters is read.
IKMS20I	12	CONVERSION ERROR OCCURRED IN COLUMN nn Message is followed by contents of buffer with a dollar sign ($) under the character in question. A formatted data conversion error (IHC215I) has occurred during data input in card column nn. Further processing on this set of data is terminated.
IKMS21I	0	END OF DATA CARD MISSING. ASSUMED PRESENT. During the input of data, either actual end of file or the next TITLE card was encountered instead of the END OF DATA card. Program continues, assuming the card was present.

NUMBER	SEVERITY	MESSAGE TEXT/EXPLANATION
IKMS22I	-	ERROR IN PERFORMING THIS TEST CAUSES ANY REMAINING TESTS TO BE SKIPPED The last test completed returned an error condition. Further testing of this data is suspended.
IKMS23I	8	END OF FILE WHILE READING FORMAT STATEMENT While reading in the data FORMAT statement, end of file was encountered. Possibly caused by excessive left parentheses in the format.
IKMS24I	12	INCORRECTLY LABELLED DATA - labels The input data has attempted to place 2 data items into the same block-treatment-replication combination.
IKMS25I	0	INPUT SAMPLE SIZES DON'T MATCH SPECIFIED SIZES. INPUT SIZES USED. Sample sizes were specified, but more (or less) than the specified number of items was provided in the input data.
IKMS26I	12	INCOMPLETE BLOCKS IN RCB DATA Not all block-treatment-replication combinations were present in the input data.
IKMS27I	12	MISSING REPLICATIONS IN BIBD DATA At least one Incomplete Block is too incomplete, i.e., less than the specified number of treatments per block or blocks per treatment were present in the input data.
IKMS28I IKMS29I IKMS30I	No messages assigned to these numbers.	
IKMS31I	12	UNEQUAL SAMPLE/VARIATE SIZES AFTER TRANSFORMATION The user supplied transformation subroutine has added or deleted an unequal number of observations in each sample of variate. This can also occur if inequality exists in the input data, since this condition is not checked after input processing.
IKMS32I	12	TOO FEW xxxxxxxx CLASSES SPECIFIED Insufficient row or column classes were provided by the user. xxxxxxxx is ROW, COLUMN, or the user provided label associated with the row or column.

NUMBER	SEVERITY	MESSAGE TEXT/EXPLANATION
IKMS33I	12	TOO FEW xxxxxxxx DICHOTOMIZATION LEVELS SPECIFIED Not enough upper dichotomization limits were provided. xxxxxxxx is ROW, COLUMN, or the user label associated with the row or column.
IKMS34I	12	INVALID SAMPLE SIZE(S) The data contained either a sample of size zero, or a sample (or combination of samples) which was too small to perform the requested test on.
IKMS35I	12	NO MEDIAN VALUE SPECIFIED A single random sample was to be used in the Wilcoxon Signed Ranks Test, but no median value was specified for testing.
IKMS36I	12	TOO MANY TIED VALUES FOR THIS PROGRAM TO HANDLE A set of ties has completely filled the available working storage array. Testing cannot continue. Program can currently handle 800 ties for each tied value.
IKMS37I- IKMS89I	No messages assigned to these numbers.	
IKMS90I	0	*****BECAUSE OF SAMPLE SIZES, ALPHA HAT MAY NOT BE CORRECT. VERIFY WITH TABLES. ***** The alpha hat value has been determined from a large sample approximation and the sample is small enough to question its validity. The test statistic is valid, however, and the exact alpha hat value should be found from tables.
IKMS91I	0	*****EXPECTED VALUE(S) MAY BE TOO SMALL FOR VALID TEST. ***** At least one of the expected values was below 4.0. The test statistic may be biased. If any expected value is less than 1.0, the test statistic is definitely biased.
IKMS92I IKMS93I	No messages assigned to these numbers.	
IKMS94I	16	PREMATURE END OF FILE ON UNIT nn End of file occurred while searching for the median of the supplied data. The program was unable to read one-half of all observations after sorting.
IKMS95I	12	NO DATA TO SORT The data sort subroutine was called to sort zero items.

NUMBER	SEVERITY	MESSAGE TEXT/EXPLANATION
IKMS96I	12	INVALID UNIT PARAMETER IN CALL TO RANKER The sort subroutine, RANKER, was called with conflicting parameters, i.e., input and output the same, input and work unit the same, or both work units the same.
IKMS97I	12	FAILURE TO CONVERGE IN CHI-SQ FUNCTION Consult IBM SSP Programmer's Manual (H20-0205) for information on routine CDTR, where the error has occurred.
IKMS98I	12	INVALID INPUT TO CHI-SQ FUNCTION The parameters specified in the function call to CDTR were invalid, i.e., outside allowable ranges.
IKMS99I	16	PROGRAM ERROR. Program has generated an invalid data type or negative return code.

AN INVESTIGATION AND IMPLEMENTATION OF NONPARAMETRIC STATISTICAL TESTS ON A DIGITAL COMPUTER

by

KEITH ERIC MOE

B. S. E., Arizona State University, 1969

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1970

In recent years, statistical analysis has come into widespread use, primarily because the digital computer has taken the burden of laborious computation out of man's hands. At the same time, the use of digital computers has put a burden on the programmer required to write statistical programs. It soon became evident that there was much duplication in statistical programming when a separate program was written for each experimental situation.

To reduce the programmer's burden and unnecessary duplication, comprehensive statistical programs, commonly referred to as "packages", have been developed. Each of these packages usually covers some area of statistics thoroughly, with special provisions for user modifications when the experimental situation requires such. These packages, however, have been lacking in one, very important area of statistics. Most packages use only parametric statistical methods, i.e., methods in which the distribution of the data is assumed to be known or estimable. Hence, the analyses provided by these programs are not reliable if the user feels that the assumptions required are not met.

This report describes the design and implementation of a statistical package which does not require the underlying distribution(s) of the data to be known. The package is a set of nonparametric statistical tests. Nonparametric statistical methods do not need as many assumptions as are needed in parametric statistical analyses. Hence, in many experimental

situations, the results provided by nonparametric analysis may be more reliable than those produced by parametric analysis. One nonparametric test which has been implemented in this package can be used to test if data meet the assumptions of parametric methods.

This package, which the author believes is the first of its kind, was designed to have much of the flexibility that its parametric predecessors have. The flexibility was extended in several instances over that of existing packages to overcome some of the problems encountered in their use. This package was designed for, and implemented on, an IBM System/360 running under O.S., using FORTRAN IV and Assembly Language. It is the hope of the author that this package will make computer statistical analysis <u>really</u> comprehensive and complete.