A COMPUTERIZED SYSTEM FOR PROJECTING CROP DATA ON A COUNTY AND DISTRICT BASIS

by

WALTER LYNN TURNER

B. S., Kansas State University, 1974
M. B. A., Kansas State University, 1976

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Agricultural Economics

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1977

Approved by:

Leonard W. Schruben

Major Professor

ACKNOWLEDGEMENTS

The author wishes to express his grateful thanks to Dr. Leonard Schruben, Department of Agricultural Economics, for his valuable advice and guidance in the preparation of this thesis.

Acknowledgement is due Dr. Frank Orazem, Department of Agricultural Economics, and Dr. Jim Gentry, College of Business Administration, for their critical review of this manuscript.

The author is indebted to Dr. Doyle Jeon for his help and guidance with the statistical aspects of this thesis.

Sincere appreciation is expressed to Ken Ward and Karl Hartronft, consultants at the Kansas State University Computing Center, for their help with the interactive and Assembler language aspects of the programs in this thesis.

The author is very grateful to Mara Rasure and June Carlson for their assistance in the preparation of the user and programmer's manuals incorporated in this thesis as sections II through V.

The Agricultural Experiment Station, at Kansas State University, funded the research on which this thesis is based.

LIST OF FIGURES

Page

TABLE OF CONTENTS

TABLE OF CONTENTS CONT'D

Page

TABLE OF CONTENTS CONT'D

Page

TABLE OF CONTENTS CONT'D

THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH THE ORIGINAL
PRINTING BEING
SKEWED
DIFFERENTLY FROM
THE TOP OF THE
PAGE TO THE
BOTTOM.

THIS IS AS RECEIVED
FROM THE
CUSTOMER.

I. INTRODUCTION

A. Problem

Reliable projections of crop data produced by computer and used as a basis for regional analyses are cumbersome and complex. The complexity results, not only from the inherent difficulty in developing a valid projection model, but also from the sheer volume of data involved. For example, a regional transportation study may involve 400-500 counties. When this figure is multiplied by fifteen years or more of documented crop data times the number of types of crops considered, it is easy to see why a data handling problem can result. This is a problem that is becoming of increasing importance as the number and scope of studies of the transportation system, for example, increase. In view of these factors, and the experiences the author has had in regional transportation studies [14], this paper is oriented toward the development of a computerized system for the projection of crop data in a form suitable for use in such a regional study.

System Characteristics

As most regional analysis problems require massive amounts of input data, so also is the size of the output of projected data impressive. Because of this, the output should be in a form such that photocopies of the printout could be used directly in a report without the necessity of retyping the output. This also means that, for clarity and readability, the output must contain appropriate descriptive titles on each page of output. (For example, see the sample program output in section IV.) Also, as the

requirements of various studies differ in terms of the number of counties involved, the output must adjust to compensate for differing numbers of counties being projected. For example, whether a district has five or fifteen counties, the program must calculate the paging of output, both lengthwise and widthwise, and see that each county's output is labeled appropriately on each page of output.

The computer program should be easy to use, both in terms of readying input data and in the amount of user time consumed in selecting the best projection model to use in making each projection. At the same time, the user should be able to feel confident that, from the models used, the best projections were selected as output, either by printing all projections or by screening them and printing the best.

Finally, the program should be flexible. Thus, it should be constructed so that various types of projection models could be tested in the program. As models are shown to be effective, they can then be added to the initial repertoire of models available. Thus, an analytical tool becomes immediately available and can be made even more effective at a later date by adding improvements to the basic program.

Projection Models

Three basic models were selected for inclusion in this initial crop projection program. They are 1) linear, 2) logarithmic, and 3) exponential, all of the simple type--one independent and one dependent variable. The principal consideration in this selection is that simple regression (i.e. projection of crop data simply from historical crop data, without considering other variables) is more suitable and quicker (in terms of model selection and data collection) than other multiple variable models. This is not to

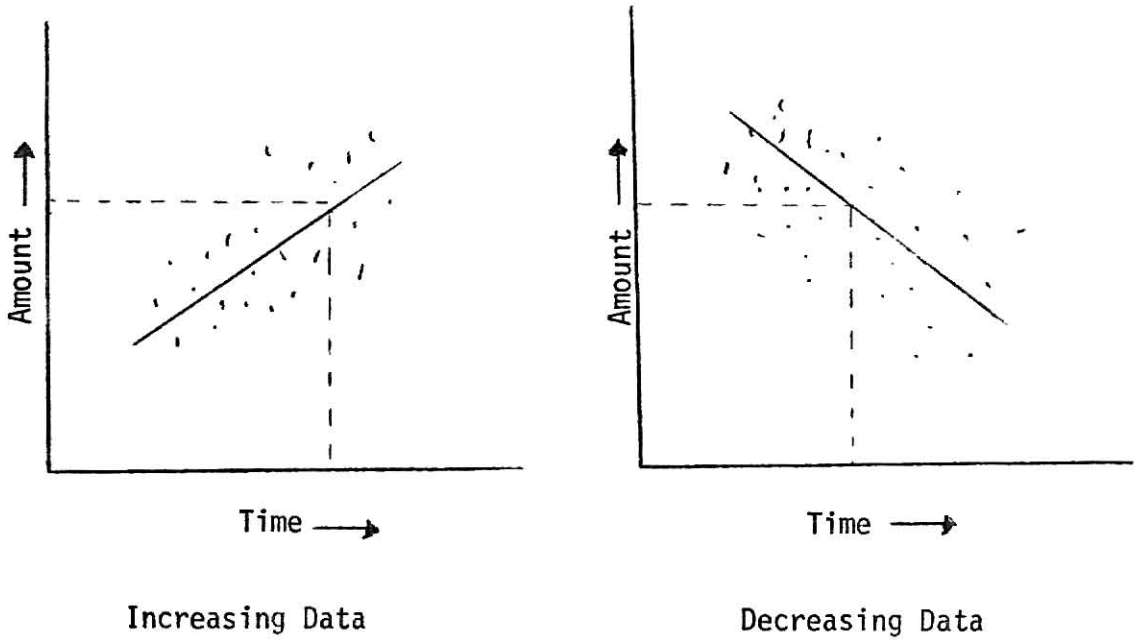
THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

say that simple models provide a better basis for projection than more complex models, but for large regional analysis problems, they may often be the only ones that are practical in terms of project time and cost constraints.

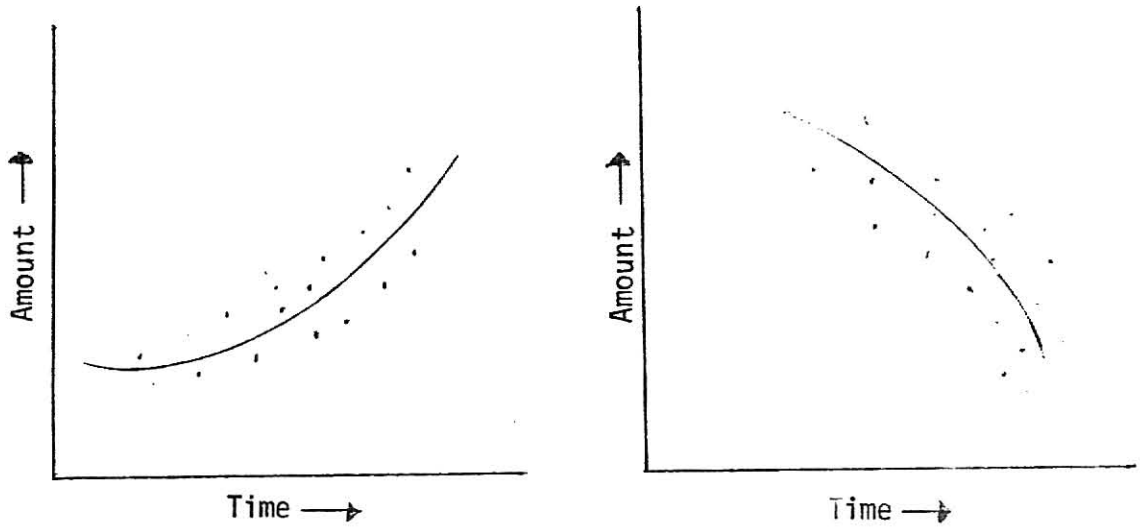
The linear model is of the form $Y = a + bx$, where Y is the crop data prediction and x is the year for which the prediction is being made. (For a detailed explanation of this model and the ones that follow, and for an insight into the regression technique used in the projections, see references [2,3,4].) This roughly translates into fitting a line to historical crop data so as to minimize the sum of the squared deviations of the observations from that line. Projected values were obtained by moving along the line to a future year and then reading the value of the associated crop data for that year. (See figure 1.)

FIGURE 1. Fit of Trend Line to Data--Linear Model



The logarithmic model is of the form $Y = a+b(\log_e x)$. The projected data is based on fitting a line to values plotted against transformed values of time. The resulting line is actually a shallow curve following the trend of the data. (See figure 2 below.)

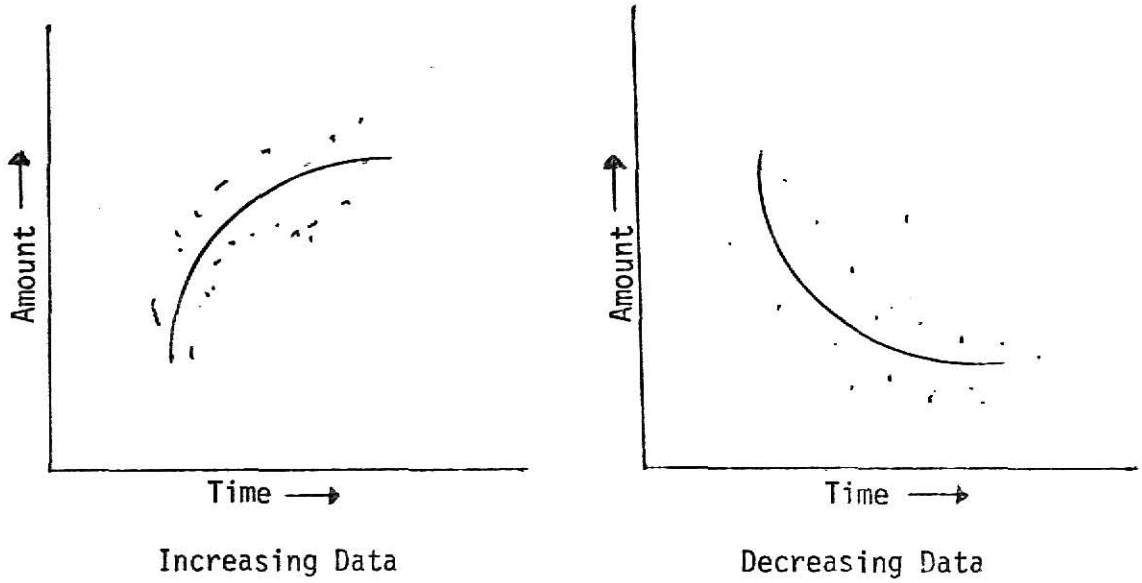
FIGURE 2. Fit of a Trend Line to Data--Logarithmic Model



Increasing Data Decreasing Data

The exponential model is of the form $Y = a+bx^c$, where c is an exponent and can vary within the program from 0.4 to 1.4. (The limits of 0.4 and 1.4 for the exponent were chosen because these growth rates are on the extremes of normally occurring long-term agricultural growth rates.) If the exponent is less than one the projection will increase or decrease at a decreasing rate. (See figure 3 on following page.)

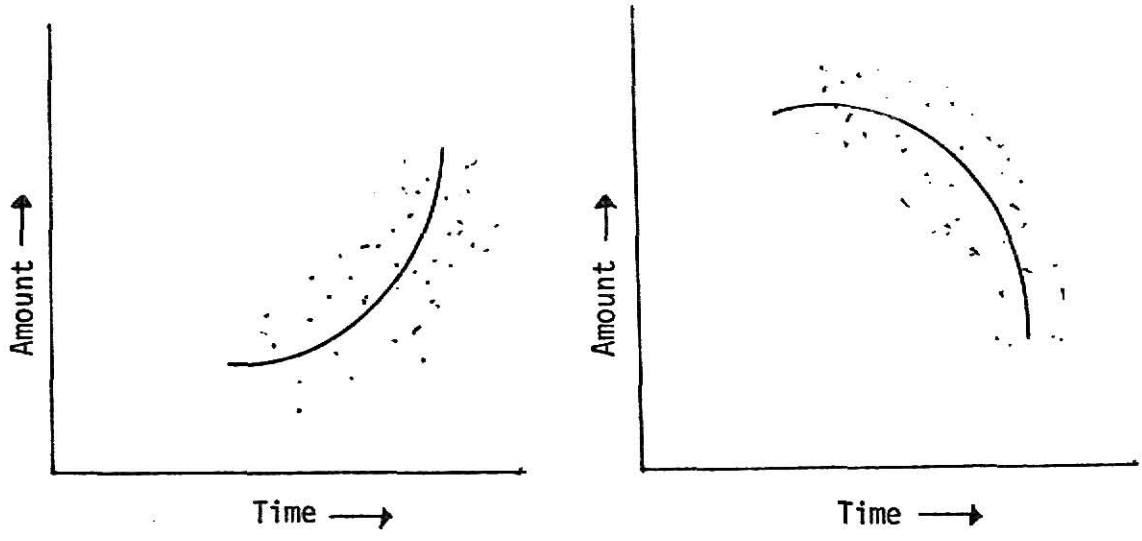
FIGURE 3. Fit of a Trend Line to Data--Exponential Model
(Exponent less than 1)



Increasing Data Decreasing Data

If the exponent is greater than one, the projection will increase or decrease at an increasing rate. (See figure 4 below.) Of course, if the exponent is one, the exponential model becomes equivalent to the linear model.

FIGURE 4. Fit of a Trend Line to Data--Exponential Model
(Exponent greater than 1)



Increasing Data Decreasing Data

Many regression programs already exist that are applicable to the regional-type studies. The various approaches and their advantages and disadvantages will be reviewed in part B of this section.

B. Literature Search

The most readily available crop-projection regression programs are available as part of a statistical package. This is a group of programs, often already compiled, that can be purchased by computing centers for performing statistical computations. The ones generally available are cited below. Programs performing multiple regression (two or more independent variables) are cited because, if two variables are analyzed, they would function as a simple regression program.

System/360 Scientific Subroutine Package [7]

This package of subroutines often comes in a compiled form on tape such that the subroutines can be called from a user program as though they were internal subroutines. The subroutines are also available in a non-compiled form in both Fortran and PL1. This package contains both REGRE for multiple regression and STEPR for multiple stepwise regression. The package contains other subroutines which must be called to control output. Even so, the user has little control over the appearance of the output.

Statpack 2 [15]

This package consists of APL routines that work on a computer terminal. The output consists of the R^2 , the standard error, and t values for the

independent values. These routines are faster to use than the others since they can be summoned via a terminal. The latest APL version allows batch input of data that can later be processed from a terminal. Thus, the only drawback to this package is the format of the output and the types of regression models offered.

Biomedical Computer Programs (BMD) [5]

This package was originally designed for biomedical applications, but now finds use in many other areas. BMD also provides a multiple regression and a stepwise regression program. It differs from the Scientific Subroutine package in that each program is complete in and of itself, and from the Statpack in that the programs will handle multiple data sets. The programs in the BMD package also provide, via options, the following: means, standard deviations, R^2's, error scatter plots, and covariance/correlation matrices. The user is allowed to instruct the computer as to the variables to include in the model. Many different combinations can be specified. Also, the input data can be transformed or combined via option cards. In the stepwise regression program, the user can specify the order of variable entry into the model.

Again, as in the Scientific Subroutines and Statpack 2, output is not in a suitable form for regional analysis. (The output needs to contain titles, have the district's counties in adjacent columns, and have a district column summing the quantities for the counties within the district.) Also, each model would have to be specifically stated for each county, and, if the stepwise function were to be utilized, the base data for other variables would need to be collected.

Statistical Package for the Social Sciences (SPSS) [10]

This package provides essentially the same type of regression services as BMD. It suffers from essentially the same faults as BMD when considered for use in a regional transportation analysis.

Statistical Analysis Subroutines (SAS) [1]

This is one of the most comprehensive packages available. SAS offers four regression programs. Briefly they are:

a) RSQUARE - this program automatically combines k variables into 2^k linear models, trying each of these and then printing each model tried and the results obtained, starting with the lowest R^2 and ending with the highest.

b) STEPWISE - performs stepwise regression using one of five built in models of variable inclusion.

c) GLM - allows the user to input equations used in combining or transforming the variables.

d) AUTOREG - useful for ordered and equally spaced time series data. This program transforms the data using parameters obtained from solving a series of equations using lagged versions of the original data (q items) This continues in steps with increasing lags until the last lag used equals t-q. (See also, the Box Jenkins model [2,9]).

This package is also not well suited to regional analysis studies due to output format. However, adaptation of the Autoreg procedure as a subroutine could prove useful in those situations with equally spaced time series data.

In addition to the statistical packages above, several universities have developed applicable programs. (See [8,16].) One program in particular, known as the Master Projection Program, was designed at Kansas State University

solely for the purpose of projecting time series data and is very applicable to crop projection [12]. However, for the particular application of regional transportation studies, the Master Projection does not meet the necessary criteria as stated in part A of this section. Two programs, based on the Master Projection program, were developed by the author to meet those criteria and will be discussed in the next part of this section.

C. Methods

In order to meet the criteria discussed in part A concerning ease of use, useable output, and flexibility, two programs were developed. One, Autoreg Term, is a Fortran H interactive terminal program, that projects one data set at a time. It was designed to meet the criteria of flexibility. It does this by providing a smaller program in which various projection models can be developed and tested in an interactive environment which greatly speeds the testing process. Autoreg Term is a simpler version of its sister program, Autoreg Multi--which handles multiple data sets. Both programs contain the projection models discussed in part A of this section.

The basic equation used in the models was Y' = Ybar - by(Xbar) + by(x) or Y = a+bx, where:

$$a = Ybar - by(Xbar)$$

$$b = \frac{\Sigma\, xy}{\Sigma\, x^2}$$

Ideas concerning the data transformations used to produce the other models were obtained from a program by Dawson [4] and the Master Projection program [12].

Also, for flexibility, both programs are structured in subroutines such that adding a new projection model entails only the addition of a few control statements in the main routine and the inclusion of another subroutine. Once a subroutine for a particular type of projection has been developed for Autoreg Term, it is a simple matter to adapt it to fit Autoreg Multi. However, because Autoreg Term is interactive, it is much easier to test projection models in it. Also, it serves as a fast way, in terms of user time, to project a data set.

Autoreg Term

This program was designed to be especially easy to use. The user must simply answer questions asked him by the program. If he inserts an invalid answer, the program will indicate it and give him another chance on most errors. Also, by answering one question with a YES, the user can get a short set of instructions typed at the terminal.

The program has two modes, automatic and manual. In the automatic mode, the program tries the models discussed in part A according to a built-in decision tree. In the manual mode, the user can control which of the models in part A are used. For further details in the use and construction of Autoreg Term, see sections II and IV, which contain a detailed user's guide and programmer's guide for Autoreg Term.

Autoreg Multi

This program meets all of the criteria outlined in part A--the flexibility as discussed above, and the ease of use and useability of output--with the following features:

1) The program automatically centers titles.

2) If a title or footnote is only one card in length instead of two, no blank second card must be input.

3) The names of the counties are automatically truncated to eight characters and they are centered above their associated data in the output tables.

4) The program is designed to provide a sum of the counties under the title 'DISTRICT'. Obviously, if the data for each county in a crop reporting district were input then the column titled 'DISTRICT' would serve as a projection for that crop reporting district.

5) The best projection models to try are selected by an internal decision tree, thus one run guarantees the best projection from among the various models available within the program.

6) Each page of output has a title block that also includes the district number.

7) All paging of output is determined automatically.

8) Three projection sets are generated containing the best, second best, and third best projections, based on the R^2 criterion. Thus, if other constraints preclude the use of the first projection set, two other sets are available.

9) At the beginning of each district's county projections, a statistical summary is provided.

10) Projections less than zero are automatically set to zero as negative crop production, yield, etc. is not possible.

11) Scaling of the input years allows correct regressions although years of data are missing. If only some counties are missing data for a given year, then an option, when turned on, will use trend to

calculate the values of the missing data prior to performing the projections.

12) The program tries a given county's base data in all the projection models (either implicitly or explicitly) before starting on the next county. By making projections in this manner, rather than trying a given projection model on all counties at the same time (as the Master Projection program does), some of the values within a projection set can be increasing while others are decreasing. This permits the reflection of shifts in land use as in the real world, often some production of a given crop in a crop production district may be increasing in some counties while production in others may be declining.

13) Multiple districts can be processed in one run. (Each district can contain up to 25 counties which will accommodate all U. S. crop reporting districts.)

For further details in the use and construction of Autoreg Multi, see sections III and V, which contain a detailed user's guide and programmer's guide for Autoreg Multi.

D. Results

Examples of output from Autoreg Term and Multi can be seen in sections II,III,IV, and V. See the parts entitled: Sample Terminal Session (section II) and Sample Program Output (section IV).

Autoreg Term

The output from this program varies depending on the options selected by the user. The end result, however, consists of a page showing the base

data. Following this is a key to type of models used in the projections printed, followed by various statistics. The statistics presented are alpha and beta (the a and b in the equations discussed in part A of this section), the mean of the base years (XBAR), and the mean of the base data (YBAR).

The computer time consumed by the program is small. It does, however, vary depending on the number of projections made by the internal decision tree, and the number of sets of base data process by the program.

The output of this program is in much the same form as the computer program packages discussed earlier. The advantage of this program lies in the requirements on the user to obtain the output and the structure of the program such that other models can be tested in a form suitable for later inclusion in Autoreg Multi.

Autoreg Multi

The output from Autoreg Multi is separated by district. Each district has its own title page showing the state, the district, the number of counties within the district, the number of base years of data and the number of years of projected data for the counties in the district.

Following the title page, each district's output contains one or two pages of statistical summary. This consists of the statistics for each projection set for each county in the district, all grouped together for easy comparison. The statistics given are the R^2 , alpha and beta of the trend line, and the mean base data value (YBAR). The type of projection model used to make the projection is also given.

Following the statistical summary are the three projection sets (see part III in section IV for further explanation). Each is labeled and put out in

order, with set one which contains the highest R^2 projections, appearing first. Every projection set also has its own set of statistics, similiar to those in the statistical summary.

The time consumed by Autoreg Multi also varies according to the features engaged. However, as a rough estimate, a district takes less than six seconds on an IBM 370 model 158 computer.

E. Conclusion

Although Autoreg Term and Autoreg Multi work well enough for the functions for which they were designed, in certain instances better techniques are needed. For example, Harper county, processed in the sample output for Autoreg Multi, had .0354 as the R^2 (coefficient of determination) of its best fitting trend line which clearly was not an excellent fit. In order to incorporate models and techniques to address situations like this, a second version of Autoreg Multi is currently being developed that contains many additional features. Many of them will also occur in a second version of Autoreg Term as a by-product of using Autoreg Term to develop them for Autoreg Multi.

The planned additional features are:

1) In order to allow a user to better interpret a low R^2 in terms of projection reliability, an F statistic will be included in the statistics for each projection.

2) To aid a user in grasping the real world implications of the R^2 of a projection, an option will be provided that, if called, will plot the base data, projected values, and upper and lower confidence bands at various levels of confidence. In addition, if desired, the values comprising the confidence band plots will be printed. From

these values, a user can obtain maximum-minimum figures for any projected year.

3) Also, to increase the R^2 values, other models will be included for the program to select from in making projections. At present, plans include the inclusion of a fourier and a polynomial model in an attempt to allow the models to better fit the cycles that often occur in crop data, as well as the overall trend of the data.

4) A new method of storage of projections will allow the base data to be projected an unlimited number of years into the future. This same method will also allow all projections made to be printed if the user so desires.

5) An option will allow the output to be punched so that it can be used directly in other programs.

The present version of Autoreg Multi is but the first step in developing an accurate, efficient method of projecting crop data. It is hoped that later versions improve forecasts while maintaining ease of use and low computer usage costs.

II. AUTOREG TERM USER'S GUIDE

I. Accessing Autoreg

Autoreg is available through the KSU computing center via any terminal. To sign on a terminal complete the following steps:

1) Hit RETURN
2) Type LOGON CMS CMS APL
3) Type in Account Number when asked
4) Type in Social Security Number when asked
5) When you receive an R; then access the program

An example of the sign on procedure is shown on the next page. The social security number and account number were typed over the blacked out spaces.

To run Autoreg, type: FMRETR AUTOREG EXEC (U VMKN9. This will retrieve the program and make it available to your terminal. When this has been done the terminal will signal by typing R;. To start Autoreg, type AUTOREG. (This starting procedure is shown on page 7.)

II. Error Corrections

After every line of answer you type on the terminal be sure to hit the RETURN button. Otherwise, the terminal will wait until you do. If you know you have made an error, try to correct it before you hit RETURN.

If you enter a wrong letter, type @ immediately after it, then type the correct letter. If 3 letters are wrong type 3@s followed by the correction. For example: you wanted TYPE but put in TIGH. Enter @@@YPE so the input line would look like: TIGH@@@YPE. If a line is a hopeless mistake, type ¢, hit RETURN and type the line over. (If you make a mistake on the data, a ¢ will only erase back to the last #. See next section.)

Example of Terminal Sign-On Procedure

```
VM/370 ONLINE     LJP359 OSYOSU

LOGON CMS CMS APL
ENTER ACCOUNT NUMBER
**********
ENTER SS NUMBER
**********
LOGON AT 11:17:32 CST TUESDAY 03/15/77
CMS V3 PLC 8 - 12/26/76 16:42
Y-03/14/77 08:13  *  THE DEFAULT A-DISK SIZE IS NOW 4 CYLINDERS.
FORMATTING DISK 'A'.
R;
```

III. Using Autoreg

An example of most of the following actions can be seen in a sample terminal session shown at the back of this section.

A. Obtaining a Brief Copy of Instructions

To get a brief copy of instructions, simply answer 'Yes' when the program asks if you want a copy of instructions. (See p. 7 for an example.) A copy of the instructions you will receive can be found in the programmer's section under subroutine Instrc.

B. Selecting Automatic or Manual Mode

Next, the computer will ask you to select the mode. If you answer AUTO, the computer will run linear, log and exponential regressions with exponents it selects to give you the best equation for the base data. (See subroutine Expflw in programmer's section for more details.)

If you select automatic, the computer will select the projections to be run according to the following scheme:

1st - Linear

2nd - Logarithmic, if linear RSQ (LINRSQ) is less than .90

3rd - Exponential, if log RSQ (LOGRSQ) is less than .80, exponential regression is performed using a built-in decision tree until all exponents between 0.4 and 1.4 are tried implicitly.

If you selected MANUAL, every time the line: Reg. Type? appears, input one of the following projection requests: LIN, LOG, EXP, STP,

where: LIN = Linear projection of the form Y = A+BX

LOG = Logarithmic projection of the form Y = A+B*LOGX

EXP = Exponential projection of the form Y = A+B* (X to the C power) (You will be asked to input the exponent (C))

STP = Terminate projections on present set of data

Examples of these mode selections can be seen on p. 7 and p. 9 .

C. Setting RSQ Cutoffs

If you selected automatic, decide if you wish to override the existing RSQ cutoffs which are: LINRSQ = .90 and LOGRSQ = .80. (If a projection RSQ doesn't exceed the cutoff RSQ, then the next type regression is tried until the cutoff is exceeded or all regression types have been tried.) If you do answer yes, then, respond to LINRSQ = with the cutoff for trying a log projection. (Write as a whole number -- ex: for .90 RSQ, input 90.) Respond to LOGRSQ = with the cutoff for trying exponential projections.

D. Printing a Copy of Each Regression

If you wish to see a copy of each projection as it is made, answer the question the program asks relating to this with a yes. (See p. 7 for an example.) This feature would be useful in conjunction with the manual mode to see a projection with a specific projection type, such as Exponential 1.3 because, otherwise, if this projection wasn't one of the best three, (best in terms of r^2) you would not see the actual projected numbers.

E. Input of Data

1) Select the number of years into the future that you wish projections to be made for, up to a maximum of 100 years. Input this figure when asked by the program. Write numbers below 10 as 09 etc.

2) Decide whether to input the years of the base data individually as 1963, 1964 etc., or as a range in the form: xxxx to xxxx. Years must be input individually if a year's data is missing. (Omit the year number for those years that data is not available for.) (See p. 7 for example.) If you elect to input the years individually you will first be asked to input the number of years of base data. (Write numbers below 10 as 09 etc.) Next, you will be asked to input the years. The years must be integers, and have a # sign between them. If the years take more than one line to input, at

the end of the line, hit RETURN and the computer will wait for you to input the rest of the years on the next line(s). Don't put a # sign after the last year on the line. The # sign serves to stack the input for the terminal--making it equivalent to the input of one year per line and then hitting RETURN.

3) When asked to input the data do so the same way as the years except that each data item must be followed by a decimal point. The # still goes between data items, but after the decimal point. For example, a sample line of input might look like: 123.45#167.#1895.3#17.

F. Interpreting Results

As each projection is made, in either mode, the name of the projection type, followed by a number in parenthesis and the RSQ of that projection type, is printed. The number, mentioned above, that follows the projection name corresponds to the number given as the projection type at the bottom of the best three projections. (See p. 10 for example.) When you stop, in either mode, the best 3 projections will be printed along with their projection types and RSQ.

G. Running Another Data Set

After the best three regressions have been printed for a given data set you can choose to run another set of data. If you do, you will be asked if you wish to change any projection values. If you answer NO, the program will return to the point that asks you to input the data. If, for example, the first data set was for the years 1960 to 1970 and the next data set is for the years 1955 to 1970, then you would answer the question with YES. The program will then ask you to enter the number of earliest change you wish to make, according to the following key:

1 = Instructions
2 = Mode
3 = RSQ cutoffs
4 = See each regression as run
5 = # of projected years
6 = Years of base data
7 = Print a copy of this key

You should input the number of the earliest change you wish to make as the program will return to that point. Thus, you will be asked the question relating to that change and all that follow again. (See p. 13 for an example.)

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

```
fmretr autoreg exec (u vmkn9
R;

autoreg

********************* AUTOREG PROGRAM *********************


DO YOU WANT A COMPLETE SET OF INSTRUCTIONS? (YES/NO)
no
DO YOU WANT AUTOMATIC OR MANUAL MODE? (AUTO/MAN)
auto
DO WISH TO SET RSQ CUTOFFS? (YES/NO)
yes
LINEAR CUTOFF=?
85
LOG CUTOFF=?
75
DO YOU WISH TO SEE EVERY REGRESSION AS IT IS RUN? (YES/NO)
no
NUMBER OF YEARS TO MAKE PROJECTIONS FOR? (IF #YEARS <10, WRITE AS 09,ETC.)
10
DO YOU WANT TO INPUT INDIVIDUAL DATA YEARS OR A RANGE? (INDIV/RANGE)
range
INPUT YEAR RANGE AS:  XXXX TO XXXX
1970 to 1974
INPUT THE DATA:
12.#13.#14.#12.#15.

PROJECTIONS AUTOMATICALLY MADE:

*****LINEAR(99)***** RSQ= 0.3676
**LOGARITHMIC(98)*** RSQ= 0.3546
*0.7 EXPONENTIAL( 7) RSQ= 0.3647
*0.8 EXPONENTIAL( 8) RSQ= 0.3657
*1.2 EXPONENTIAL(12) RSQ= 0.3695
*1.3 EXPONENTIAL(13) RSQ= 0.3704
*1.4 EXPONENTIAL(14) RSQ= 0.3713

TYPE '1'.  THEN POSITION THE PAPER TO THE TOP OF THE NEXT PAGE AND HIT RETURN.
```

YEAR	BEST THREE PROJECTIONS IN ORDER BY RSQ		
1970	***BASE**>	12.00	<**BASE***
1971	***DATA**>	13.00	<**DATA***
1972	**>	14.00	<**
1973	**>	12.00	<**
1974	**>	15.00	<**
1975	14.92	14.86	14.81
1976	15.61	15.50	15.40
1977	16.35	16.17	16.01
1978	17.12	16.87	16.63
1979	17.93	17.59	17.27
1980	18.77	18.33	17.92
1981	19.64	19.09	18.58
1982	20.54	19.87	19.26
1983	21.47	20.66	19.94
1984	22.43	21.48	20.63
PROJECTION TYPE=	14.00	13.00	12.00
RSQ=	0.3713	0.3704	0.3695
ALPHA=	12.04	11.97	11.90
BETA=	0.23	0.28	0.34
XBAR=	4.96	4.36	3.84
YBAR=	13.20	13.20	13.20

DO YOU WISH TO REGRESS ANOTHER DATA SET? (YES/NO)
yes
DO YOU WISH TO RESET ANY PROJECTION VALUES OTHER THAN DATA?)
yes
INPUT THE NUMBER OF THE EARLIEST CHANGE. (7 PRINTS A KEY.)
7

```
*****KEY TO CHANGES*****
1 = INSTRUCTIONS
2 = MODE
3 = RSQ CUTOFFS
4 = SEE EACH REGRESSION
5 = # OF PROJECTED YEARS
6 = YEARS OF BASE DATA
INPUT THE NUMBER OF THE EARLIEST CHANGE. (7 PRINTS A KEY.)
2
DO YOU WANT AUTOMATIC OR MANUAL MODE? (AUTO/MAN)
man
DO YOU WISH TO SEE EVERY REGRESSION AS IT IS RUN? (YES/NO)
no
NUMBER OF YEARS TO MAKE PROJECTIONS FOR? (IF #YEARS <10, WRITE AS 09,ETC.)
15
DO YOU WANT TO INPUT INDIVIDUAL DATA YEARS OR A RANGE? (INDIV/RANGE)
indiv
NUMBER OF BASE YEARS OF DATA? (IF # OF YEARS<10, WRITE AS  07 ETC.)
05
INPUT THE YEARS:
1900#1910#1920#1930#1940
INPUT THE DATA:
13.#140.#160.#120.#130.

BEGIN MANUAL REGRESSION SECTION.

REG. TYPE?
lin
*****LINEAR(99)***** RSQ= 0.3450

REG. TYPE?
help
INVALID REG. TYPE--TRY AGAIN.


***** KEY TO REGRESSION TYPES *****

        LIN = LINEAR
        LOG = LOGARITHMIC
        EXP = EXPONENTIAL
```

```
jkl
INVALID REG. TYPE--TRY AGAIN.


***** KEY TO REGRESSION TYPES *****

      LIN = LINEAR
      LOG = LOGARITHMIC
      EXP = EXPONENTIAL
      STP = STOP REGRESSION


REG. TYPE?
log
**LOGARITHMIC(98)*** RSQ= 0.7619

REG. TYPE?
exp
SPECIFY EXPONENT: SUCH AS 0.9
0.3
*0.3 EXPONENTIAL( 3) RSQ= 0.6406

REG. TYPE?
exp
SPECIFY EXPONENT: SUCH AS 0.9
0.5
*0.5 EXPONENTIAL( 5) RSQ= 0.5491

REG. TYPE?
exp
SPECIFY EXPONENT: SUCH AS 0.9
1.2
*1.2 EXPONENTIAL(12) RSQ= 0.2845

REG. TYPE?
exp
SPECIFY EXPONENT: SUCH AS 0.9
```

```
0.2
*0.2 EXPONENTIAL( 2) RSQ= 0.6842

REG. TYPE?
stp

TYPE '1'.  THEN POSITION THE PAPER TO THE TOP OF THE NEXT PAGE AND HIT RETURN.
1
```

YEAR	BEST THREE PROJECTIONS IN ORDER BY RSQ		
1900	***BASE**>	13.00	<**BASE***
1910	***DATA**>	140.00	<**DATA***
1920	**>	160.00	<**
1930	**>	120.00	<**
1940	**>	130.00	<**
1941	153.72	156.69	157.82
1942	154.51	157.78	159.07
1943	155.28	158.85	160.29
1944	156.04	159.90	161.50
1945	156.78	160.93	162.68
1946	157.51	161.94	163.85
1947	158.22	162.94	165.00
1948	158.91	163.92	166.14
1949	159.59	164.88	167.26
1950	160.26	165.83	168.36
1951	160.92	166.76	169.45
1952	161.56	167.68	170.52
1953	162.19	168.59	171.58
1954	162.81	169.48	172.63
1955	163.41	170.36	173.66
PROJECTION TYPE=	98.00	2.00	3.00
RSQ=	0.7619	0.7842	0.6406
ALPHA=	27.72	-74.21	-17.85
BETA=	33.71	109.34	57.24
XBAR=	2.52	1.71	2.28
YBAR=	112.60	112.60	112.60

DO YOU WISH TO REGRESS ANOTHER DATA SET? (YES/NO)

```
yes
DO YOU WISH TO RESET ANY PROJECTION VALUES OTHER THAN DATA?)
yes
INPUT THE NUMBER OF THE EARLIEST CHANGE. (7 PRINTS A KEY.)
2
DO YOU WANT AUTOMATIC OR MANUAL MODE? (AUTO/MAN)
help
DO YOU WANT AUTOMATIC OR MANUAL MODE? (AUTO/MAN)
auto
DO WISH TO SET RSQ CUTOFFS? (YES/NO)
error
DO WISH TO SAT RSQ CUTOFFS? (YES/NO)
no
DO YOU WISH TO SEE EVERY REGRESSION AS IT IS RUN? (YES/NO)
error
DO YOU WISH TO SEE EVERY REGRESSION AS IT IS RUN? (YES/NO)
no
NUMBER OF YEARS TO MAKE PROJECTIONS FOR? (IF #YEARS <10, WRITE AS 09,ETC.)
05
DO YOU WANT TO INPUT INDIVIDUAL DATA YEARS OR A RANGE? (INDIV/RANGE)
error
DO YOU WANT TO INPUT INDIVIDUAL DATA YEARS OR A RANGE? (INDIV/RANGE)
range
INPUT YEAR RANGE AS:  XXXX TO XXXX
1900 to 2000@5
RANGE EXCEEDS MAXIMUM OF 100--PLEASE INPUT SMALLER RANGE.
INPUT YEAR RANGE AS:  XXXX TO XXXX
1900 to 1800
STARTING YEAR OF RANGE WAS GREATER THAN LAST YEAR--TRY AGAIN.
INPUT YEAR RANGE AS:  XXXX TO XXXX
1970 to 1974
INPUT THE DATA:
12.#13.#14.#15.

NUMBER OF DATA ITEMS NOT EQUAL TO NUMBER OF BASE YEARS SPECIFIED!
DO YOU WISH TO CHANGE THE NUMBER OF BASE YEARS? (YES/NO)
```

```
no
INPUT THE DATA:
12.#14.#15.#16.#13.

PROJECTIONS AUTOMATICALLY MADE:

*****LINEAR(99)***** RSQ= 0.1600
**LOGARITHMIC(98)*** RSQ= 0.3167
*0.7 EXPONENTIAL( 7) RSQ= 0.2046
*0.6 EXPONENTIAL( 6) RSQ= 0.2203
*0.5 EXPONENTIAL( 5) RSQ= 0.2362
*0.4 EXPONENTIAL( 4) RSQ= 0.2524

TYPE '1'.  THEN POSITION THE PAPER TO THE TOP OF THE NEXT PAGE AND HIT RETURN.
1
```

YEAR	BEST THREE PROJECTIONS IN ORDER BY RSQ		
1970	***BASE**>	12.00	<**BASE***
1971	***DATA**>	14.00	<**DATA***
1972	**>	15.00	<**
1973	**>	10.00	<**
1974	**>	13.00	<**
1975	15.17	15.22	15.22
1976	15.38	15.51	15.53
1977	15.57	15.77	15.82
1978	15.74	15.02	16.09
1979	15.88	16.25	16.35
PROJECTION TYPE=	98.00	4.00	5.00
RSQ=	0.3167	0.2524	0.2362
ALPHA=	12.66	10.64	11.35
BETA=	1.40	2.23	1.58
XBAR=	0.96	1.50	1.68
YBAR=	14.00	14.00	14.00

DO YOU WISH TO REGRESS ANOTHER DATA SET? (YES/NO)
no
CONNECT= 00:24:24 TOTCPU= 000:00.81
CONNECT COST $.97 CPU COST $1.50 I/O COST $.03
TOTAL COST $2.59 ACCOUNT BALANCE $465.03
LOGOFF AT 14:28:07 CST THURSDAY 02/10/77

```
/*ROUTE  PUNCH VMKN9
/*ROUTE  PRINT CS
// EXEC FORTHC,PARM.FORT='OPT=2,DECK'
//SYSIN DD *
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER XYEAR, PRTDAT,PRONAM,XSYEAR
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ,LINCUT,LOGCUT
C  N=NUMBER OF BASE YEAR'S DATA.  M=TOTAL NUMBER OF YEARS' DATA.
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRONAM,LINRSQ,LOGRSQ
      INTEGER*4 YES/'YES'/,NO/'NO'/,MANUAL/'MAN '/,AUTO/'AUTO'/
      INTEGER*4 RANGE/'RANG'/,INDIV/'INDI'/
      INTEGER*4 LIN/'LIN'/,LOG/'LOG'/,EXP/'EXP'/,STOP/'STP'/
C*****************************************************************
C                                                                *
C   THIS NEXT SECTION INITIALIZES THE MATRICES DOPMAT AND BESPRO WITH  *
C   ZEROS, SO THAT THEY ARE DEFINED WHEN THEY ARE REFERENCED.          *
C                                                                *
C*****************************************************************
      DO 17 I=1,6
      DO 17 J=1,3
   17 DOPMAT(I,J)=0.0000
      DO 18 I=1,100
      DO 18 J=1,3
   18 BESPRO(I,J)=0.0000
      WRITE(10,201)
  201 FORMAT(' '/'********************* AUTOREG PROGRAM *********************
     &***'/' '/' ')
  205 WRITE(10,202)
  202 FORMAT('DO YOU WANT A COMPLETE SET OF INSTRUCTIONS? (YES/NO)')
      READ(9,203,END=310) INSTRC
  203 FORMAT(A3)
      IF(INSTRC.EQ.YES) CALL INSTR (&210)
      IF(INSTRC.NE.NO) GO TO 205
  206 FORMAT(' '/' '/'***** KEY TO REGRESSION TYPES *****'/' '/7X,'LIN =
     & LINEAR'/7X,'LOG = LOGARITHMIC'/7X,'EXP = EXPONENTIAL'/7X,'STP = S
     &TOP REGRESSION'/' ')
  210 WRITE(10,207)
  207 FORMAT('DO YOU WANT AUTOMATIC OR MANUAL MODE? (AUTO/MAN)')
      READ(9,208,END=312) IMODE
  208 FORMAT(A4)
      IF(IMODE.EQ.MANUAL) GO TO 209
      IF(IMODE.NE.AUTO) GO TO 210
      IMAN=0
  213 WRITE(10,211)
  211 FORMAT('DO WISH TO SET RSQ CUTOFFS? (YES/NO)')
      READ(9,203,END=313) ICUT
      IF(ICUT.EQ.YES) GO TO 212
      IF(ICUT.NE.NO) GO TO 213
      LINCUT=.90
      LOGCUT=.80
      GO TO 282
  209 IMAN=1
  282 WRITE(10,280)
  280 FORMAT('DO YOU WISH TO SEE EVERY REGRESSION AS IT IS RUN? (YES/NO)
     &')
      READ(9,203,END=314) ISEE
      IF(ISEE.EQ.YES) GO TO 400
      IF(ISEE.NE.NO) GO TO 282
```

```
      PRTDAT=0
      GO TO 216
  400 PRTDAT=1
  216 WRITE(10,222)
  222 FORMAT('NUMBER OF YEARS TO MAKE PROJECTIONS FOR? (IF #YEARS <10, W
     &RITE AS 09,ETC.)')
      READ(9,307,END=315) K
      IF(K.LT.101) GO TO 217
      WRITE(10,224)
  224 FORMAT('NUMBER OF YEARS IS GREATER THAN MAXIMUM OF 100.  TRY AGAIN
     &.')
      GO TO 216
  217 WRITE(10,214)
  214 FORMAT('DO YOU WANT TO INPUT INDIVIDUAL DATA YEARS OR A RANGE? (IN
     &DIV/RANGE)')
      READ(9,215,END=316) INTYPE
  215 FORMAT(A4)
      IF(INTYPE.EQ.RANGE) GO TO 223
      IF(INTYPE.NE.INDIV) GO TO 217
  238 WRITE(10,234)
  234 FORMAT('NUMBER OF BASE YEARS OF DATA? (IF # OF YEARS<10, WRITE AS
     & 07 ETC.)')
      READ(9,307,END=317) N
  307 FORMAT(I2)
      IF(N.GT.100) GO TO 241
      WRITE(10,235)
  235 FORMAT('INPUT THE YEARS:')
      DO 304 I=1,N
  304 READ(9,305,END=236) XYEAR(I)
  305 FORMAT(I4)
      CALL MINVAL
      M=N+K
      GO TO 230
  241 WRITE(10,242)
  242 FORMAT('NUMBER OF YEARS EXCEEDS MAXIMUM OF 100--PLEASE INPUT A SMA
     &LLER NUMBER.')
      GO TO 238
  236 REWIND 9
      WRITE(10,237)
  237 FORMAT('NUMBER OF YEARS READ NOT EQUAL TO NUMBER OF BASE YEARS YOU
     & SPECIFIED--TRY AGAIN.')
      GO TO 238
  212 WRITE(10,219)
  219 FORMAT('LINEAR CUTOFF=?')
      READ(9,220,END=318) LINCUT
  220 FORMAT(F2.2)
      WRITE(10,221)
  221 FORMAT('LOG CUTOFF=?')
      READ(9,220,END=318) LOGCUT
      GO TO 282
  223 WRITE(10,225)
  225 FORMAT('INPUT YEAR RANGE AS:  XXXX TO XXXX')
      READ(9,226,END=319) IMIN,IMAX
  226 FORMAT(I4,4X,I4)
      IF(IMIN.GT.IMAX) GO TO 227
      IRANGE=(IMAX-IMIN)+1
      IF(IRANGE.GT.100) GO TO 228
      DO 229 I=1,IRANGE
      XYEAR(I)=(IMIN-1)+I
  229 XBASE(I)=I
```

```
      N=IRANGE
      M=N+K
      GO TO 230
  227 WRITE(10,231)
  231 FORMAT('STARTING YEAR OF RANGE WAS GREATER THAN LAST YEAR--TRY AGA
     &IN.')
      GO TO 223
  228 WRITE(10,232)
  232 FORMAT('RANGE EXCEEDS MAXIMUM OF 100--PLEASE INPUT SMALLER RANGE.'
     &)
      GO TO 223
  230 WRITE(10,240)
  240 FORMAT('INPUT THE DATA:')
      DO 300 I=1,N
  300 READ(9,301,END=239) YBASE(I)
  301 FORMAT(F9.2)
      IF(IMAN.NE.1) GO TO 243
      WRITE(10,244)
  244 FORMAT(' '/'BEGIN MANUAL REGRESSION SECTION.')
  246 WRITE(10,245)
  245 FORMAT(' '/'REG. TYPE?')
      READ(9,247,END=320) IREG
  247 FORMAT(A3)
      IF(IREG.EQ.LIN) GO TO 248
      IF(IREG.EQ.LOG) GO TO 249
      IF(IREG.EQ.EXP) GO TO 250
      IF(IREG.EQ.STOP) GO TO 251
      WRITE(10,252)
  252 FORMAT('INVALID REG. TYPE--TRY AGAIN.')
      WRITE(10,206)
      GO TO 246
  243 IREGTP=1
      WRITE(10,702)
  702 FORMAT(' '/'PROJECTIONS AUTOMATICALLY MADE:'/' ')
      PRONAM=99
      CALL REGRES
      IF(LINRSQ.GE.LINCUT) GO TO 251
      PRONAM=98
      IREGTP=2
      CALL REGRES
      IF(LOGRSQ.GE.LOGCUT) GO TO 251
      IREGTP=3
      CALL EXPFLW
      GO TO 251
  248 IREGTP=1
      PRONAM=99
      CALL REGRES
      GO TO 246
  249 IREGTP=2
      PRONAM=98
      CALL REGRES
      GO TO 246
  239 REWIND 9
      WRITE(10,253)
  253 FORMAT('NUMBER OF DATA ITEMS NOT EQUAL TO NUMBER OF BASE YEARS SPE
     &CIFIED ')
  390 WRITE(10,391)
  391 FORMAT('DO YOU WISH TO CHANGE THE NUMBER OF BASE YEARS? (YES/NO)')
      READ(9,247,END=321) IDECS
      IF(IDECS.EQ.NO) GO TO 230
```

```
      IF(IMAN.EQ.1) GO TO 238
      GO TO 225
  251 WRITE(10,700)
  700 FORMAT(' '/'TYPE ''1''.  THEN POSITION THE PAPER TO THE TOP OF THE
     & NEXT PAGE AND HIT RETURN.')
      READ(9,701,END=901) IM
  701 FORMAT(I1)
      WRITE(10,100)
  100 FORMAT(' '/11X,'YEAR',13X,'BEST THREE PROJECTIONS IN ORDER BY RSQ'
     1)
      WRITE(10,50) XYEAR(1),YBASE(1)
   50 FORMAT(' '/11X,I4,13X,'***BASE**>',2X,F10.2,5X,'<**BASE***')
      WRITE(10,51) XYEAR(2),YBASE(2)
   51 FORMAT(11X,I4,18X,'***DATA**>',2X,F10.2,5X,'<**DATA***')
      WRITE(10,52) (XYEAR(I),YBASE(I),I=3,N)
   52 FORMAT(11X,I4,25X,'**>',2X,F10.2,5X,'<**')
      WRITE(10,53)
   53 FORMAT(' ')
      DO 199 I=1,K
      XSYEAR=XYEAR(N)+1
  199 WRITE(10,101) XSYEAR,(BESPRO(I,J),J=1,3)
  101 FORMAT(11X,I4,3(10X,F10.2))
      WRITE(10,102) (DOPMAT(1,J),J=1,3)
  102 FORMAT (' '/'PROJECTION TYPE=',9X,F10.2,2(10X,F10.2))
      WRITE(10,103) (DOPMAT(2,J),J=1,3)
  103 FORMAT(1X,'RSQ=',10X,3(10X,F10.4))
      WRITE(10,104) (DOPMAT(3,J),J=1,3)
  104 FORMAT(1X,'ALPHA=',8X,3(10X,F10.2))
      WRITE(10,105) (DOPMAT(4,J),J=1,3)
  105 FORMAT(1X,'BETA=',9X,3(10X,F10.2))
      WRITE(10,106) (DOPMAT(5,J),J=1,3)
  106 FORMAT(1X,'XBAR=',9X,3(10X,F10.2))
      WRITE(10,107) (DOPMAT(6,J),J=1,3)
  107 FORMAT(1X,'YBAR=',9X,3(10X,F10.2))
  392 WRITE(10,254)
  254 FORMAT(' '/' '/' '/' '/'DO YOU WISH TO REGRESS ANOTHER DATA SET? (
     &YES/NO)')
      READ(9,247,END=322) IDECS
      IF(IDECS.EQ.NO) GO TO 999
      IF(IDECS.EQ.YES) GO TO 650
      GO TO 251
  250 IREGTP=3
  393 WRITE(10,255)
  255 FORMAT('SPECIFY EXPONENT: SUCH AS 0.9')
      READ(9,306,END=323) EXPON
  306 FORMAT(F3.1)
      CALL REGRES
      GO TO 246
C
C   SET RSQ OF DATA IN BESPRO TO ZERO SO THAT PROSTO WILL OVERWRITE THE
C   OLD DATA WITH THE NEW.
C
  650 DOPMAT(2,3)=0.0
      DOPMAT(2,2)=0.0
      DOPMAT(2,1)=0.0
  652 WRITE(10,651)
  651 FORMAT('DO YOU WISH TO RESET ANY PROJECTION VALUES OTHER THAN DATA
     &?')
      READ(9,247,END=324)IDECS
      IF(IDECS.EQ.NO) GO TO 230
```

```
      IF(IDECS.NE.YES) GO TO 652
  656 WRITE(10,653)
  653 FORMAT('INPUT THE NUMBER OF THE EARLIEST CHANGE. (7 PRINTS A KEY.)
     &')
      READ(9,655,END=325) IMB
  655 FORMAT(I1)
      GO TO (205,210,213,282,216,217,657),IMB
  657 WRITE(10,654)
  654 FORMAT(' '/'*****KEY TO CHANGES*****'/'1 = INSTRUCTIONS'/'2 = MODE
     &'/'3 = RSQ CUTOFFS'/'4 = SEE EACH REGRESSION'/'5 = # OF PROJECTED
     &YEARS'/'6 = YEARS OF BASE DATA')
      GO TO 656
  901 REWIND 9
      WRITE(10,902)
  902 FORMAT('***ERROR*** YOU PUT A #SIGN AFTER THE LAST DATA ITEM--TRY
     &POSITIONING THE PAPER AGAIN ')
      GO TO 251
C
C   THE REWIND 9'S SET THE WRITE BACK TO FILE FT09F001.
C
  310 REWIND 9
      WRITE(10,311)
  311 FORMAT('***ERROR*** NO ANSWER ENTERED ')
      GO TO 205
  312 REWIND 9
      WRITE(10,311)
      GO TO 210
  313 REWIND 9
      WRITE(10,311)
      GO TO 213
  314 REWIND 9
      WRITE(10,311)
      GO TO 282
  315 REWIND 9
      WRITE(10,311)
      GO TO 216
  316 REWIND 9
      WRITE(10,311)
      GO TO 217
  317 REWIND 9
      WRITE(10,311)
      GO TO 238
  318 REWIND 9
      WRITE(10,311)
      GO TO 212
  319 REWIND 9
      WRITE(10,311)
      GO TO 213
  320 REWIND 9
      WRITE(10,311)
      GO TO 246
  321 REWIND 9
      WRITE(10,311)
      GO TO 390
  322 REWIND 9
      WRITE(10,311)
      GO TO 392
  323 REWIND 9
      WRITE(10,311)
      GO TO 393
```

```
  324 REWIND 9
      WRITE(10,311)
      GO TO 652
  325 REWIND 9
      WRITE(10,311)
      GO TO 656
  999 STOP
      END
C*****************************************************************************
C                                                                             *
C   THIS SUBROUTINE REPLACES THE YEAR INPUT X'S WITH SMALLER X'S IN           *
C   THE VICINITY OF 1.  THIS IS SO THAT THE EXPONENTIAL AND LOG-              *
C   ARITHMIC PROJECTIONS WILL WORK CORRECTLY.  THE SMALLER VALUES ARE         *
C   OBTAINED BY TAKING THE SMALLEST INPUT X AND SUBTRACTING ITS VALUE         *
C   MINUS ONE FROM ALL INPUT X'S.  THUS, WITH THIS TYPE OF SCALING            *
C   THE INPUT X'S DON'T HAVE TO BE SEQUENTIAL YEARS.                          *
C                                                                             *
C*****************************************************************************
      SUBROUTINE MINVAL
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ
      INTEGER XYEAR,PRGNAM,PRTDAT
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRGNAM,LINRSQ,LOGRSQ
      MINXYR=XYEAR(1)
      DO 6 I=1,N
      IF(XYEAR(I).GT.MINXYR) GO TO 6
      MINXYR=XYEAR(I)
    6 CONTINUE
      IREDUC=MINXYR-1
      DO 7 I=1,N
    7 XBASE(I)=XYEAR(I)-IREDUC
      RETURN
      END
      SUBROUTINE INSTR (*)
C*****************************************************************************
C                                                                             *
C   THIS SUBROUTINE WRITES A SHORT SET OF INSTRUCTIONS TO THE USER.           *
C                                                                             *
C*****************************************************************************
      WRITE(10,1)
      WRITE(10,2)
      WRITE(10,3)
      WRITE(10,4)
      WRITE(10,5)
      WRITE(10,6)
      WRITE(10,7)
      WRITE(10,8)
      WRITE(10,9)
      WRITE(10,10)
      WRITE(10,11)
      WRITE(10,12)
      WRITE(10,13)
      WRITE(10,14)
      WRITE(10,15)
      WRITE(10,16)
      WRITE(10,17)
      WRITE(10,41)
      WRITE(10,42)
```

```
      WRITE(10,18)
      WRITE(10,19)
      WRITE(10,20)
      WRITE(10,21)
      WRITE(10,22)
      WRITE(10,23)
      WRITE(10,24)
      WRITE(10,25)
      WRITE(10,26)
      WRITE(10,27)
      WRITE(10,28)
      WRITE(10,29)
      WRITE(10,30)
      WRITE(10,31)
      WRITE(10,32)
      WRITE(10,33)
      WRITE(10,50)
      WRITE(10,51)
      WRITE(10,53)
      WRITE(10,54)
      WRITE(10,34)
      WRITE(10,35)
      WRITE(10,43)
      WRITE(10,44)
      WRITE(10,37)
      WRITE(10,38)
      WRITE(10,39)
      WRITE(10,40)
      WRITE(10,45)
      WRITE(10,46)
      WRITE(10,60)
      WRITE(10,61)
      WRITE(10,62)
      WRITE(10,63)
      WRITE(10,64)
      WRITE(10,65)
    1 FORMAT('********************************* AUTOREG REGRESSION ********
     &************************')
    2 FORMAT(' '/,' '/,'THIS PROGRAM ALLOWS YOU TO RUN LINEAR,LOG,AND EX
     &PONENTIAL REGRESSIONS ON DATA YOU')
    3 FORMAT('INPUT AT A TERMINAL.  IN RUNNING AUTOREG, YOU ARE ALLOWED
     &TO MAKE THE FOLLOWING')
    4 FORMAT('DECISIONS:  (AFTER YOU TYPE IN THE ANSWER TO A DECISION, A
     &LWAYS HIT RETURN.)')
    5 FORMAT(' '/'     1) SELECT AUTOMATIC OR MANUAL MODE.  IF YOU SELEC
     &T MANUAL, YOU CONTROL WHICH')
    6 FORMAT('REGRESSION TYPES ARE PERFORMED.  (SEE 5 BELOW FOR THE TYPE
     &S.)  IF YOU SELECT AUTO')
    7 FORMAT('-MATIC,THE COMPUTER WILL SELECT THE PROJECTIONS TO BE RUN
     &ACCORDING TO THE FOL-')
    8 FORMAT('LOWING SCHEME:')
    9 FORMAT('     1ST-LINEAR')
   10 FORMAT('     2ND-LOGARITHMIC, IF LINEAR RSQ (LINRSQ) IS LESS THAN
     &.90')
   11 FORMAT('     3RD-EXPONENTIAL, IF LOG RSQ (LOGRSQ) IS LESS THAN .80
     &')
   12 FORMAT('         EXPONENTIAL REGRESSION IS PERFORMED USING A BUILT
     &-IN DECISION TREE UNTIL')
   13 FORMAT('         ALL EXPONENTS BETWEEN 0.4 AND 1.4 ARE TRIED IMPLI
     &CITLY.')
```

```
14 FORMAT(' '/'     2) IF YOU SELECTED AUTOMATIC, DECIDE IF YOU WISH
  & TO OVERRIDE THE RSQ CUTOFFS')
15 FORMAT('DISCUSSED ABOVE.  IF YOU DO, RESPOND TO LINRSQ= WITH THE C
  &UTOFF FOR TRYING A LOG')
16 FORMAT('PROJECTION.  (WRITE AS A WHOLE NUMBER--EX: FOR .90 RSQ, IN
  &PUT 90.)  RESPOND TO ')
17 FORMAT('LOGRSQ= WITH THE CUTOFF FOR TRYING EXPONENTIAL PROJECTIONS
  &.')
41 FORMAT(' '/'     3) DECIDE IF YOU WISH EACH PROJECTION MADE TO BE
  &PRINTED AS THE PROJECTION')
42 FORMAT('IS MADE.')
18 FORMAT(' '/'     4) DECIDE WHETHER TO INPUT THE YEARS OF THE DATA
  &INDIVIDUALLY AS 1963,1964,')
19 FORMAT('ETC. OR AS A RANGE IN THE FORM:  1963 TO 1975.  YEARS MUST
  & BE INPUT INDIVIDUALLY')
20 FORMAT('IF A YEAR''S DATA IS MISSING.  (OMIT THE YEAR NUMBER FOR T
  &HOSE YEARS THAT DATA IS')
21 FORMAT('NOT AVAILABLE FOR.)')
22 FORMAT(' '/'     5) SELECT THE NUMBER OF YEARS OF BASE DATA YOU AR
  &E GOING TO INPUT (MAX=100)')
23 FORMAT('AND SELECT THE NUMBER OF YEARS INTO THE FUTURE THAT YOU WI
  &SH PROJECTIONS TO BE')
24 FORMAT('MADE FOR (MAX=100).  (WRITE NUMBERS BELOW 10 AS EX: 09.)')
25 FORMAT(' '/'     6) IF YOU SELECTED MANUAL MODE, EVERY TIME THE PR
  &OMPT--REG. TYPE?--APPEARS')
26 FORMAT('INPUT ONE OF THE FOLLOWING PROJECTION REQUESTS:')
27 FORMAT('     LIN = LINEAR PROJECTION OF THE FORM Y=A+BX')
28 FORMAT('     LOG = LOGARITHMIC PROJECTION OF THE FORM Y=A+B*LOGX')
29 FORMAT('     EXP = EXPONENTIAL PROJECTION OF THE FORM Y=A+B*(X TO
  &THE C POWER)')
30 FORMAT('           (YOU WILL BE ASKED TO INPUT THE EXPONENT (C))')
31 FORMAT('     STP = TERMINATE PROJECTIONS ON PRESENT SET OF DATA')
32 FORMAT(' '/'     7) YOU WILL BE ASKED TO INPUT THE YEARS AND DATA.
  &  INPUT THEM WITH A')
33 FORMAT('# SIGN BETWEEN EACH ITEM.  AT THE END OF A LINE OF INPUT,
  &  HIT RETURN AND')                                                  !
34 FORMAT('YOU WILL BE ALLOWED TO INPUT THE REST OF THE DATA ON THE N
  &EXT LINE OR, IF ALL THE')
35 FORMAT('DATA HAS BEEN INPUT, THE RETURN WILL SIGNAL THE PROGRAM TO
  & GO TO THE NEXT STEP.')
50 FORMAT('INPUT ALL DATA EITHER WITH A DECIMAL IN OR BEHIND THE NUMB
  &ER.  FOR EXAMPLE')
51 FORMAT('A SAMPLE LINE OF INPUT MIGHT LOOK LIKE: 123.#123.45#123.')
53 FORMAT('DON''T PUT A #SIGN AFTER THE LAST DATA ITEM.  BE SURE TO I
  &NCLUDE A DECIMAL')
54 FORMAT('POINT FOR EACH DATA ITEM ')
37 FORMAT('PRINTED.  WHEN YOU STOP, THE BEST 3 PROJECTIONS WILL BE PR
  &INTED ALONG WITH THEIR')
38 FORMAT('PROJECTION TYPES AND RSQ.  IN EITHER OF THE TWO MODES, THE
  & TYPES OF PROJECTIONS')
39 FORMAT('MADE WILL BE PRINTED, ALONG WITH THE BEST 3 PROJECTIONS, T
  &HEIR RSQ''S AND')
43 FORMAT(' '/' '/'AS EACH PROJECTION IS MADE, IN EITHER MODE, THE NA
  &ME OF THE PROJECTION TYPE,')
44 FORMAT('FOLLOWED BY A NUMBER IN PARENTHESIS AND THE RSQ OF THAT PR
  &OJECTION TYPE, IS')
40 FORMAT('PROJECTION TYPES.  THE NUMBER, MENTIONED ABOVE, THAT FOLLO
  &WS THE PROJECTION')
45 FORMAT('NAME IS THE KEY TO THE NUMBER GIVEN AS THE PROJECTION TYPE
  & AT THE BOTTOM OF')
```

```
   46 FORMAT('THE BEST THREE PROJECTIONS.')
   60 FORMAT(' '/'AT THE END OF A REGRESSION, IF YOU CHOOSE TO REGRESS A
     &NOTHER SET OF DATA,')
   61 FORMAT('YOU CAN PICK THE VALUES TO RESET, SUCH AS THE MODE--AUTO O
     &R MANUAL.  ALWAYS')
   62 FORMAT('SELECT THE NUMBER OF THE EARLIEST CHANGE YOU WISH TO MAKE
     &AS YOU WILL CAUSE THE')
   63 FORMAT('PROGRAM TO RETURN TO THE POINT COINCIDING WITH THE PARAME
     &TER WHO'S NUMBER')
   64 FORMAT('YOU SELECTED, AND THUS YOU WILL ALSO GET TO INPUT DECISION
     &S FOR ALL PARAMETERS')
   65 FORMAT('WITH LARGER NUMBERS THAN THE ONE YOU SELECTED.'/' '/' '/)
      RETURN
      END
C*************************************************************************
C                                                                        *
C   THIS SUBROUTINE PROJECTS INPUT DATA ACCORDING TO THE VALUE OF        *
C   IREGTP.  IF:  IREGTP =1  ; THE PROJECTION IS LINEAR                  *
C                        2  ; THE PROJECTION IS LOGARITHMIC              *
C                        3  ; THE PROJECTION IS EXPONENTIAL              *
C                                                                        *
C   THE LOG PROJECTION IS OF THE FORM Y=A+B LOGX.  THE EXPONENTIAL       *
C   PROJECTION IS OF THE FORM Y=A+BX**EXPON; WHERE EXPON IS INPUT TO     *
C   THE SUBROUTINE AND LIES BETWEEN .4 AND 1.4, EXCLUDING 1.0.           *
C                                                                        *
C*************************************************************************
      SUBROUTINE REGPES
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ
      DIMENSION XSPROJ(100)
      INTEGER PRTDAT,XYEAR,XSYEAR,PRONAM
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRONAM,LINRSQ,LOGRSQ
      SUMXY=0.000000000
      SUMSQX=0.000000000
      SUMX=0.000000000
      SUMY=0.000000000
      SUMSQY=0.00000000
C*************************************************************************
C                                                                        *
C   THIS SECTION MANIPULATES THE INPUT X VALUES ACCORDING TO THE TYPE    *
C   OF PROJECTION SPECIFIED BY IREGTP.  THE X VALUES ARE THE TRANS-      *
C   FORMED X VALUES OUTPUT BY SUBROUTINE MINVAL.                         *
C                                                                        *
C*************************************************************************
      GO TO (620,621,622),IREGTP
  622 DO 708 I=1,N
  708 XSBASE(I)=XBASE(I)**EXPON
      PRONAM=(EXPON*10)+.0001
      GO TO 623
  621 DO 700 I=1,N
  700 XSBASE(I)=DLOG(XBASE(I))
      GO TO 623
  620 DO 701 I=1,N
  701 XSBASE(I)=XBASE(I)
C*************************************************************************
C                                                                        *
C   THIS SECTION PROJECTS THE INPUT DATA K YEARS AND ASSIGNS THE         *
C   RESULTS TO OUTPUT VECTOR YSPROJ.                                     *
```

```
C                                                                       *
C***********************************************************************
  623 DO 600 I=1,N
      SUMY=SUMY+YBASE(I)
      SUMX=SUMX+XSBASE(I)
      SUMSQX=SUMSQX+XSBASE(I)**2
      SUMSQY=SUMSQY+YBASE(I)**2
  600 SUMXY=SUMXY+YBASE(I)*XSBASE(I)
      SUMXSQ=SUMX**2
      XBAR=SUMX/N
      YBAR=SUMY/N
      TOP=SUMXY-((SUMX*SUMY)/N)
      BETA=TOP/(SUMSQX-SUMXSQ/N)
      ALPHA=YBAR-(BETA*XBAR)
      GO TO (624,625,626),IREGTP
  624 DO 709 I=1,K
      XSPROJ(I)=XBASE(N)+I
  709 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
      GO TO 640
  626 DO 710 I=1,K
      XSPROJ(I)=(XBASE(N)+I)**EXPON
  710 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
      GO TO 640
  625 DO 601 I=1,K
      XSPROJ(I)=DLOG(XBASE(N)+I)
  601 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
C***********************************************************************
C                                                                       *
C   THIS SECTION COMPUTES THE VALUE OF RSQ AND WRITES THE PROJECTION    *
C   NAME AND RSQ VALUE TO THE TERMINAL.                                 *
C                                                                       *
C***********************************************************************
  640 TOP=(SUMXY/N-XBAR*YBAR)
      BOT=DSQRT((SUMSQX/N-XBAR**2)*(SUMSQY/N-YBAR**2))
      RSQ=(TOP/BOT)**2
      GO TO (501,502,503),IREGTP
  501 WRITE(10,504) PRONAM,RSQ
  504 FORMAT('*****LINEAR(',I2,')***** RSQ= ',F6.4)
      GO TO 599
  502 WRITE(10,505) PRONAM,RSQ
  505 FORMAT('**LOGARITHMIC(',I2,')*** RSQ= ',F6.4)
      GO TO 599
  503 WRITE(10,506) EXPON,PRONAM,RSQ
  506 FORMAT('*',F3.1,' EXPONENTIAL(',I2,') RSQ= ',F6.4)
  599 IF(PRTDAT.NE.1) GO TO 996
C***********************************************************************
C                                                                       *
C   THIS SECTION PRINTS THE INPUT DATA FOR THE BASE YEARS AND THE       *
C   PROJECTED YEARS' DATA FOR EACH PROJECTION MADE IF PRTDAT = 1.       *
C                                                                       *
C***********************************************************************
      WRITE(10,616)
  616 FORMAT(' ')
      WRITE(10,699) (XYEAR(I),YBASE(I),I=1,N)
      WRITE(10,616)
      DO 702 I=1,K
      XSYEAR=XYEAR(N)+I
  702 WRITE (10,699) XSYEAR,YSPROJ(I)
  699 FORMAT (1X,I4,10X,F8.2)
  996 IF(IREGTP.EQ.1) LINRSQ=RSQ
```

```
      CALL PRCSTO
  998 RETURN
      END
C*****************************************************************
C   THIS SUBROUTINE CONTAINS THE LOGIC FOR PICKING THE BEST EXPONENTIAL*
C   PROJECTION.  THIS IS DONE BY PICKING THE NEXT PROJECTION TO TRY    *
C   BASED ON THE RSQ OF THE LAST PROJECTION JUST MADE.  THIS SUB-      *
C   ROUTINE ASSUMES THAT IF THE LINEAR RSQ (LINRSQ) WAS NOT ABOVE 0.7  *
C   THAT THAT THE TREND WOULD COME NEARER TO AN EXPONENTIAL PROJECTION *
C   THAT WAS FURTHER AWAY FROM 1.0 AND THUS THE SUBROUTINE LOOKS FIRST *
C   AT EXPONENTIAL PROJECTIONS WHERE THE EXPONENT IS FURTHER AWAY FROM *
C   1.0.  OTHERWISE (WHEN THE RSQ OF THE LINEAR PROJECTION IS ABOVE    *
C   .70) THE EXPONENTIAL PROJECTIONS WITH AN EXPONENT NEAR 1.0 ARE     *
C   TRIED FIRST IN AN EFFORT TO SAVE THE NUMBER OF PROJECTIONS NEC-    *
C   CESSARY.                                                          *
C*****************************************************************
C                          SYMBOL DEFINITION                        *
C                                                                   *
C   LINRSQ = R SQUARE OF THE LINEAR REGRESSION.                     *
C                                                                   *
C   REGRES = SUBROUTINE WHICH PERFORMS THE FOLLOWING TYPES OF REGRES-  *
C            SION.  IREGTP IS THE KEY TO THE TYPE OF REGRESSION RE-    *
C            REGRESSION REGRES IS TO PERFORM.                          *
C            LINEAR       WHEN IREGTP = 1.0                            *
C            LOGARITHMIC               = 2.0                          *
C            EXPONENTIAL               = 3.0,                         *
C                                                                   *
C   EXPON  = VALUE PASSED IN COMMON TO SUBROUTINE REGRES TO DETER-     *
C            MINE THE EXPONENT USED IN EXPONENTIAL REGRESSION          *
C                                                                   *
C   ERSQX  = RSQ FOR AN EXPONENTIAL PROJECTION USING THE EXPONENT X    *
C            EX:  ERSQ12 MEANS THE EXPONENT 1.2.                       *
C                                                                   *
C*****************************************************************
      SUBROUTINE EXPFLW
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER XYEAR,PRTDAT,PRONAM,XSYEAR
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRONAM,LINRSQ,LOGRSQ
      IF(LINRSQ.LT.0.70) GO TO 500
      EXPON=.900000000
      CALL REGRES
      ERSQ9=RSQ
      IF(ERSQ9.LT.LINRSQ) GO TO 501
      EXPON=.700000000
      CALL REGRES
      ERSQ7=RSQ
      IF(ERSQ7.LT.ERSQ9) GO TO 502
      EXPON=.500000000
      CALL REGRES
      ERSQ5=RSQ
      IF(ERSQ5.LT.ERSQ7) GO TO 503
  505 EXPON=.400000000
      CALL REGRES
      GO TO 888
  503 EXPON=.600000000
      CALL REGRES
      GO TO 888
```

```
510 EXPON=1.100000000
    CALL REGRES
    ERSQ11=RSQ
    IF(ERSQ11.LT.ERSQ9) GO TO 888
    GO TO 530
502 EXPON=.800000000
    CALL REGRES
    ERSQ8=RSQ
    IF(ERSQ8.LT.ERSQ9) GO TO 510
501 EXPON=1.100000000
    CALL REGRES
    ERSQ11=RSQ
    IF(ERSQ11.LT.LINRSQ) GO TO 888
530 EXPON=1.200000000
    CALL REGRES
    ERSQ12=RSQ
    IF(ERSQ12.LT.ERSQ11) GO TO 888
    EXPON=1.300000000
    CALL REGRES
    ERSQ13=RSQ
    IF(ERSQ13.LT.ERSQ12) GO TO 888
531 EXPON=1.400000000
    CALL REGRES
    GO TO 888
500 EXPON=.700000000
    CALL REGRES
    ERSQ7=RSQ
    IF(ERSQ7.LT.LINRSQ) GO TO 516
    EXPON=.600000000
    CALL REGRES
    ERSQ6=RSQ
    IF(ERSQ6.LT.ERSQ7) GO TO 517
    EXPON=.500000000
    CALL REGRES
    ERSQ5=RSQ
    IF(ERSQ5.LT.ERSQ6) GO TO 888
    GO TO 505
517 EXPON=.800000000
    CALL REGRES
    ERSQ8=RSQ
    IF(ERSQ8.LT.ERSQ7) GO TO 888
    EXPON=.900000000
    CALL REGRES
    GO TO 888
516 EXPON=.800000000
    CALL REGRES
    ERSQ8=RSQ
    IF(ERSQ8.LT.LINRSQ) GO TO 522
    EXPON=.900000000
    CALL REGRES
    GO TO 888
522 EXPON=1.200000000
    CALL REGRES
    ERSQ12=RSQ
    IF(ERSQ12.LT.LINRSQ) GO TO 888
    EXPON=1.300000000
    CALL REGRES
    ERSQ13=RSQ
    IF(ERSQ13.GE.ERSQ12) GO TO 531
    EXPON=1.100000000
```

```
      CALL REGRES
  888 RETURN
      END
C*****************************************************************************
C                                                                              *
C   THIS SUBROUTINE KEEPS THE BEST THREE PROJECTIONS AND THEIR                 *
C   ASSOCIATED DOPE VECTORS (CONTAINING RSQ, ALPHA, BETA, YBAR, XBAR,          *
C   AND PRONAM.)  THEY ARE STORED IN MATRIX DOPMAT (FOR THE STATISTICS)        *
C   AND IN MATRIX BESPRO FOR THE BEST THREE PROJECTIONS.  THE                  *
C   PROJECTIONS ARE FILED IN THE MATRIX IN ORDER OF THEIR RSQ, WITH            *
C   THE PROJECTION WITH THE HIGHEST RSQ GOING INTO THE FIRST COLUMN            *
C   OF THE MATRIX AND SO ON.  NEW PROJECTIONS ARE COMPARED TO THOSE            *
C   ON FILE IN THIS SUBROUTINE AND IF THEIR RSQ EXCEEDS THAT OF THE            *
C   LOWEST RSQ-PROJECTION ON FILE, THE NEW PROJECTION IS STORED IN             *
C   THE COLUMN BEFITTING ITS RSQ RANK AND ONE OF THE OTHER PROJEC-             *
C   TIONS IS DROPPED AND THE OTHER PROJECTIONS AND THEIR ASSOCIATED            *
C   DOPE VECTORS ARE MOVED OVER TO MAKE ROOM FOR THE NEW PROJECTION            *
C   AND ITS DOPE VECTOR, WHERE IT, AS DICTATED BY ITS RSQ RANK,                *
C   SHOULD GO.                                                                 *
C   **THUS THE PRINTING OF BESPRO WOULD GIVE THE BEST THREE PROJEC-            *
C   TIONS FOUND FOR A GIVEN SET OF INPUT DATA.                                 *
C                                                                              *
C   PRONAM = NAME OF THE TYPE OF PROJECTION JUST MADE BY REGRES.               *
C            PRONAM IS REPRESENTED BY A NUMBER, THE KEY TO WHICH IS:           *
C                                                                              *
C   -------KEY------------------PRONAM--------------------KEY--------         *
C                                                                              *
C               99 = LINEAR                                                    *
C               98 = LOGARITHMIC                                               *
C               OTHERWISE :  10 =EXPONENT                                      *
C                            EX: 1.4->PRONAM=14                                *
C                                                                              *
C*****************************************************************************
      SUBROUTINE PROSTO
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ
      INTEGER PRTDAT,XYEAR,XSYEAR,PRONAM
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRONAM,LINRSQ,LOGRSQ
      IF(RSQ.LE.DOPMAT(2,3)) GO TO 999
      IF(RSQ.LE.DOPMAT(2,2)) GO TO 1
      IF(RSQ.LE.DOPMAT(2,1)) GO TO 2
      DO 3 I=1,6
      DOPMAT(I,3)=DOPMAT(I,2)
    3 DOPMAT(I,2)=DOPMAT(I,1)
      DOPMAT(1,1)=PRONAM
      DOPMAT(2,1)=RSQ
      DOPMAT(3,1)=ALPHA
      DOPMAT(4,1)=BETA
      DOPMAT(5,1)=XBAR
      DOPMAT(6,1)=YBAR
      DO 4 I=1,K
      BESPRO(I,3)=BESPRO(I,2)
      BESPRO(I,2)=BESPRO(I,1)
    4 BESPRO(I,1)=YSPROJ(I)
      GO TO 999
    2 DO 5 I=1,6
    5 DOPMAT(I,3)=DOPMAT(I,2)
      DOPMAT(1,2)=PRONAM
```

```
      DOPMAT(2,2)=RSQ
      DOPMAT(3,2)=ALPHA
      DOPMAT(4,2)=BETA
      DOPMAT(5,2)=XBAR
      DOPMAT(6,2)=YBAR
      DO 6 I=1,K
      BESPRO(I,3)=BESPRO(I,2)
    6 BESPRO(I,2)=YSPROJ(I)
      GO TO 999
    1 DOPMAT(1,3)=PROGAM
      DOPMAT(2,3)=RSQ
      DOPMAT(3,3)=ALPHA
      DOPMAT(4,3)=BETA
      DOPMAT(5,3)=XBAR
      DOPMAT(6,3)=YBAR
      DO 7 I=1,K
    7 BESPRO(I,3)=YSPROJ(I)
  999 RETURN
      END
```

III. AUTOREG TERM PROGRAMMER'S GUIDE

I. How to Setup AUTOREG for Terminal Use

A. Compile Step

The listing of AUTOREG in the user section shows the JCL cards necessary to produce a punched copy of the object deck which can then be read via CMS to establish AUTOREG on disk so it is accessible via the terminal. When compiling AUTOREG be sure to omit the normal //GO.SYSIN DD * card. The JCL shown below is the JCL mentioned above that should be placed at the beginning of AUTOREG when compiling it.

```
//Jobcard
/* Route Punch VMXXX
// EXEC FORTHC,PARM.FORT='OPT=2,DECK'
//SYSIN DD *
```

The second JCL card above causes the punched copy of the object deck to be routed to VMXXX where XXX are the last three digits of your account number. The account number must have FM space allocated to it in order to store a compiled copy of AUTOREG on disk. (See the accountant at the KSU Computing Center to obtain FM space.)

The third JCL card above causes the AUTOREG program to be compiled in Fortran H with optimization level 2 (which makes execution of the compiled program faster) and causes an object deck of the compiled program to be punched.

B. Setup on CMS Disk

Once a compiled version of AUTOREG has been routine to your CMS file you have 24 hours to transfer the object deck to your permanent CMS disk storage (FM space). Otherwise, the object deck will be purged by the Computing Center. To establish the object deck in your permanent FM space, issue the following CMS commands:

```
Read Fort Text
Load Fort
Genmod Fort
FMS Fort Module (U VMXXX
```

where: XXX are the last three digits of the account number you have FM space for. These CMS commands read the compiled object deck, link-edit it for CMS and store it in your FM space.)

C. Create the Correct CMS Environment

To establish AUTOREG so it runs correctly, (so the user only has to retrieve it and type AUTOREG) you must create a special EXEC procedure. To do this, create a file named AUTOREG EXEC in CMS and put the following statements in it:

```
&CONTROL OF NOMSG
FMRETR FORT MODULE (U VMXXX
FILEDEF 09 TERM (PERM LRECL 80 BLKSIZE 80 RECFM F
FILEDEF 10 TERM (PERM LRECL 120 BLKSIZE 120
FORT
CP LOGOFF
&EXIT
```

The first statement arranges it so that external terminal messsages (such as from another terminal) aren't received while the user is running AUTOREG. The second statement retrieves AUTOREG (called Fort Module) from your FM space - VMXXX. The next two statements establish the record length and blocksize for the read and write buffers and link unit number 10 to the terminal. These two steps are necessary for a Fortran H program to work interactively. The FORT statement tells CMS to start executing AUTOREG. The seventh statement automatically logs the user off the terminal when he finishes with AUTOREG and the eighth statement signals the end of the Autoreg Exec procedure.

The Autoreg Exec procedure should also be stored in your FM space. It is actually the Autoreg Exec procedure that the user retrieves. Autoreg Exec then retrieves AUTOREG (Fort Module) and runs it.

D. Making AUTOREG Accessible to Other Users

To allow users with other account numbers access to AUTOREG, so they can retrieve a copy of it and run it, issue the following CMS commands:

```
FMPUBLIC  AUTOREG EXEC
FMPUBLIC  FORT MODULE
```

II. Main Program

The main program serves three basic purposes in the AUTOREG program: 1) It serves to interact with the user, translating the user requests into calls to various subroutines structured such that what the user requests is done; 2) Since most users are not familiar with computers, all user inputs are tested for errors and whenever possible the user is notified of the error, requested to try again, and given help in determining just what to input, and 3) Data input is done, error checked, and the three best projections are printed with their associated statistics, followed by a section allowing the user to run another data set changing only the control items that differ from those of the previous data set. The function of each section of the main program is relatively clear due to the terminal messages shown in the listing and the flow shown in the flowchart following the listing. Because of this, only those sections or functions that aren't particularly clear will be elaborated on here.

The section that asks the user whether he wants automatic or manual regression works as follows. If manual regression is requested, IMAN is set to 1. If automatic regression is requested, IMAN is set to 0. Later, after all the data is input, if IMAN = 0, EXPFLW is called and controls the calling of REGRES. If IMAN = 1 (manual) then a section of the main interacts with the user and allows him to control the calling of REGRES and select the type of regressions to be performed.

The REWIND 9 statements are used in the case of read errors to set the Fortran H writing routines back to file FT09F001, which is the terminal. This is needed because a read error causes the print file number to be incremented by one.

The section that reads in linear and log cutoffs later uses these values to determine whether to call Expflw and run the exponential regressions or not. If the r^2 from the linear regression exceeds .90 (default) or a user input value, then the running of regressions for that particular data set terminates as a regression with a good enough fit has already been found. The same applies in the case of a log projection except that its cutoff r^2 is preset to .80 unless the user overrides this.

After the MAIN routine has called the subroutines, controlled the regressions and printed the best three projections, it contains another user interactive decision section. This decision section is invoked if the user decides to run other data sets after completing the first one. Rather than make the user answer all the interactive parameter questions again, he can indicate the point in the program where the earliest question relating to a change he would like to make in the previously set parameters occurs. The program then returns to that point and the user must answer that parameter question and all that follow again.

The reads are done on unit 9 and the writes are done on unit 10. (See previous section.) The use of separate device numbers causes Fortran to use separate buffers for reading and writing and thus avoids the opening and closing of buffers each time a read-write combination occurs (which is the whole basis for interaction.) Also, the Fortran H compiler at KSU won't compile the program correctly for terminal use unless separate buffers are used.

Main Program Listing

```
/*ROUTE  PUNCH VMKN9
/*ROUTE  PRINT OS
// EXEC FORTHC,PARM.FORT='OPT=2,DECK'
//SYSIN DD *
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER XYEAR, PRTDAT,PRONAM,XSYEAR
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ,LINCUT,LOGCUT
C  N=NUMBER OF BASE YEAR'S DATA.  M=TOTAL NUMBER OF YEARS' DATA.
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRONAM,LINRSQ,LOGRSQ
      INTEGER*4 YES/'YES'/,NO/'NO'/,MANUAL/'MAN '/,AUTO/'AUTO'/
      INTEGER*4 RANGE/'RANG'/,INDIV/'INDI'/
      INTEGER*4 LIN/'LIN'/,LOG/'LOG'/,EXP/'EXP'/,STOP/'STP'/
C*****************************************************************
C                                                                *
C   THIS NEXT SECTION INITIALIZES THE MATRICES DOPMAT AND BESPRO WITH  *
C   ZEROS, SO THAT THEY ARE DEFINED WHEN THEY ARE REFERENCED.          *
C   .                                                            *
C*****************************************************************
      DO 17 I=1,6
      DO 17 J=1,3
   17 DOPMAT(I,J)=0.0000
      DO 18 I=1,100
      DO 18 J=1,3
   18 BESPRO(I,J)=0.0000
      WRITE(10,201)
  201 FORMAT(' '/'********************* AUTOREG PROGRAM *****************
     &***'/' '/' ')
  205 WRITE(10,202)
  202 FORMAT('DO YOU WANT A COMPLETE SET OF INSTRUCTIONS? (YES/NO)')
      READ(9,203,END=310) INSTRC
  203 FORMAT(A3)
      IF(INSTRC.EQ.YES) CALL INSTR (&210)
      IF(INSTRC.NE.NO) GO TO 205
  206 FORMAT(' '/' '/'***** KEY TO REGRESSION TYPES *****'/' '/7X,'LIN =
     & LINEAR'/7X,'LOG = LOGARITHMIC'/7X,'EXP = EXPONENTIAL'/7X,'STP = S
     &TOP REGRESSION'/' ')
  210 WRITE(10,207)
  207 FORMAT('DO YOU WANT AUTOMATIC OR MANUAL MODE? (AUTO/MAN)')
      READ(9,208,END=312) IMODE
  208 FORMAT(A4)
      IF(IMODE.EQ.MANUAL) GO TO 209
      IF(IMODE.NE.AUTO) GO TO 210
      IMAN=0
  213 WRITE(10,211)
  211 FORMAT('DO WISH TO SET RSQ CUTOFFS? (YES/NO)')
      READ(9,203,END=313) ICUT
      IF(ICUT.EQ.YES) GO TO 212
      IF(ICUT.NE.NO) GO TO 213
      LINCUT=.90
      LOGCUT=.80
      GO TO 282
  209 IMAN=1
  282 WRITE(10,280)
  280 FORMAT('DO YOU WISH TO SEE EVERY REGRESSION AS IT IS RUN? (YES/NO)
     &')
      READ(9,203,END=314) ISEE
      IF(ISEE.EQ.YES) GO TO 400
      IF(ISEE.NE.NO) GO TO 282
```

```
      PRTDAT=0
      GO TO 216
400   PRTDAT=1
216   WRITE(10,222)
222   FORMAT('NUMBER OF YEARS TO MAKE PROJECTIONS FOR? (IF #YEARS <10, W
     &RITE AS 09,ETC.)')
      READ(9,307,END=315) K
      IF(K.LT.101) GO TO 217
      WRITE(10,224)
224   FORMAT('NUMBER OF YEARS IS GREATER THAN MAXIMUM OF 100.  TRY AGAIN
     &.')
      GO TO 216
217   WRITE(10,214)
214   FORMAT('DO YOU WANT TO INPUT INDIVIDUAL DATA YEARS OR A RANGE? (IN
     &DIV/RANGE)')
      READ(9,215,END=316) INTYPE
215   FORMAT(A4)
      IF(INTYPE.EQ.RANGE) GO TO 223
      IF(INTYPE.NE.INDIV) GO TO 217
238   WRITE(10,234)
234   FORMAT('NUMBER OF BASE YEARS OF DATA? (IF # OF YEARS<10, WRITE AS
     & 07 ETC.)')
      READ(9,307,END=317) N
307   FORMAT(I2)
      IF(N.GT.100) GO TO 241
      WRITE(10,235)
235   FORMAT('INPUT THE YEARS:')
      DO 304 I=1,N
304   READ(9,305,END=236) XYEAR(I)
305   FORMAT(I4)
      CALL MINVAL
      M=N+K
      GO TO 230
241   WRITE(10,242)
242   FORMAT('NUMBER OF YEARS EXCEEDS MAXIMUM OF 100--PLEASE INPUT A SMA
     &LLER NUMBER.')
      GO TO 238
236   REWIND 9
      WRITE(10,237)
237   FORMAT('NUMBER OF YEARS READ NOT EQUAL TO NUMBER OF BASE YEARS YOU
     & SPECIFIED--TRY AGAIN.')
      GO TO 238
212   WRITE(10,219)
219   FORMAT('LINEAR CUTOFF=?')
      READ(9,220,END=318) LINCUT
220   FORMAT(F2.2)
      WRITE(10,221)
221   FORMAT('LOG CUTOFF=?')
      READ(9,220,END=318) LOGCUT
      GO TO 282
223   WRITE(10,225)
225   FORMAT('INPUT YEAR RANGE AS:  XXXX TO XXXX')
      READ(9,226,END=319) IMIN,IMAX
226   FORMAT(I4,4X,I4)
      IF(IMIN.GT.IMAX) GO TO 227
      IRANGE=(IMAX-IMIN)+1
      IF(IRANGE.GT.100) GO TO 228
      DO 229 I=1,IRANGE
      XYEAR(I)=(IMIN-1)+I
229   XBASE(I)=I
```

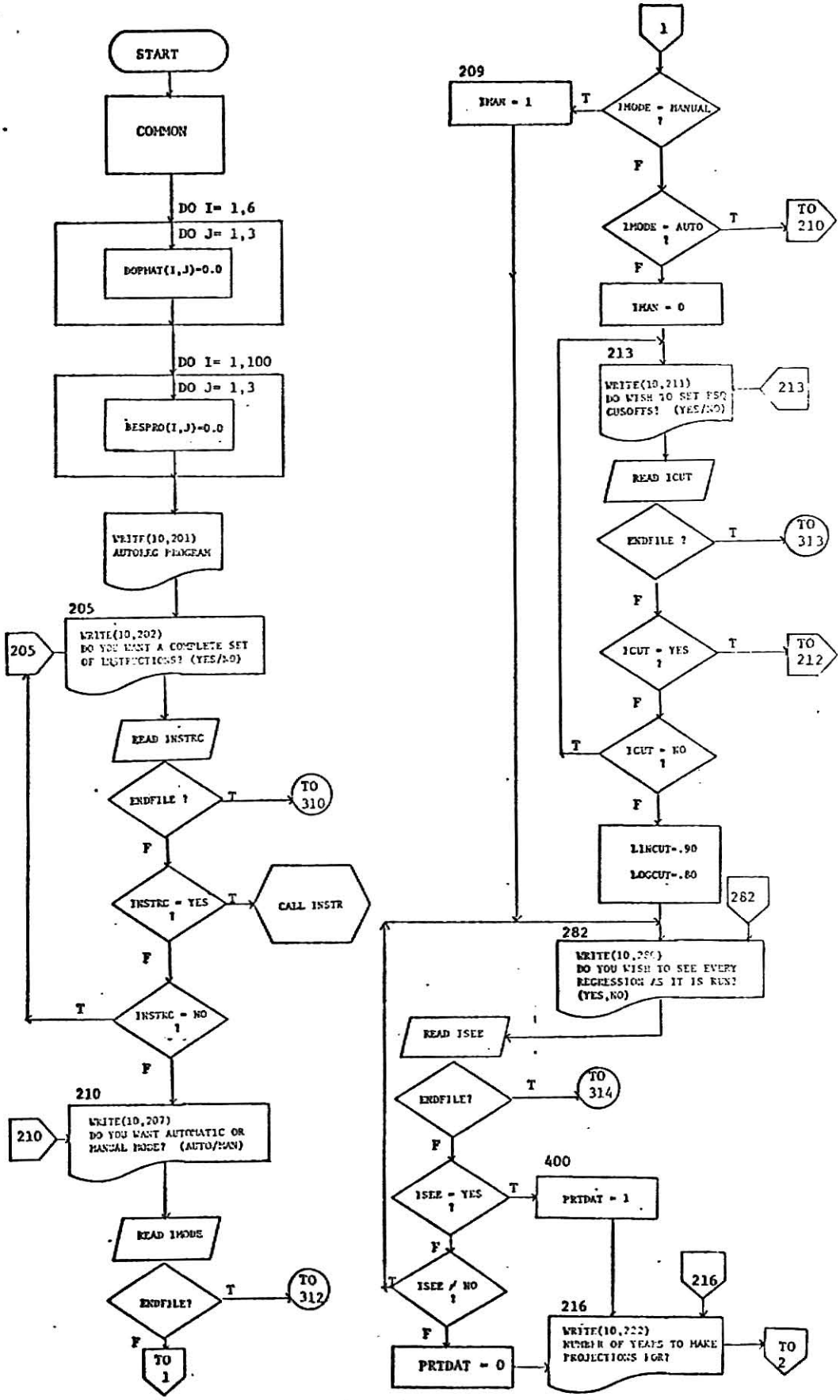
```
      N=IRANGE
      M=N+K
      GO TO 230
  227 WRITE(10,231)
  231 FORMAT('STARTING YEAR OF RANGE WAS GREATER THAN LAST YEAR--TRY AGA
     &IN.')
      GO TO 223
  228 WRITE(10,232)
  232 FORMAT('RANGE EXCEEDS MAXIMUM OF 100--PLEASE INPUT SMALLER RANGE.'
     &)
      GO TO 223
  230 WRITE(10,240)
  240 FORMAT('INPUT THE DATA:')
      DO 300 I=1,N
  300 READ(9,301,END=239) YBASE(I)
  301 FORMAT(F9.2)
      IF(IMAN.NE.1) GO TO 243
      WRITE(10,244)
  244 FORMAT(' '/'BEGIN MANUAL REGRESSION SECTION.')
  246 WRITE(10,245)
  245 FORMAT(' '/'REG. TYPE?')
      READ(9,247,END=320) IREG
  247 FORMAT(A3)
      IF(IREG.EQ.LIN) GO TO 248
      IF(IREG.EQ.LOG) GO TO 249
      IF(IREG.EQ.EXP) GO TO 250
      IF(IREG.EQ.STOP) GO TO 251
      WRITE(10,252)
  252 FORMAT('INVALID REG. TYPE--TRY AGAIN.')
      WRITE(10,206)
      GO TO 246
  243 IREGTP=1
      WRITE(10,702)
  702 FORMAT(' '/'PROJECTIONS AUTOMATICALLY MADE:'/' ')
      PRONAM=99
      CALL REGRES
      IF(LINRSQ.GE.LINCUT) GO TO 251
      PRONAM=98
      IREGTP=2
      CALL REGRES
      IF(LOGRSQ.GE.LOGCUT) GO TO 251
      IREGTP=3
      CALL EXPFLW
      GO TO 251
  248 IREGTP=1
      PRONAM=99
      CALL REGRES
      GO TO 246
  249 IREGTP=2
      PRONAM=98
      CALL REGRES
      GO TO 246
  239 REWIND 9
      WRITE(10,253)
  253 FORMAT('NUMBER OF DATA ITEMS NOT EQUAL TO NUMBER OF BASE YEARS SPE
     &CIFIED ')
  390 WRITE(10,391)
  391 FORMAT('DO YOU WISH TO CHANGE THE NUMBER OF BASE YEARS? (YES/NO)')
      READ(9,247,END=321) IDECS
      IF(IDECS.EQ.NO) GO TO 230
```

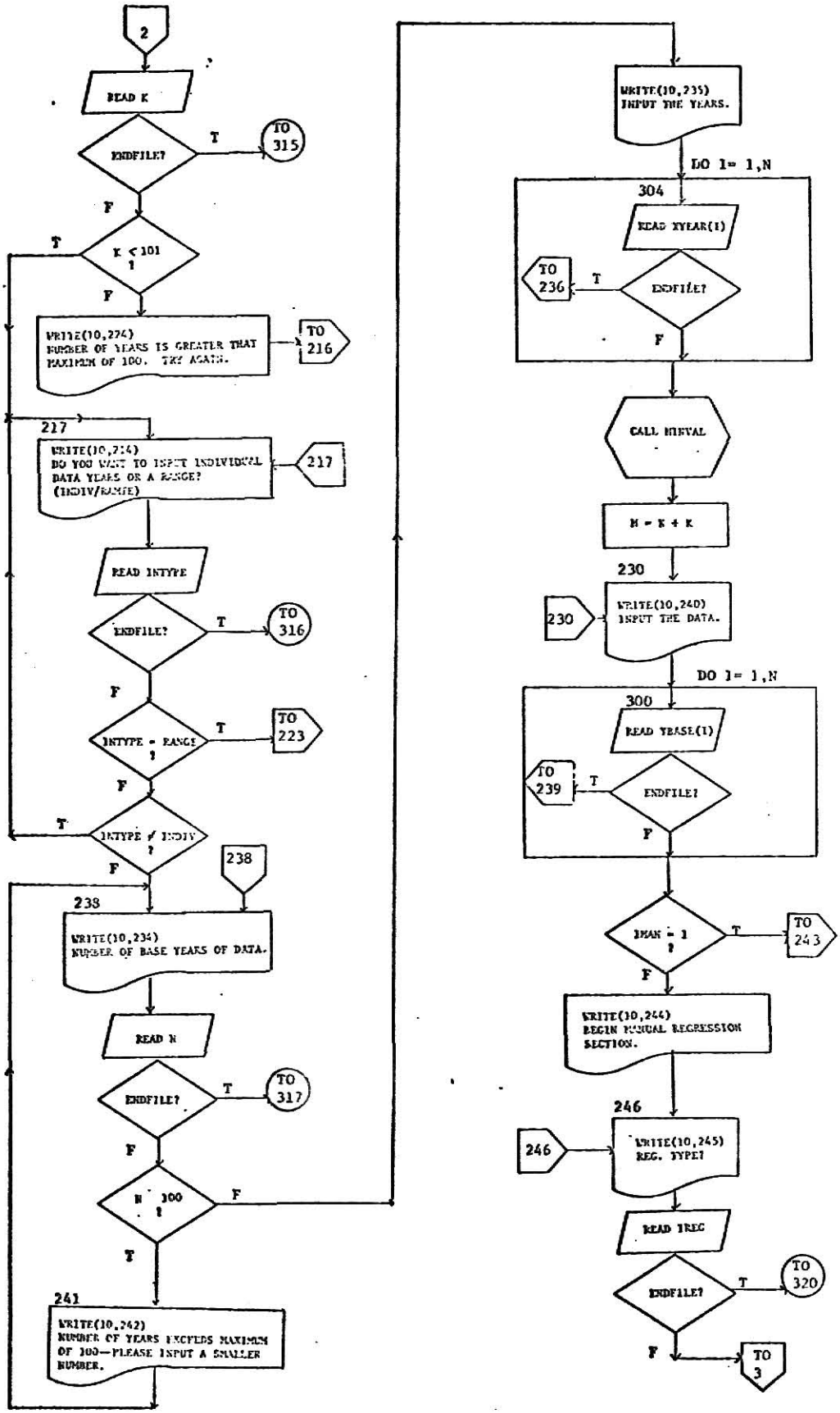
```
      IF(IMAN.EQ.1) GO TO 238
      GO TO 223
  251 WRITE(10,700)
  700 FORMAT(' '/'TYPE ''1''.  THEN POSITION THE PAPER TO THE TOP OF THE
     & NEXT PAGE AND HIT RETURN.')
      READ(9,701,END=901) IM
  701 FORMAT(I1)
      WRITE(10,100)
  100 FORMAT(' '/11X,'YEAR',13X,'BEST THREE PROJECTIONS IN ORDER BY RSQ'
     1)
      WRITE(10,50) XYEAR(1),YBASE(1)
   50 FORMAT(' '/11X,I4,18X,'***BASE**>',2X,F10.2,5X,'<**BASE***')
      WRITE(10,51) XYEAR(2),YBASE(2)
   51 FORMAT(11X,I4,18X,'***DATA**>',2X,F10.2,5X,'<**DATA***')
      WRITE(10,52) (XYEAR(I),YBASE(I),I=3,N)
   52 FORMAT(11X,I4,25X,'**>',2X,F10.2,5X,'<**')
      WRITE(10,53)
   53 FORMAT(' ')
      DO 199 I=1,K
      XSYEAR=XYEAR(N)+I
  199 WRITE(10,101) XSYEAR,(BESPRO(I,J),J=1,3)
  101 FORMAT(11X,I4,3(10X,F10.2))
      WRITE(10,102) (DOPMAT(1,J),J=1,3)
  102 FORMAT (' '/'PROJECTION TYPE=',9X,F10.2,2(10X,F10.2))
      WRITE(10,103) (DOPMAT(2,J),J=1,3)
  103 FORMAT(1X,'RSQ=',10X,3(10X,F10.4))
      WRITE(10,104) (DOPMAT(3,J),J=1,3)
  104 FORMAT(1X,'ALPHA=',8X,3(10X,F10.2))
      WRITE(10,105) (DOPMAT(4,J),J=1,3)
  105 FORMAT(1X,'BETA=',9X,3(10X,F10.2))
      WRITE(10,106) (DOPMAT(5,J),J=1,3)
  106 FORMAT(1X,'XBAR=',9X,3(10X,F10.2))
      WRITE(10,107) (DOPMAT(6,J),J=1,3)
  107 FORMAT(1X,'YBAR=',9X,3(10X,F10.2))
  392 WRITE(10,254)
  254 FORMAT(' '/' '/' '/' '/'DO YOU WISH TO REGRESS ANOTHER DATA SET? (
     &YES/NO)')
      READ(9,247,END=322) IDECS
      IF(IDECS.EQ.NO) GO TO 999
      IF(IDECS.EQ.YES) GO TO 650
      GO TO 251
  250 IREGTP=3
  393 WRITE(10,255)
  255 FORMAT('SPECIFY EXPONENT: SUCH AS 0.9')
      READ(9,306,END=323) EXPON
  306 FORMAT(F3.1)
      CALL REGRES
      GO TO 246
C
C   SET RSQ OF DATA IN BESPRO TO ZERO SO THAT PROSTO WILL OVERWRITE THE
C   OLD DATA WITH THE NEW.
C
  650 DOPMAT(2,3)=0.0
      DOPMAT(2,2)=0.0
      DOPMAT(2,1)=0.0
  652 WRITE(10,651)
  651 FORMAT('DO YOU WISH TO RESET ANY PROJECTION VALUES OTHER THAN DATA
     &?)')
      READ(9,247,END=324)IDECS
      IF(IDECS.EQ.NO) GO TO 230
```

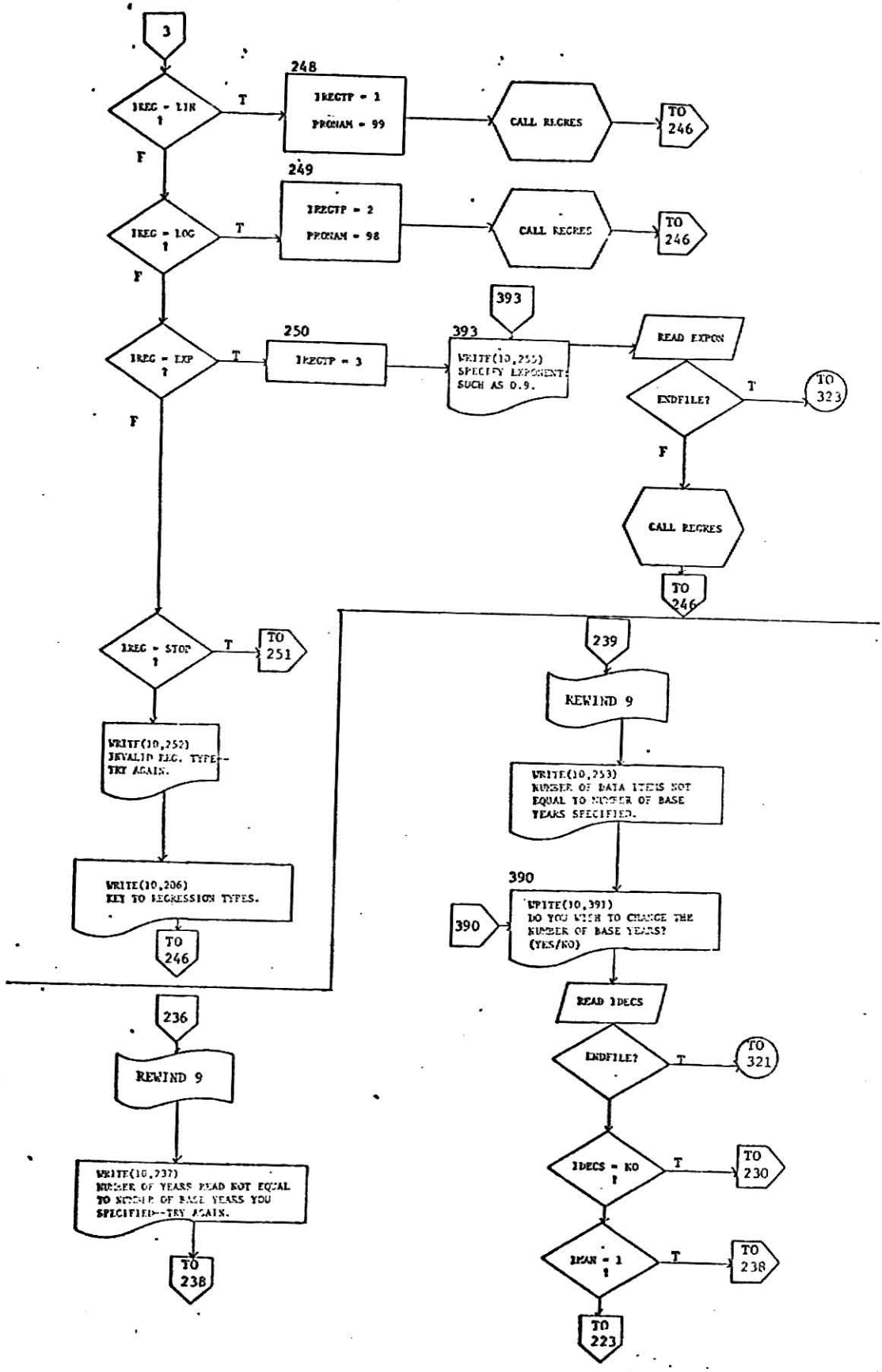
```
      IF(IDECS.NE.YES) GO TO 652
  656 WRITE(10,653)
  653 FORMAT('INPUT THE NUMBER OF THE EARLIEST CHANGE. (7 PRINTS A KEY.)
     &')
      READ(9,655,END=325) IMB
  655 FORMAT(I1)
      GO TO (205,210,213,282,216,217,657),IMB
  657 WRITE(10,654)
  654 FORMAT(' '/'*****KEY TO CHANGES*****'/'1 = INSTRUCTIONS'/'2 = MODE
     &'/'3 = RSQ CUTOFFS'/'4 = SEE EACH REGRESSION'/'5 = # OF PROJECTED
     &YEARS'/'6 = YEARS OF BASE DATA')
      GO TO 656
  901 REWIND 9
      WRITE(10,902)
  902 FORMAT('***ERROR*** YOU PUT A #SIGN AFTER THE LAST DATA ITEM--TRY
     &POSITIONING THE PAPER AGAIN ')
      GO TO 251
C
C   THE REWIND 9'S SET THE WRITE BACK TO FILE FT09F001.
C
  310 REWIND 9
      WRITE(10,311)
  311 FORMAT('***ERROR*** NO ANSWER ENTERED ')
      GO TO 205
  312 REWIND 9
      WRITE(10,311)
      GO TO 210
  313 REWIND 9
      WRITE(10,311)
      GO TO 213
  314 REWIND 9
      WRITE(10,311)
      GO TO 282
  315 REWIND 9
      WRITE(10,311)
      GO TO 216
  316 REWIND 9
      WRITE(10,311)
      GO TO 217
  317 REWIND 9
      WRITE(10,311)
      GO TO 238
  318 REWIND 9
      WRITE(10,311)
      GO TO 212
  319 REWIND 9
      WRITE(10,311)
      GO TO 223
  320 REWIND 9
      WRITE(10,311)
      GO TO 246
  321 REWIND 9
      WRITE(10,311)
      GO TO 390
  322 REWIND 9
      WRITE(10,311)
      GO TO 392
  323 REWIND 9
      WRITE(10,311)
      GO TO 393
```

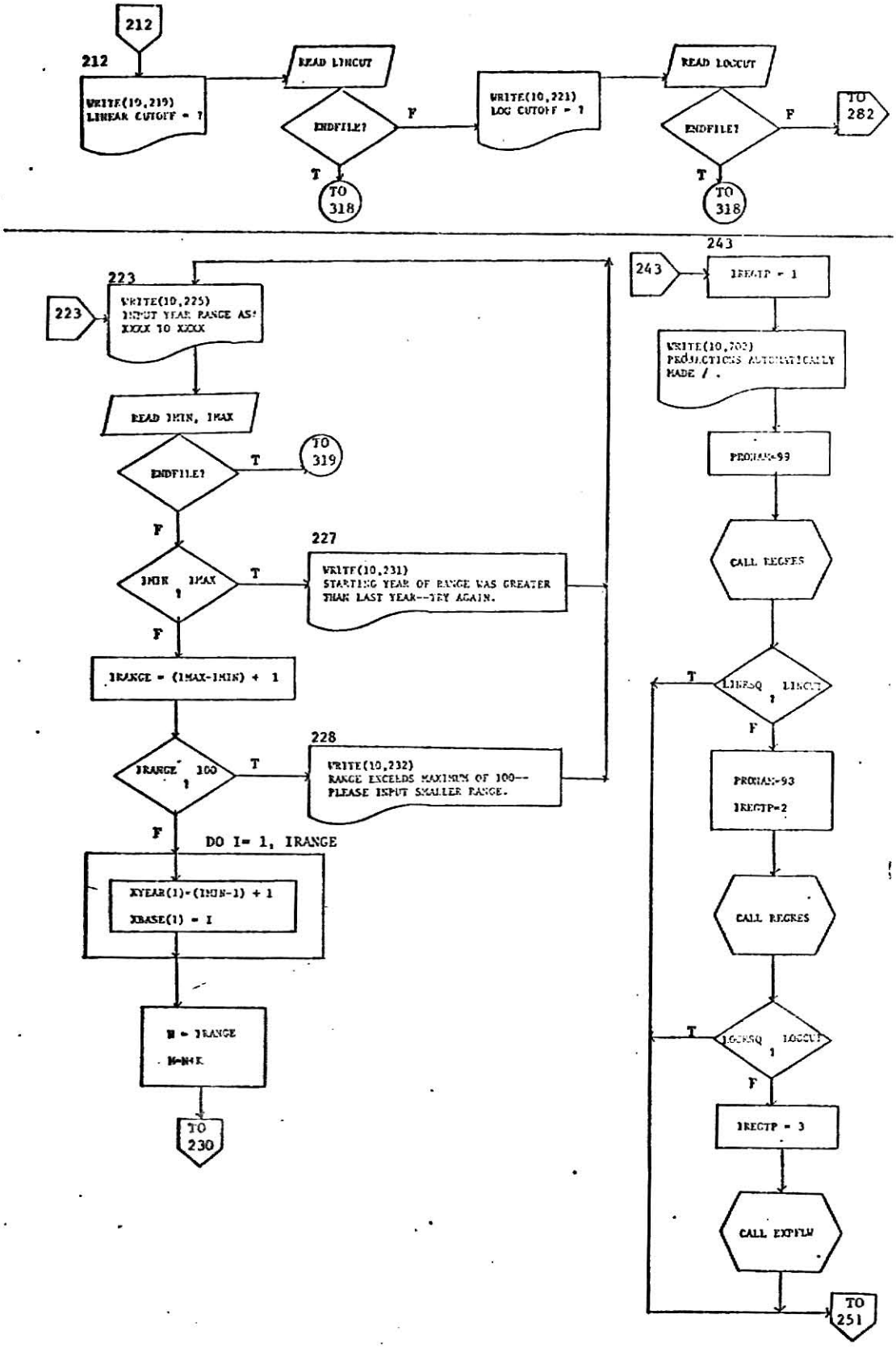
```
324 REWIND 9
    WRITE(10,311)
    GO TO 652
325 REWIND 9
    WRITE(10,311)
    GO TO 656
999 STOP
    END
```

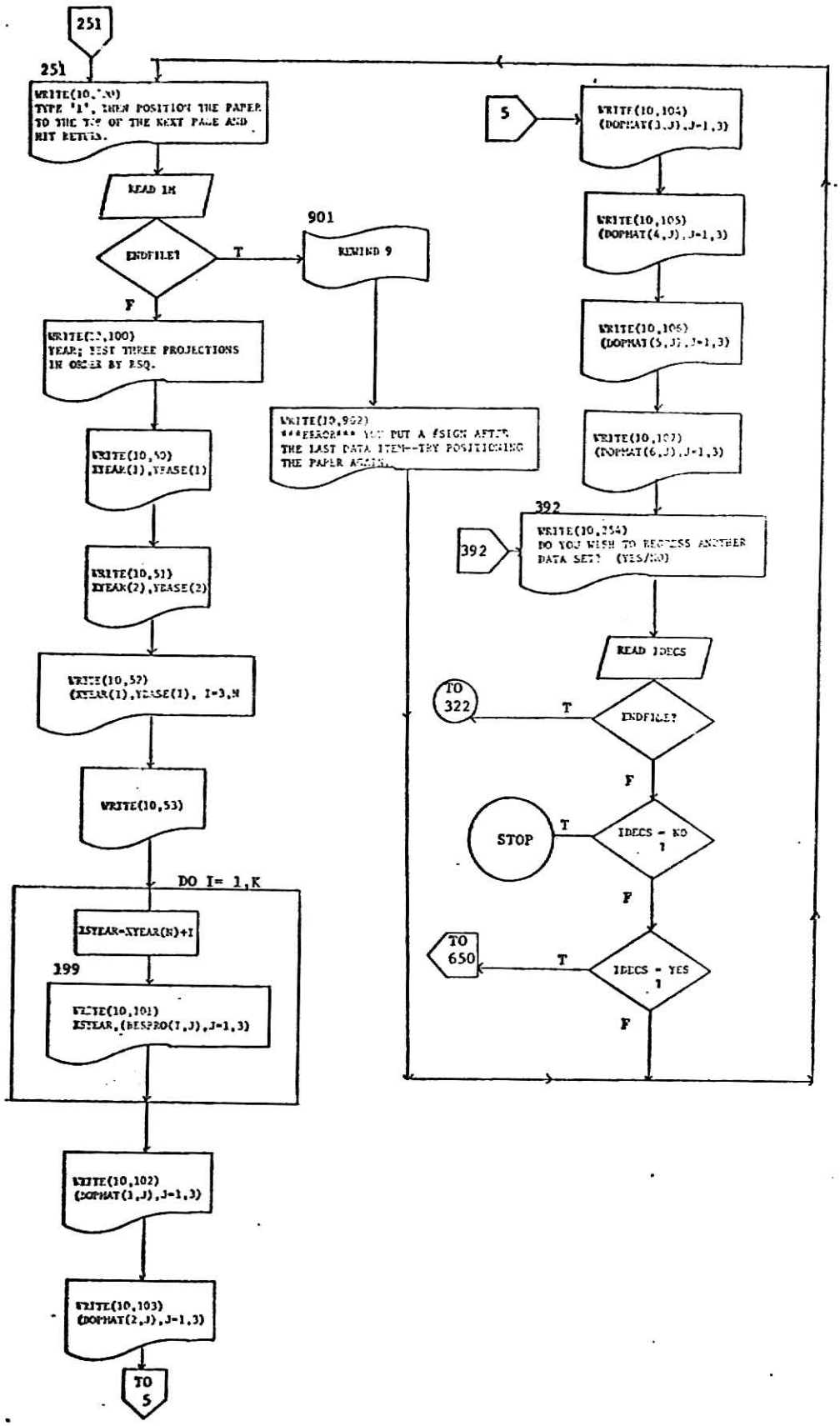
Main Program Flowchart

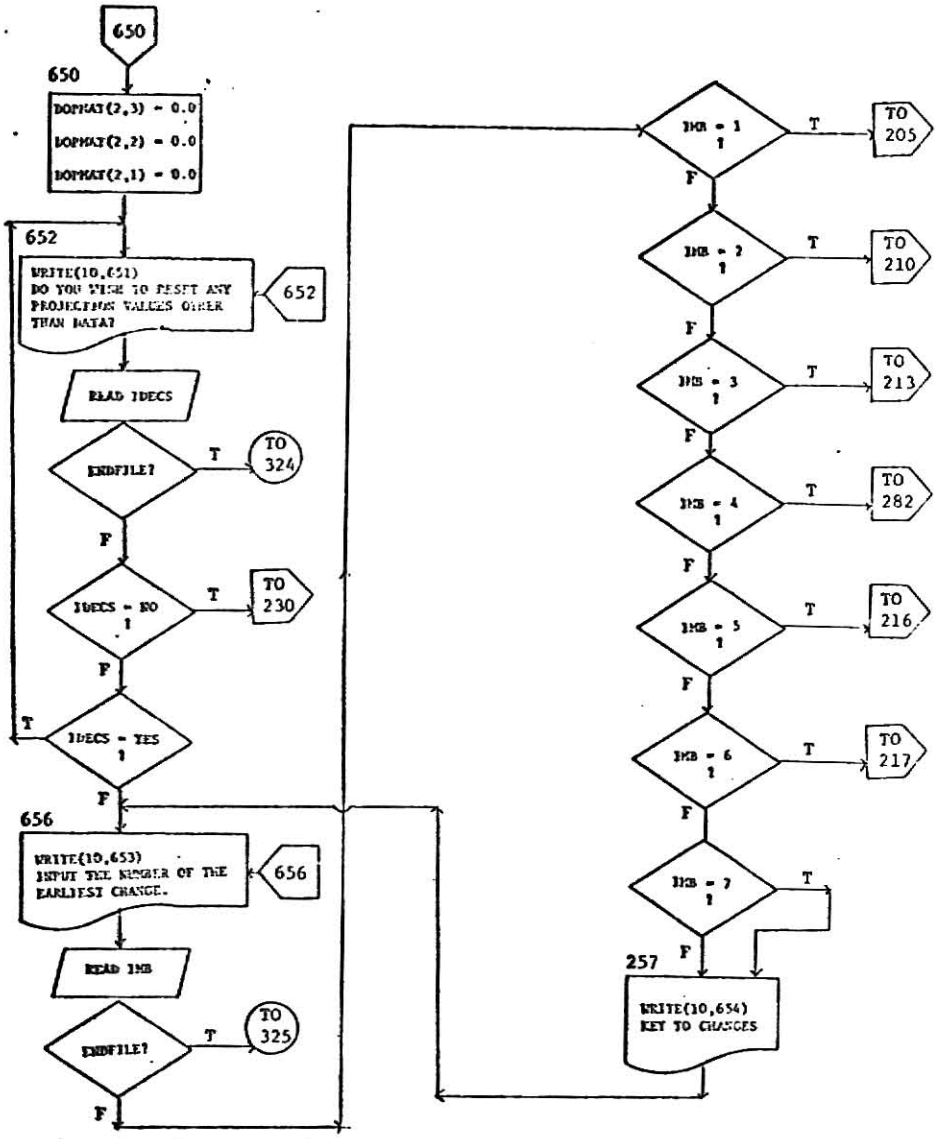


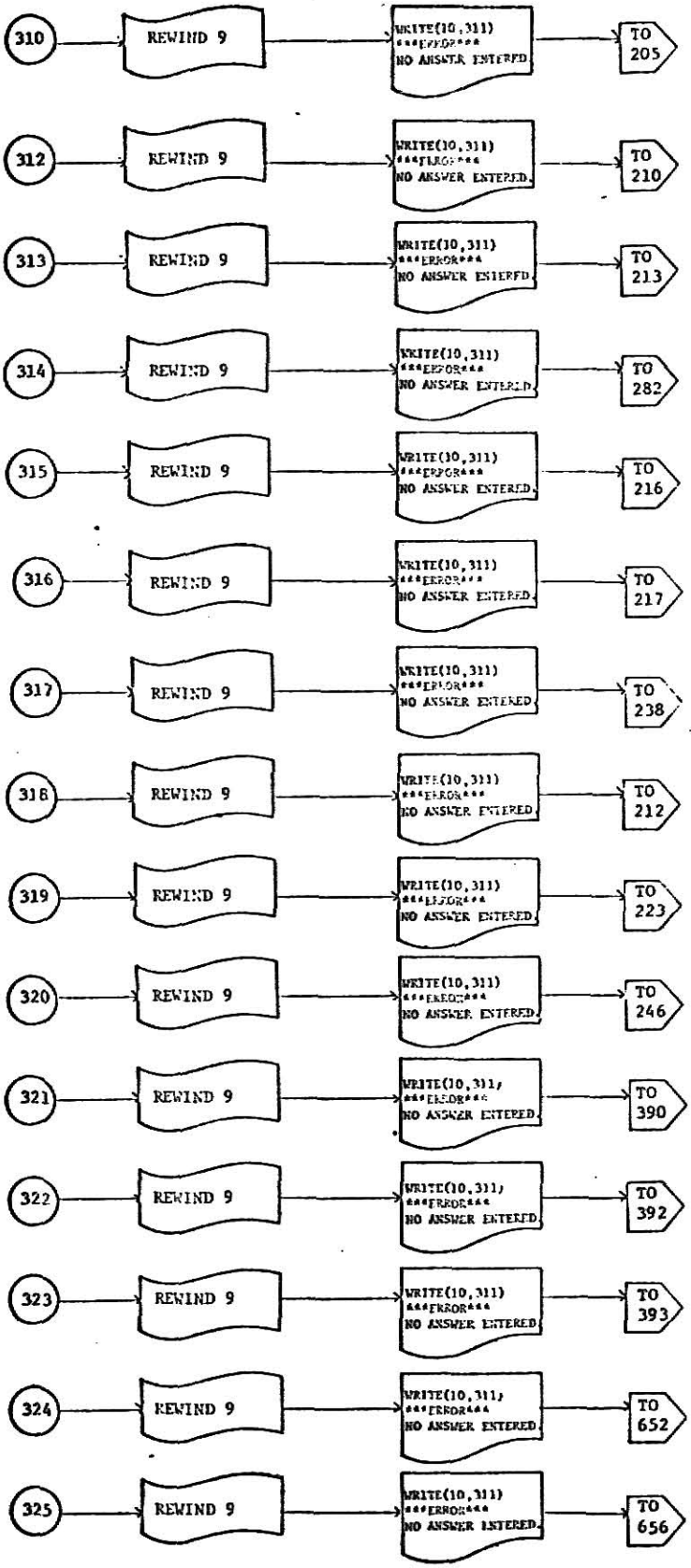

2
READ K
ENDFILE?
T
TO 315
F
K ≤ 101 ?
T
F
WRITE(10,274) NUMBER OF YEARS IS GREATER THAT MAXIMUM OF 100. TRY AGAIN.
TO 216
217
WRITE(10,214) DO YOU WANT TO INPUT INDIVIDUAL DATA YEARS OR A RANGE? (INDIV/RANGE)
217
READ INTYPE
ENDFILE?
T
TO 316
F
INTYPE = RANGE ?
T
TO 223
F
INTYPE ≠ INDIV ?
T
F
238
238
WRITE(10,234) NUMBER OF BASE YEARS OF DATA.
READ N
ENDFILE?
T
TO 317
F
N ≤ 100 ?
F
T
241
WRITE(10,242) NUMBER OF YEARS EXCEEDS MAXIMUM OF 100—PLEASE INPUT A SMALLER NUMBER.
WRITE(10,235) INPUT THE YEARS.
DO I= 1,N
304
READ XYEAR(I)
TO 236
T
ENDFILE?
F
CALL MINVAL
M = K + K
230
230
WRITE(10,240) INPUT THE DATA.
DO I= 1,N
300
READ YBASE(I)
TO 239
T
ENDFILE?
F
IMAN = 1 ?
T
TO 243
F
WRITE(10,244) BEGIN MANUAL REGRESSION SECTION.
246
246
WRITE(10,245) REG. TYPE?
READ IREG
ENDFILE?
T
TO 320
F
TO 3


3
IREG = LIN
T
248
IREGTP = 1
PROGAM = 99
CALL REGRES
TO 246
F
IREG = LOG
T
249
IREGTP = 2
PROGAM = 98
CALL REGRES
TO 246
F
IREG = EXP
T
250
IREGTP = 3
393
393
WRITE(10,255) SPECIFY EXPONENT SUCH AS 0.9.
READ EXPON
ENDFILE?
T
TO 323
F
CALL REGRES
TO 246
F
IREG = STOP
T
TO 251
WRITE(10,252) INVALID REG. TYPE--TRY AGAIN.
WRITE(10,206) KEY TO REGRESSION TYPES.
TO 246
236
REWIND 9
WRITE(10,237) NUMBER OF YEARS READ NOT EQUAL TO NUMBER OF BASE YEARS YOU SPECIFIED--TRY AGAIN.
TO 238
239
REWIND 9
WRITE(10,253) NUMBER OF DATA ITEMS NOT EQUAL TO NUMBER OF BASE YEARS SPECIFIED.
390
390
WRITE(10,391) DO YOU WISH TO CHANGE THE NUMBER OF BASE YEARS? (YES/NO)
READ IDECS
ENDFILE?
T
TO 321
IDECS = NO
T
TO 230
IEAN = 1
T
TO 238
TO 223


212
212
WRITE(10,219)
LINEAR CUTOFF = ?
READ LINCUT
ENDFILE?
F
T
TO 318
WRITE(10,221)
LOG CUTOFF = ?
READ LOGCUT
ENDFILE?
F
T
TO 282
TO 318
223
223
WRITE(10,225)
INPUT YEAR RANGE AS:
XXXX TO XXXX
READ IMIN, IMAX
ENDFILE?
T
TO 319
F
IMIN .GT. IMAX
T
227
WRITE(10,231)
STARTING YEAR OF RANGE WAS GREATER
THAN LAST YEAR--TRY AGAIN.
F
IRANGE = (IMAX-IMIN) + 1
IRANGE .GT. 100
T
228
WRITE(10,232)
RANGE EXCEEDS MAXIMUM OF 100--
PLEASE INPUT SMALLER RANGE.
F
DO I= 1, IRANGE
XYEAR(I)=(IMIN-1) + I
XBASE(I) = I
M = IRANGE
M=M+K
TO 230
243
243
IREGTP = 1
WRITE(10,703)
PROJECTIONS AUTOMATICALLY
MADE / .
PROMAX=99
CALL REGRES
LINESQ .LT. LINCUT
T
F
PROMAX=93
IREGTP=2
CALL REGRES
LOGESQ .LT. LOGCUT
T
F
IREGTP = 3
CALL EXPFLW
TO 251


251
251
WRITE(10,*)
TYPE '1', THEN POSITION THE PAPER
TO THE TOP OF THE NEXT PAGE AND
HIT RETURN.
READ 1M
ENDFILE?
T
901
REWIND 9
F
WRITE(10,100)
YEAR; TEST THREE PROJECTIONS
IN ORDER BY RSQ.
WRITE(10,50)
XYEAR(1),YEASE(1)
WRITE(10,51)
XYEAR(2),YEASE(2)
WRITE(10,52)
(XYEAR(I),YEASE(I), I=3,N
WRITE(10,53)
DO I= 1,K
ISYEAR=XYEAR(N)+I
199
WRITE(10,101)
ISYEAR,(BESPRO(I,J),J=1,3)
WRITE(10,102)
(DOPMAT(1,J),J=1,3)
WRITE(10,103)
(DOPMAT(2,J),J=1,3)
TO
5
WRITE(10,992)
ERROR YOU PUT A #SIGN AFTER
THE LAST DATA ITEM--TRY POSITIONING
THE PAPER AGAIN.
5
WRITE(10,104)
(DOPMAT(3,J),J=1,3)
WRITE(10,105)
(DOPMAT(4,J),J=1,3)
WRITE(10,106)
(DOPMAT(5,J),J=1,3)
WRITE(10,107)
(DOPMAT(6,J),J=1,3)
392
392
WRITE(10,254)
DO YOU WISH TO REGRESS ANOTHER
DATA SET? (YES/NO)
READ IDECS
TO
322
T
ENDFILE?
F
STOP
T
IDECS = NO
1
F
TO
650
T
IDECS = YES
1
F


650
650
DOPMAT(2,3) = 0.0
DOPMAT(2,2) = 0.0
DOPMAT(2,1) = 0.0
652
WRITE(10,651)
DO YOU WISH TO RESET ANY
PROJECTION VALUES OTHER
THAN DATA?
652
READ IDECS
ENDFILE?
T
TO 324
F
IDECS = NO 1
T
TO 230
F
IDECS = YES 1
T
F
656
WRITE(10,653)
INPUT THE NUMBER OF THE
EARLIEST CHANGE.
656
READ IMB
ENDFILE?
T
TO 325
F
IMB = 1
T
TO 205
F
IMB = 2
T
TO 210
F
IMB = 3
T
TO 213
F
IMB = 4
T
TO 282
F
IMB = 5
T
TO 216
F
IMB = 6
T
TO 217
F
IMB = 7
T
F
257
WRITE(10,654)
KEY TO CHANGES


310
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 205
312
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 210
313
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 213
314
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 282
315
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 216
316
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 217
317
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 238
318
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 212
319
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 223
320
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 246
321
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 390
322
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 392
323
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 393
324
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 652
325
REWIND 9
WRITE(10,311)
ERROR
NO ANSWER ENTERED
TO 656

III. Subroutine Regres

This subroutine fits a trend line to the base data input and uses this to extrapolate projected data for the number of years specified by the user in the main program. The trend line is fitted using normal values of Y and transformed values of X that vary according to the value of IREGTP and EXPON. If IREGTP equals 1 the X's aren't transformed and the resulting linear projection's equation takes the form $Y = a + bX$. When IREGTP equals 2, the X's are transformed by taking their logarithms with the resulting projection equation being of the form $Y = a + b(\log X)$. When IREGTP is equal to 3, the X's are taken to the EXPON power before calculating the trend line equation. This equation takes the form $Y = a + b(X^{EXPON})$ where EXPON is input by the user or passed to REGRES by Expflw. When EXPON is passed in from Expflw, it always lies between 0.4 and 1.4 inclusive.

The subroutine works by first transforming the X values. Next the equation of the trend line is calculated along with the value of RSQ. All computations for finding the trend line equation are done in double precision. The projected values of Y are found by transforming future values of X in the same way that the base data X's were transformed and then plugging these future transformed X values into the calculated regression equation.

The name of the projection type and the RSQ of the projection are printed next. Finally, if the user indicated so in the main program, a copy of the base and projected data is printed to the terminal. Then, subroutine Prosto is called to store the results. If the user didn't ask to see each regression as it was run, then REGRES would call Prosto without printing the base and projected data to the terminal.

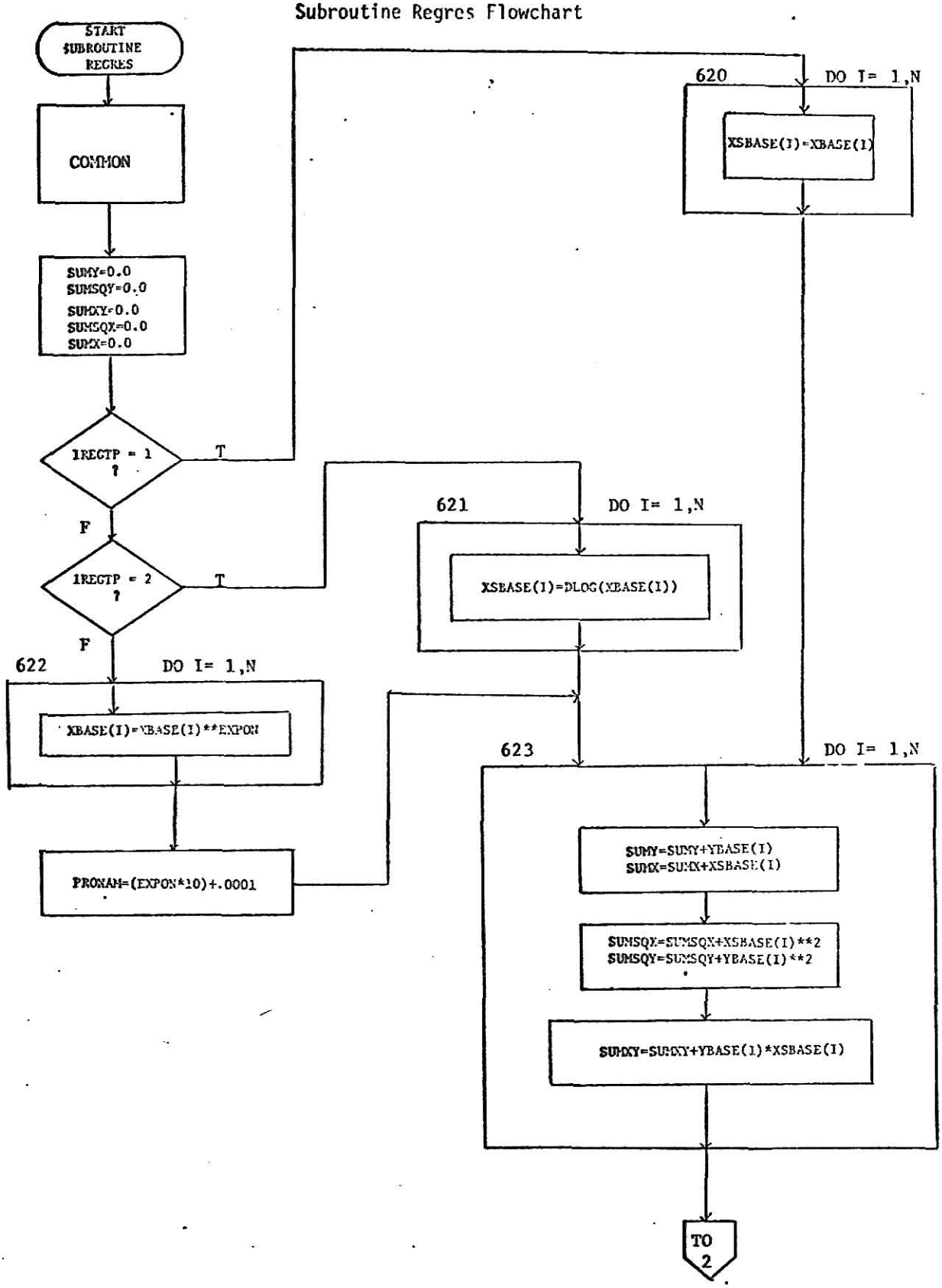
A flowchart of REGRES follows a listing of it.

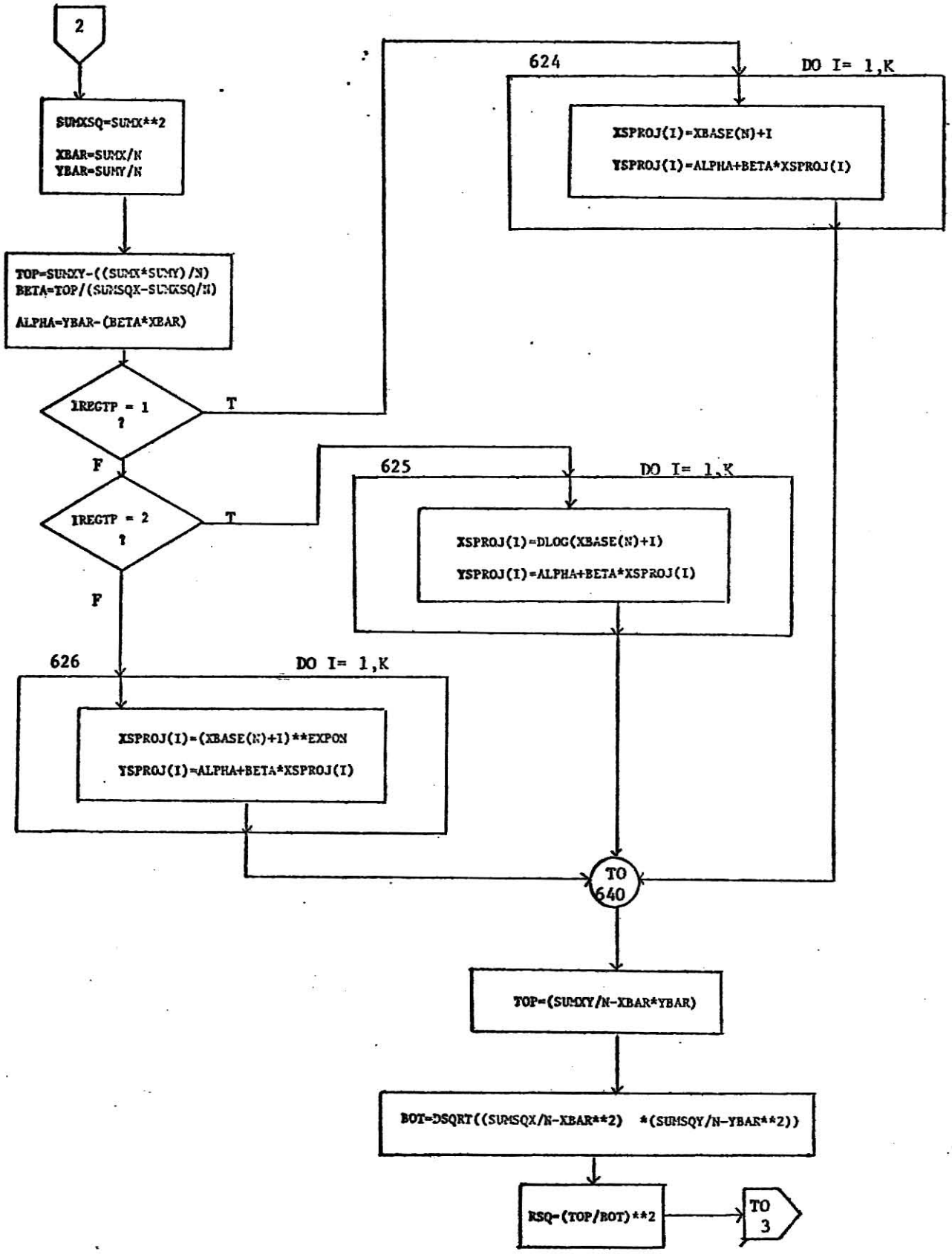
Subroutine Regres

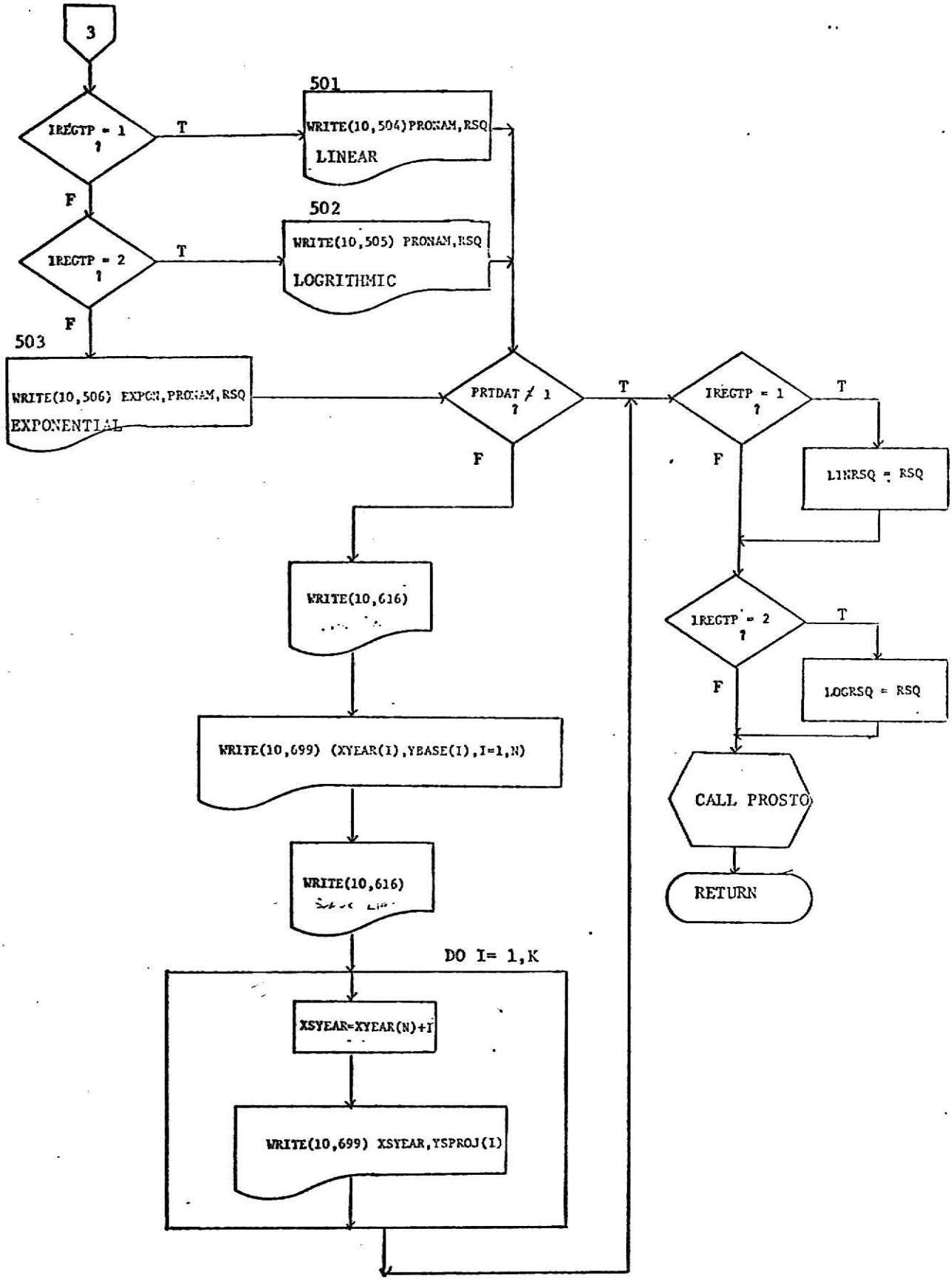
```
C*************************************************************************
C                                                                        *
C   THIS SUBROUTINE PROJECTS INPUT DATA ACCORDING TO THE VALUE OF        *
C   IREGTP.  IF:  IREGTP =1  ; THE PROJECTION IS LINEAR                  *
C                        2  ; THE PROJECTION IS LOGARITHMIC              *
C                        3  ; THE PROJECTION IS EXPONENTIAL              *
C                                                                        *
C   THE LOG PROJECTION IS OF THE FORM Y=A+B LOGX.  THE EXPONENTIAL       *
C   PROJECTION IS OF THE FORM Y=A+BX**EXPON; WHERE EXPON IS INPUT TO     *
C   THE SUBROUTINE AND LIES BETWEEN .4 AND 1.4, EXCLUDING 1.0.           *
C                                                                        *
C*************************************************************************
      SUBROUTINE REGRES
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ
      DIMENSION XSPROJ(100)
      INTEGER PRTDAT,XYEAR,XSYEAR,PRONAM
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRONAM,LINRSQ,LOGRSQ
      SUMXY=0.0000000000
      SUMSQX=0.0000000000
      SUMX=0.0000000000
      SUMY=0.0000000000
      SUMSQY=0.0000000000
C*************************************************************************
C                                                                        *
C   THIS SECTION MANIPULATES THE INPUT X VALUES ACCORDING TO THE TYPE    *
C   OF PROJECTION SPECIFIED BY IREGTP.  THE X VALUES ARE THE TRANS-      *
C   FORMED X VALUES OUTPUT BY SUBROUTINE MINVAL.                         *
C                                                                        *
C*************************************************************************
      GO TO (620,621,622),IREGTP
  622 DO 708 I=1,N
  708 XSBASE(I)=XBASE(I)**EXPON
      PRONAM=(EXPON*10)+.0001
      GO TO 623
  621 DO 700 I=1,N
  700 XSBASE(I)=DLOG(XBASE(I))
      GO TO 623
  620 DO 701 I=1,N
  701 XSBASE(I)=XBASE(I)
C*************************************************************************
C                                                                        *
C   THIS SECTION PROJECTS THE INPUT DATA K YEARS AND ASSIGNS THE         *
C   RESULTS TO OUTPUT VECTOR YSPROJ.                                     *
C                                                                        *
C*************************************************************************
  623 DO 600 I=1,N
      SUMY=SUMY+YBASE(I)
      SUMX=SUMX+XSBASE(I)
      SUMSQX=SUMSQX+XSBASE(I)**2
      SUMSQY=SUMSQY+YBASE(I)**2
  600 SUMXY=SUMXY+YBASE(I)*XSBASE(I)
      SUMXSQ=SUMX**2
      XBAR=SUMX/N
      YBAR=SUMY/N
      TOP=SUMXY-((SUMX*SUMY)/N)
      BETA=TOP/(SUMSQX-SUMXSQ/N)
      ALPHA=YBAR-(BETA*XBAR)
```

```
      GO TO (624,625,626),IREGTP
  624 DO 709 I=1,K
      XSPROJ(I)=XBASE(N)+I
  709 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
      GO TO 640
  626 DO 710 I=1,K
      XSPROJ(I)=(XBASE(N)+I)**EXPON
  710 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
      GO TO 640
  625 DO 601 I=1,K
      XSPROJ(I)=DLOG(XBASE(N)+I)
  601 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
C*************************************************************************
C                                                                        *
C   THIS SECTION COMPUTES THE VALUE OF RSQ AND WRITES THE PROJECTION     *
C   NAME AND RSQ VALUE TO THE TERMINAL.                                  *
C                                                                        *
C*************************************************************************
  640 TOP=(SUMXY/N-XBAR*YBAR)
      BOT=DSQRT((SUMSQX/N-XBAR**2)*(SUMSQY/N-YBAR**2))
      RSQ=(TOP/BOT)**2
      GO TO (501,502,503),IREGTP
  501 WRITE(10,504) PRONAM,RSQ
  504 FORMAT('*****LINEAR(',I2,')***** RSQ= ',F6.4)
      GO TO 599
  502 WRITE(10,505) PRONAM,RSQ
  505 FORMAT('**LOGARITHMIC(',I2,')*** RSQ= ',F6.4)
      GO TO 599
  503 WRITE(10,506) EXPON,PRONAM,RSQ
  506 FORMAT('*',F3.1,' EXPONENTIAL(',I2,') RSQ= ',F6.4)
  599 IF(PRTCAT.NE.1) GO TO 996
C*************************************************************************
C                                                                        *
C   THIS SECTION PRINTS THE INPUT DATA FOR THE BASE YEARS AND THE        *
C   PROJECTED YEARS' DATA FOR EACH PROJECTION MADE IF PRTCAT = 1.        *
C                                                                        *
C*************************************************************************
      WRITE(10,616)
  616 FORMAT(' ')
      WRITE(10,699) (XYEAR(I),YBASE(I),I=1,N)
      WRITE(10,616)
      DO 702 I=1,K
      XSYEAR=XYEAR(N)+I
  702 WRITE (10,699) XSYEAR,YSPROJ(I)
  699 FORMAT (1X,I4,10X,F8.2)
  996 IF(IREGTP.EQ.1) LINRSQ=RSQ
      CALL PRCSTG
  998 RETURN
      END
```

Subroutine Regres Flowchart




2
SUMXSQ=SUMX**2
XBAR=SUMX/N
YBAR=SUMY/N
TOP=SUMXY-((SUMX*SUMY)/N)
BETA=TOP/(SUMSQX-SUMXSQ/N)
ALPHA=YBAR-(BETA*XBAR)
IREGTP = 1 ?
T
F
624
DO I= 1,K
XSPROJ(I)=XBASE(N)+I
YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
IREGTP = 2 ?
T
F
625
DO I= 1,K
XSPROJ(I)=DLOG(XBASE(N)+I)
YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
626
DO I= 1,K
XSPROJ(I)=(XBASE(N)+1)**EXPON
YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
TO 640
TOP=(SUMXY/N-XBAR*YBAR)
BOT=DSQRT((SUMSQX/N-XBAR**2) *(SUMSQY/N-YBAR**2))
RSQ=(TOP/BOT)**2
TO 3


3
IREGTP = 1
T
501
WRITE(10,504)PRONAM,RSQ
LINEAR
F
IREGTP = 2
T
502
WRITE(10,505) PRONAM,RSQ
LOGRITHMIC
F
503
WRITE(10,506) EXPON,PRONAM,RSQ
EXPONENTIAL
PRTDAT ≠ 1
T
F
IREGTP = 1
T
F
LINRSQ = RSQ
IREGTP = 2
T
F
LOGRSQ = RSQ
CALL PROSTO
RETURN
WRITE(10,616)
WRITE(10,699) (XYEAR(I),YBASE(I),I=1,N)
WRITE(10,616)
DO I= 1,K
XSYEAR=XYEAR(N)+I
WRITE(10,699) XSYEAR,YSPROJ(I)

IV. Subroutine Instr

This subroutine, when called, writes to the terminal a short set of instructions. These instructions, shown on the following page, detail how to use the overall program. The calling of this subroutine is optional and the user must request it. That way, the user only has to wait for the instructions to be typed if he actually needs a copy. He is given the option to call instructions at the start of the program and again each time he chooses to run another set of data.

A flowchart is not shown for the subroutine due to its simplicity.

********** AUTOREG REGRESSION **********

THIS PROGRAM ALLOWS YOU TO RUN LINEAR,LOG,AND EXPONENTIAL REGRESSIONS ON DATA YOU INPUT AT A TERMINAL. IN RUNNING AUTOREG, YOU ARE ALLOWED TO MAKE THE FOLLOWING DECISIONS: (AFTER YOU TYPE IN THE ANSWER TO A DECISION, ALWAYS HIT RETURN.)

1) SELECT AUTOMATIC OR MANUAL MODE. IF YOU SELECT MANUAL, YOU CONTROL WHICH REGRESSION TYPES ARE PERFORMED. (SEE 5 BELOW FOR THE TYPES.) IF YOU SELECT AUTO-MATIC,THE COMPUTER WILL SELECT THE PROJECTIONS TO BE RUN ACCORDING TO THE FOL-LOWING SCHEME:

1ST-LINEAR
2ND-LOGARITHMIC, IF LINEAR RSQ (LINRSQ) IS LESS THAN .90
3RD-EXPONENTIAL, IF LOG RSQ (LOGRSQ) IS LESS THAN .80
EXPONENTIAL REGRESSION IS PERFORMED USING A BUILT-IN DECISION TREE UNTIL ALL EXPONENTS BETWEEN 0.4 AND 1.4 ARE TRIED IMPLICITLY.

2) IF YOU SELECTED AUTOMATIC, DECIDE IF YOU WISH TO OVERRIDE THE RSQ CUTOFFS DISCUSSED ABOVE. IF YOU DO, RESPOND TO LINRSQ= WITH THE CUTOFF FOR TRYING A LOG PROJECTION. (WRITE AS A WHOLE NUMBER--EX: FOR .99 RSQ, INPUT 99.) RESPOND TO LOGRSQ= WITH THE CUTOFF FOR TRYING EXPONENTIAL PROJECTIONS.

3) DECIDE IF YOU WISH EACH PROJECTION MADE TO BE PRINTED AS THE PROJECTION IS MADE.

4) DECIDE WHETHER TO INPUT THE YEARS OF THE DATA INDIVIDUALLY AS 1963,1964, ETC. OR AS A RANGE IN THE FORM: 1963 TO 1975. YEARS MUST BE INPUT INDIVIDUALLY IF A YEAR'S DATA IS MISSING. (OMIT THE YEAR NUMBER FOR THOSE YEARS THAT DATA IS NOT AVAILABLE FOR.)

5) SELECT THE NUMBER OF YEARS OF BASE DATA YOU ARE GOING TO INPUT (MAX=100) AND SELECT THE NUMBER OF YEARS INTO THE FUTURE THAT YOU WISH PROJECTIONS TO BE MADE FOR (MAX=100). (WRITE NUMBERS BELOW 10 AS EX: 09.)

6) IF YOU SELECTED MANUAL MODE, EVERY TIME THE PROMPT--REG. TYPE?--APPEARS INPUT ONE OF THE FOLLOWING PROJECTION REQUESTS:

LIN = LINEAR PROJECTION OF THE FORM Y=A+BX
LOG = LOGARITHMIC PROJECTION OF THE FORM Y=A+B*LOGX
EXP = EXPONENTIAL PROJECTION OF THE FORM Y=A+B*(X TO THE C POWER)
(YOU WILL BE ASKED TO INPUT THE EXPONENT (C))
STP = TERMINATE PROJECTIONS ON PRESENT SET OF DATA

7) YOU WILL BE ASKED TO INPUT THE YEARS AND DATA. INPUT THEM WITH A # SIGN BETWEEN EACH ITEM. AT THE END OF A LINE OF INPUT, HIT RETURN AND INPUT ALL DATA EITHER WITH A DECIMAL IN OR BEHIND THE NUMBER. FOR EXAMPLE A SAMPLE LINE OF INPUT MIGHT LOOK LIKE: 123.#123.45#123.
DON'T PUT A #SIGN AFTER THE LAST DATA ITEM. BE SURE TO INCLUDE A DECIMAL POINT FOR EACH DATA ITEM!
YOU WILL BE ALLOWED TO INPUT THE REST OF THE DATA ON THE NEXT LINE OR, IF ALL THE DATA HAS BEEN INPUT, THE RETURN WILL SIGNAL THE PROGRAM TO GO TO THE NEXT STEP.

AS EACH PROJECTION IS MADE, IN EITHER MODE, THE NAME OF THE PROJECTION TYPE, FOLLOWED BY A NUMBER IN PARENTHESIS AND THE RSQ OF THAT PROJECTION TYPE, IS PRINTED. WHEN YOU STOP, THE BEST 3 PROJECTIONS WILL BE PRINTED ALONG WITH THEIR PROJECTION TYPES AND RSQ. IN EITHER OF THE TWO MODES, THE TYPES OF PROJECTIONS MADE WILL BE PRINTED, ALONG WITH THE BEST 3 PROJECTIONS, THEIR RSQ'S AND PROJECTION TYPES. THE NUMBER, MENTIONED ABOVE, THAT FOLLOWS THE PROJECTION NAME IS THE KEY TO THE NUMBER GIVEN AS THE PROJECTION TYPE AT THE BOTTOM OF THE BEST THREE PROJECTIONS.

AT THE END OF A REGRESSION, IF YOU CHOOSE TO REGRESS ANOTHER SET OF DATA, YOU CAN PICK THE VALUES TO RESET, SUCH AS THE MODE--AUTO OR MANUAL. ALWAYS SELECT THE NUMBER OF THE EARLIEST CHANGE YOU WISH TO MAKE AS YOU WILL CAUSE THE PROGRAM TO RETURN TO THE POINT COINCIDING WITH THE PARAMETER NO'S NUMBER YOU SELECTED, AND THUS YOU WILL ALSO GET TO INPUT DECISIONS FOR ALL PARAMETERS WITH LARGER NUMBERS THAN THE ONE YOU SELECTED.

Program Listing

```
      SUBROUTINE INSTR (*)
C*************************************************************************
C                                                                        *
C   THIS SUBROUTINE WRITES A SHORT SET OF INSTRUCTIONS TO THE USER.      *
C                                                                        *
C*************************************************************************
      WRITE(10,1)
      WRITE(10,2)
      WRITE(10,3)
      WRITE(10,4)
      WRITE(10,5)
      WRITE(10,6)
      WRITE(10,7)
      WRITE(10,8)
      WRITE(10,9)
      WRITE(10,10)
      WRITE(10,11)
      WRITE(10,12)
      WRITE(10,13)
      WRITE(10,14)
      WRITE(10,15)
      WRITE(10,16)
      WRITE(10,17)
      WRITE(10,41)
      WRITE(10,42)
      WRITE(10,18)
      WRITE(10,19)
      WRITE(10,20)
      WRITE(10,21)
      WRITE(10,22)
      WRITE(10,23)
      WRITE(10,24)
      WRITE(10,25)
      WRITE(10,26)
      WRITE(10,27)
      WRITE(10,28)
      WRITE(10,29)
      WRITE(10,30)
      WRITE(10,31)
      WRITE(10,32)
      WRITE(10,33)
      WRITE(10,50)
      WRITE(10,51)
      WRITE(10,53)
      WRITE(10,54)
      WRITE(10,34)
      WRITE(10,35)
      WRITE(10,43)
      WRITE(10,44)
      WRITE(10,37)
      WRITE(10,38)
      WRITE(10,39)
      WRITE(10,40)
      WRITE(10,45)
      WRITE(10,46)
      WRITE(10,60)
      WRITE(10,61)
      WRITE(10,62)
      WRITE(10,63)
      WRITE(10,64)
```

```
      WRITE(10,65)
    1 FORMAT('****************************** AUTOREG REGRESSION ********
     &************************')
    2 FORMAT(' '/,' '/,'THIS PROGRAM ALLOWS YOU TO RUN LINEAR,LOG,AND EX
     &PONENTIAL REGRESSIONS ON DATA YOU')
    3 FORMAT('INPUT AT A TERMINAL.  IN RUNNING AUTOREG, YOU ARE ALLOWED
     &TO MAKE THE FOLLOWING')
    4 FORMAT('DECISIONS:  (AFTER YOU TYPE IN THE ANSWER TO A DECISION, A
     &LWAYS HIT RETURN.)')
    5 FORMAT(' '/'     1) SELECT AUTOMATIC OR MANUAL MODE.  IF YOU SELEC
     &T MANUAL, YOU CONTROL WHICH')
    6 FORMAT('REGRESSION TYPES ARE PERFORMED.  (SEE 5 BELOW FOR THE TYPE
     &S.)  IF YOU SELECT AUTO')
    7 FORMAT('-MATIC,THE COMPUTER WILL SELECT THE PROJECTIONS TO BE RUN
     &ACCORDING TO THE FOL-')
    8 FORMAT('LOWING SCHEME:')
    9 FORMAT('     1ST-LINEAR')
   10 FORMAT('     2ND-LOGARITHMIC, IF LINEAR RSQ (LINRSQ) IS LESS THAN
     &.90')
   11 FORMAT('     3RD-EXPONENTIAL, IF LOG RSQ (LOGRSQ) IS LESS THAN .80
     &')
   12 FORMAT('         EXPONENTIAL REGRESSION IS PERFORMED USING A BUILT
     &-IN DECISION TREE UNTIL')
   13 FORMAT('         ALL EXPONENTS BETWEEN 0.4 AND 1.4 ARE TRIED IMPLI
     &CITLY.')
   14 FORMAT(' '/'     2) IF YOU SELECTED AUTOMATIC, DECIDE IF YOU WISH
     & TO OVERRIDE THE RSQ CUTOFFS')
   15 FORMAT('DISCUSSED ABOVE.  IF YOU DO, RESPOND TO LINRSQ= WITH THE C
     &UTOFF FOR TRYING A LOG')
   16 FORMAT('PROJECTION.  (WRITE AS A WHOLE NUMBER--EX: FOR .90 RSQ, IN
     &PUT 90.)  RESPOND TO ')
   17 FORMAT('LOGRSQ= WITH THE CUTOFF FOR TRYING EXPONENTIAL PROJECTIONS
     &.')
   41 FORMAT(' '/'     3) DECIDE IF YOU WISH EACH PROJECTION MADE TO BE
     &PRINTED AS THE PROJECTION')
   42 FORMAT('IS MADE.')
   18 FORMAT(' '/'     4) DECIDE WHETHER TO INPUT THE YEARS OF THE DATA
     &INDIVIDUALLY AS 1963,1964,')
   19 FORMAT('ETC. OR AS A RANGE IN THE FORM:  1963 TO 1975.  YEARS MUST
     & BE INPUT INDIVIDUALLY')
   20 FORMAT('IF A YEAR''S DATA IS MISSING.  (OMIT THE YEAR NUMBER FOR T
     &HOSE YEARS THAT DATA IS')
   21 FORMAT('NOT AVAILABLE FOR.)')
   22 FORMAT(' '/'     5) SELECT THE NUMBER OF YEARS OF BASE DATA YOU AR
     &E GOING TO INPUT (MAX=100)')
   23 FORMAT('AND SELECT THE NUMBER OF YEARS INTO THE FUTURE THAT YOU WI
     &SH PROJECTIONS TO BE')
   24 FORMAT('MADE FOR (MAX=100).  (WRITE NUMBERS BELOW 10 AS EX: 09.)')
   25 FORMAT(' '/'     6) IF YOU SELECTED MANUAL MODE, EVERY TIME THE PR
     &OMPT--REG. TYPE?--APPEARS')
   26 FORMAT('INPUT ONE OF THE FOLLOWING PROJECTION REQUESTS:')
   27 FORMAT('     LIN = LINEAR PROJECTION OF THE FORM Y=A+BX')
   28 FORMAT('     LOG = LOGARITHMIC PROJECTION OF THE FORM Y=A+B*LOGX')
   29 FORMAT('     EXP = EXPONENTIAL PROJECTION OF THE FORM Y=A+B*(X TO
     &THE C POWER)')
   30 FORMAT('           (YOU WILL BE ASKED TO INPUT THE EXPONENT (C))')
   31 FORMAT('     STP = TERMINATE PROJECTIONS ON PRESENT SET OF DATA')
   32 FORMAT(' '/'     7) YOU WILL BE ASKED TO INPUT THE YEARS AND DATA.
     &  INPUT THEM WITH A')
   33 FORMAT('# SIGN BETWEEN EACH ITEM.  AT THE END OF A LINE OF INPUT,
```

```
     &  HIT RETURN AND')
  34 FORMAT('YOU WILL BE ALLOWED TO INPUT THE REST OF THE DATA ON THE N
     &EXT LINE OR, IF ALL THE')
  35 FORMAT('DATA HAS BEEN INPUT, THE RETURN WILL SIGNAL THE PROGRAM TO
     & GO TO THE NEXT STEP.')
  50 FORMAT('INPUT ALL DATA EITHER WITH A DECIMAL IN OR BEHIND THE NUMB
     &ER.  FOR EXAMPLE')
  51 FORMAT('A SAMPLE LINE OF INPUT MIGHT LOOK LIKE: 123.#123.45#123.')
  53 FORMAT('DON''T PUT A #SIGN AFTER THE LAST DATA ITEM.  BE SURE TO I
     &NCLUDE A DECIMAL')
  54 FORMAT('POINT FOR EACH DATA ITEM ')
  37 FORMAT('PRINTED.  WHEN YOU STOP, THE BEST 3 PROJECTIONS WILL BE PR
     &INTED ALONG WITH THEIR')
  38 FORMAT('PROJECTION TYPES AND RSQ.  IN EITHER OF THE TWO MODES, THE
     & TYPES OF PROJECTIONS')
  39 FORMAT('MADE WILL BE PRINTED, ALONG WITH THE BEST 3 PROJECTIONS, T
     &HEIR RSQ''S AND')
  43 FORMAT(' '/' '/'AS EACH PROJECTION IS MADE, IN EITHER MODE, THE NA
     &ME OF THE PROJECTION TYPE,')
  44 FORMAT('FOLLOWED BY A NUMBER IN PARENTHESIS AND THE RSQ OF THAT PR
     &OJECTION TYPE, IS')
  40 FORMAT('PROJECTION TYPES.  THE NUMBER, MENTIONED ABOVE, THAT FOLLO
     &WS THE PROJECTION')
  45 FORMAT('NAME IS THE KEY TO THE NUMBER GIVEN AS THE PROJECTION TYPE
     & AT THE BOTTOM OF')
  46 FORMAT('THE BEST THREE PROJECTIONS.')
  60 FORMAT(' '/'AT THE END OF A REGRESSION, IF YOU CHOOSE TO REGRESS A
     &NOTHER SET OF DATA,')
  61 FORMAT('YOU CAN PICK THE VALUES TO RESET, SUCH AS THE MODE--AUTO O
     &R MANUAL.  ALWAYS')
  62 FORMAT('SELECT THE NUMBER OF THE EARLIEST CHANGE YOU WISH TO MAKE
     &AS YOU WILL CAUSE THE')
  63 FORMAT('PROGRAM TO RETURN TO THE POINT COINCIDING WITH THE PARAME
     &TER WHO''S NUMBER')
  64 FORMAT('YOU SELECTED, AND THUS YOU WILL ALSO GET TO INPUT DECISION
     &S FOR ALL PARAMETERS')
  65 FORMAT('WITH LARGER NUMBERS THAN THE ONE YOU SELECTED.'/' '/' '/)
     RETURN
     END
```

V. Subroutine Minval

This subroutine adjusts the input years (XYEAR) to integer numbers starting with 1. This is done so that the X and Y values used in the various regressions will be scaled correctly in relation to each other. The scaling is done by finding the smallest year in the input years (called MINXYR). This number minus one (called IREDUC) is then subtracted from each value in XYEAR and the scaled results are stored in vector XBASE.

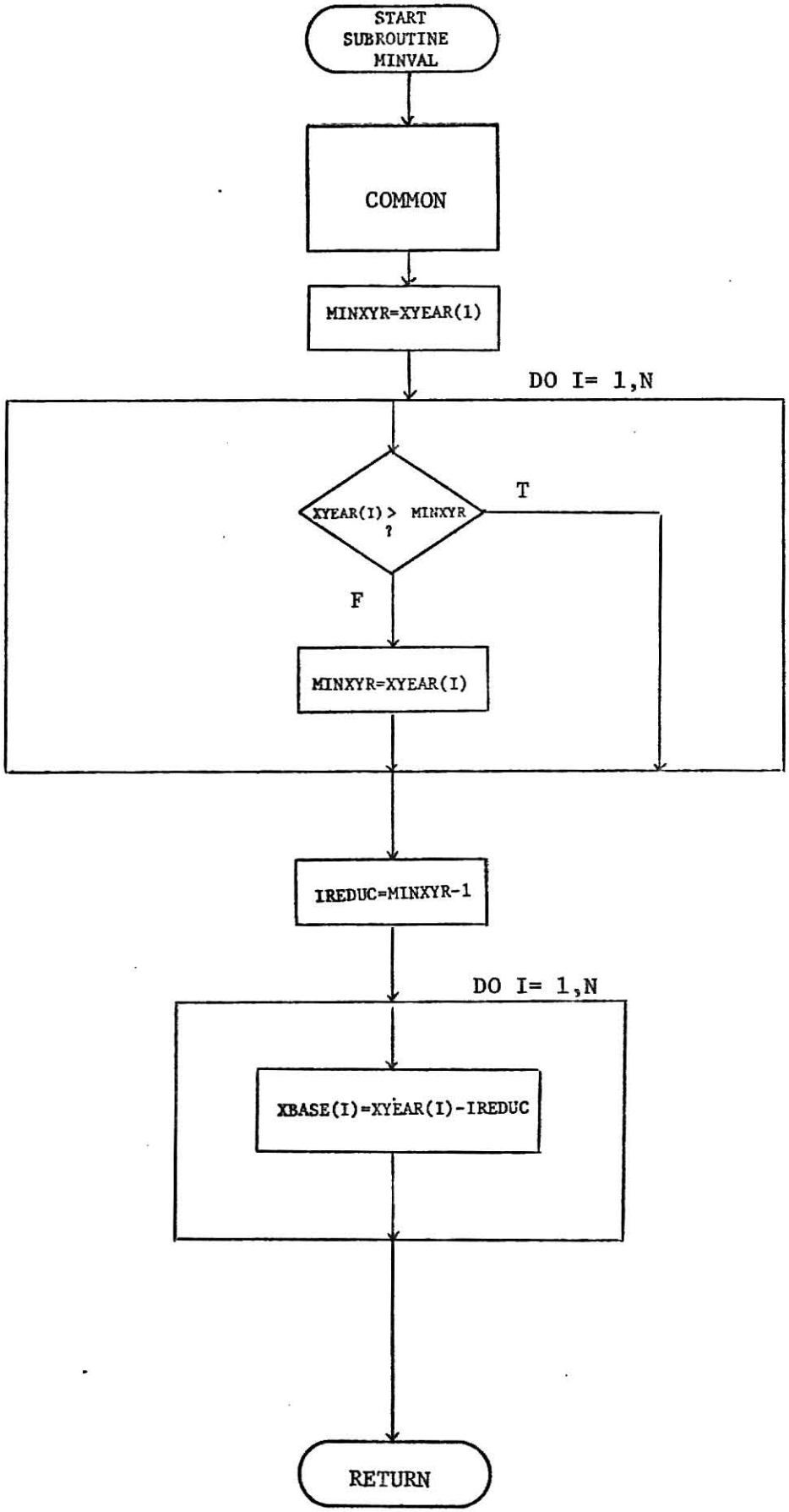
This method of reducing the years to smaller scaled values allows the use of years that are greater than one year more than the previous. For example: 1930, 1937, 1945, 1970 is a perfectly acceptable stream of input years.

A flowchart of subroutine MINVAL follows a listing of it.

Subroutine Minval

```
C*************************************************************************
C                                                                        *
C   THIS SUBROUTINE REPLACES THE YEAR INPUT X'S WITH SMALLER X'S IN      *
C   THE VICINITY OF 1.  THIS IS SO THAT THE EXPONENTIAL AND LOG-         *
C   ARITHMIC PROJECTIONS WILL WORK CORRECTLY.  THE SMALLER VALUES ARE    *
C   OBTAINED BY TAKING THE SMALLEST INPUT X AND SUBTRACTING ITS VALUE    *
C   MINUS ONE FROM ALL INPUT X'S.  THUS, WITH THIS TYPE OF SCALING       *
C   THE INPUT X'S DON'T HAVE TO BE SEQUENTIAL YEARS.                     *
C                                                                        *
C*************************************************************************
      SUBROUTINE MINVAL
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ
      INTEGER XYEAR,PRONAM,PRTCAT
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTCAT,PRONAM,LINRSQ,LOGRSQ
      MINXYR=XYEAR(1)
      DO 6 I=1,N
      IF(XYEAR(I).GT.MINXYR) GO TO 6
      MINXYR=XYEAR(I)
    6 CONTINUE
      IREDUC=MINXYR-1
      DO 7 I=1,N
    7 XBASE(I)=XYEAR(I)-IREDUC
      RETURN
      END
```

Flowchart for Minval



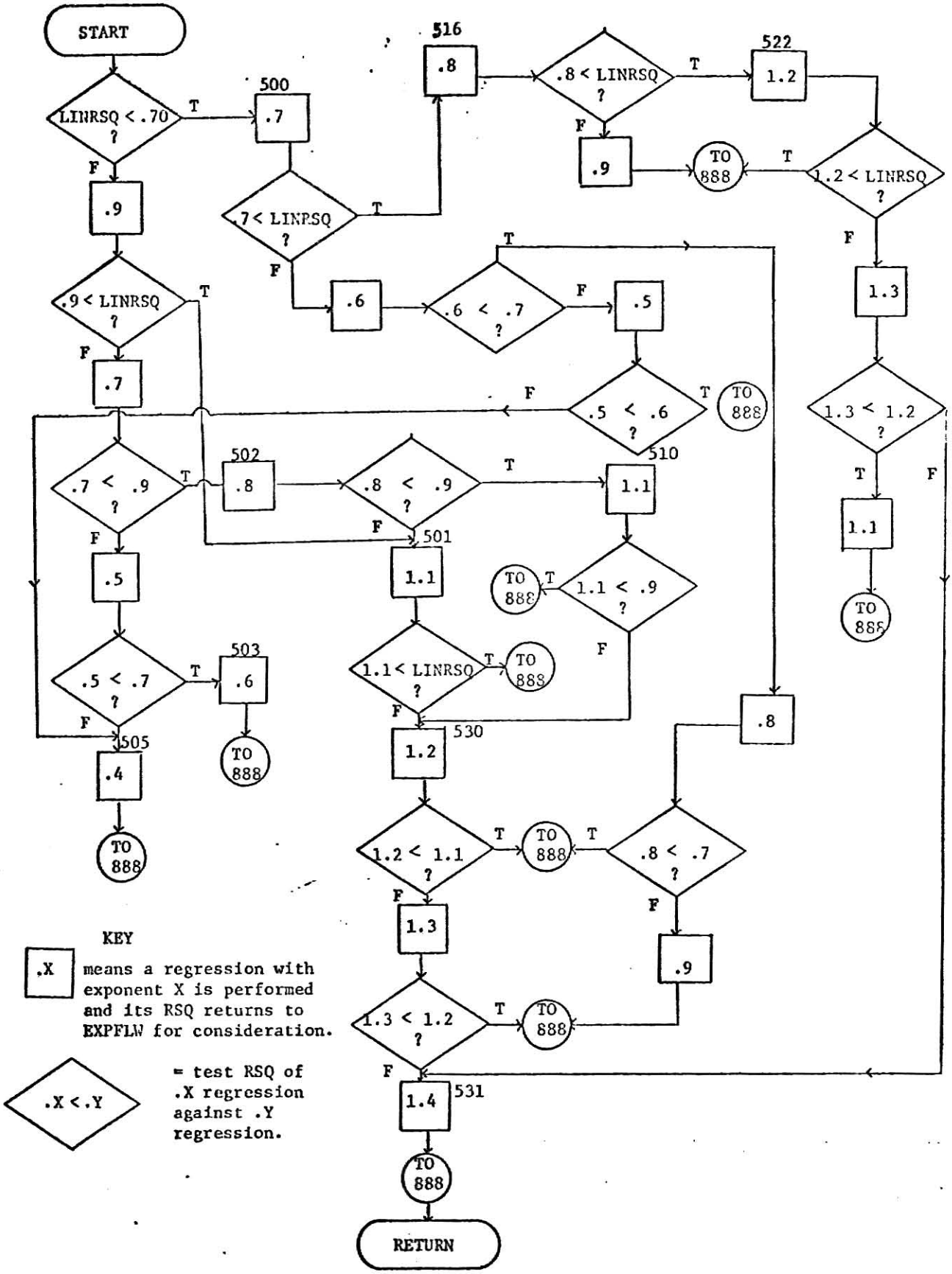
VI. Subroutine Expflw

Expflw is a subroutine that determines the order that different exponential projections with varying exponents are to be performed. Expflw then controls the regression by establishing the exponent and then calling subroutine REGRES to which the exponent is passed.

The order the regressions of various exponents are to be performed in is determined by examining the RSQ of the linear projection or else the previously performed exponential regression's RSQ (such as ERSQ12, etc.). Expflw then uses a built-in decision tree to decide the next regression. The first decision is whether the linear projection was a good fit. If so, then Expflw tries exponents that are close to 1.0 first in order to achieve a higher RSQ. If a linear projection was a bad fit, it is assumed that the trend best fitting the base data may be curvilinear so the exponents tried first are further from 1.0.

Once Expflw has determined its starting point then all the exponents between 0.4 and 1.4 are tried in regressions either implicitly or explicitly. For example, if the linear r^2 (LINRSQ) was .78 and the r^2 of a 0.7 exponent regression is less than .78, then all the exponents below 0.7 can also be eliminated implicitly. Expflw would then try the exponents between 0.7 and 1.0 if they were better than 1.0 (in term of RSQ) it would stop, otherwise the exponents above 1.0 would be tried. The overall decision tree used by Expflw is shown on the following page in a simplified flowchart. A longer flowchart follows a listing of the subroutine.

SIMPLE FLOWCHART FOR EXPFLW

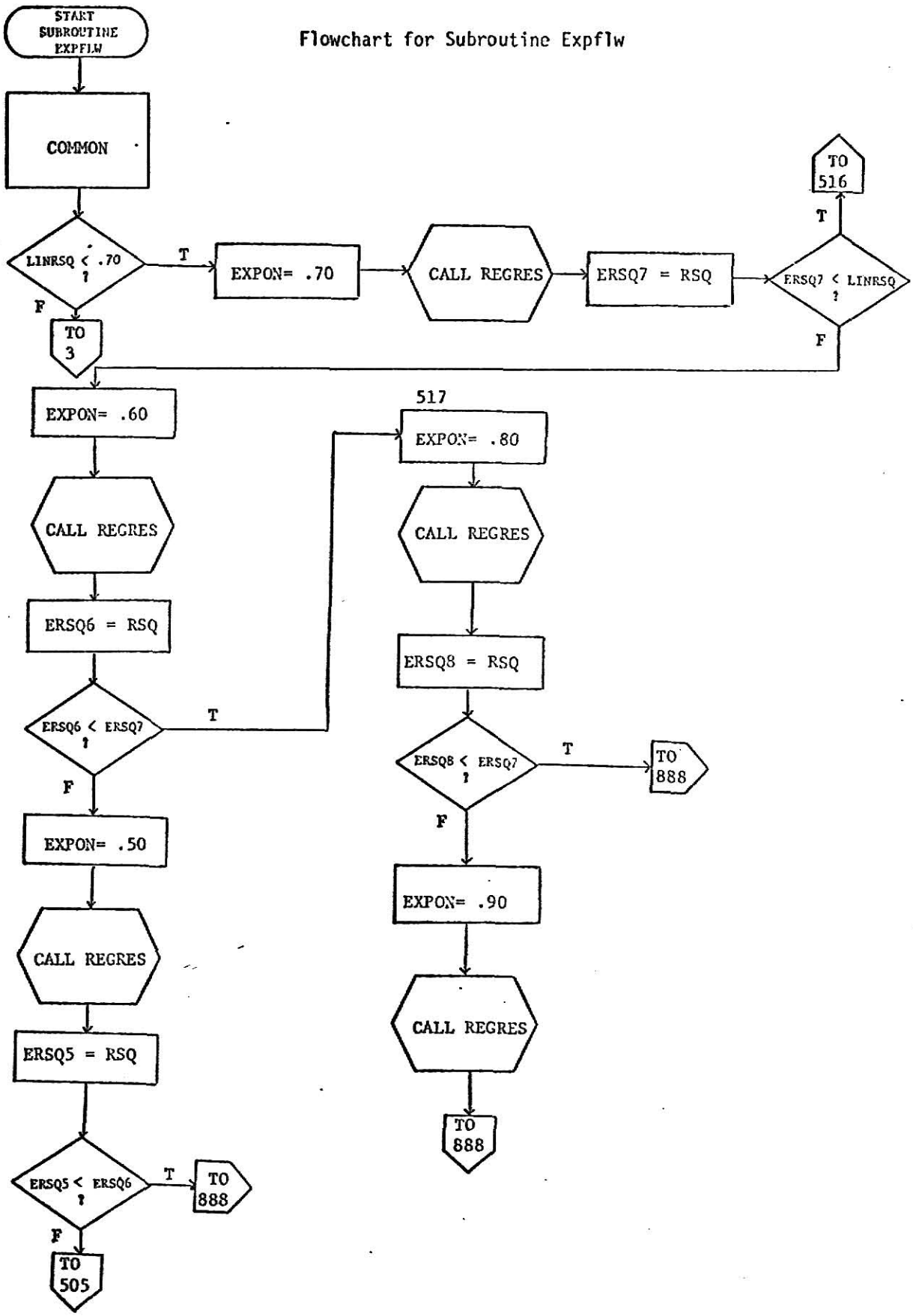


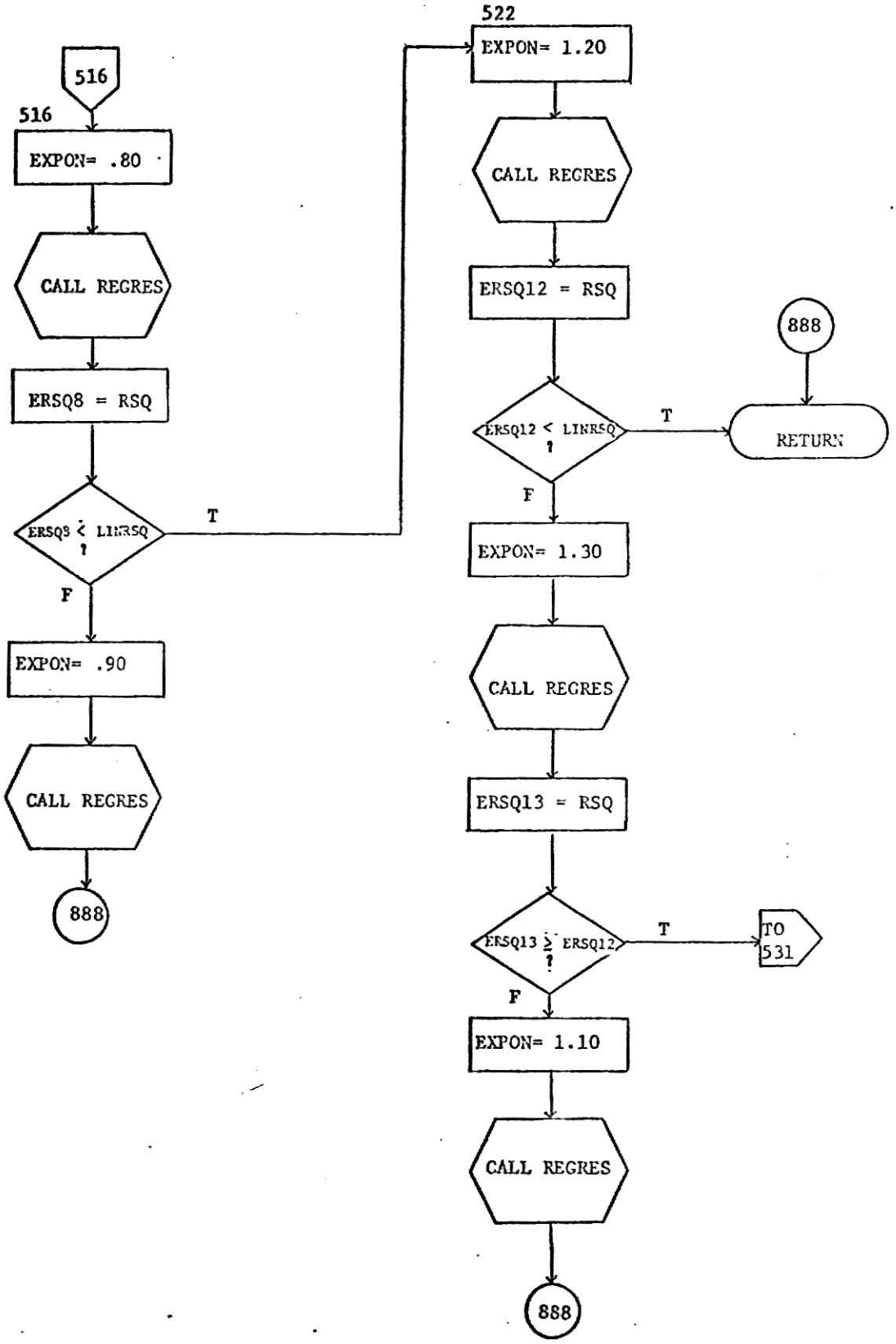
Subroutine Expflw Listing

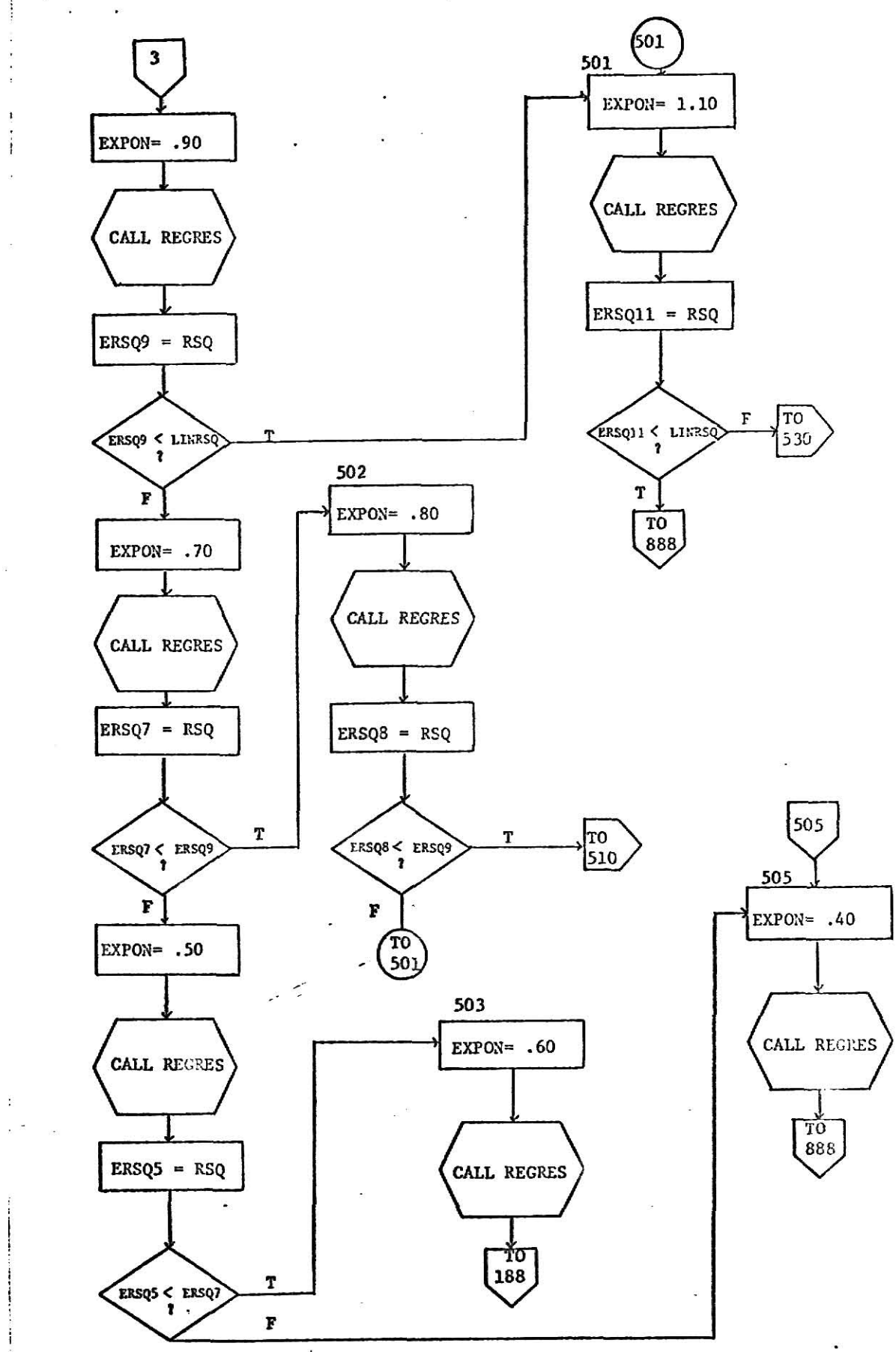
```
C*************************************************************************
C   THIS SUBROUTINE CONTAINS THE LOGIC FOR PICKING THE BEST EXPONENTIAL  *
C   PROJECTION.  THIS IS DONE BY PICKING THE NEXT PROJECTION TO TRY      *
C   BASED ON THE RSQ OF THE LAST PROJECTION JUST MADE.  THIS SUB-        *
C   ROUTINE ASSUMES THAT IF THE LINEAR RSQ (LINRSQ) WAS NOT ABOVE 0.7    *
C   THAT THAT THE TREND WOULD COME NEARER TO AN EXPONENTIAL PROJECTION   *
C   THAT WAS FURTHER AWAY FROM 1.0 AND THUS THE SUBROUTINE LOOKS FIRST   *
C   AT EXPONENTIAL PROJECTIONS WHERE THE EXPONENT IS FURTHER AWAY FROM   *
C   1.0.  OTHERWISE (WHEN THE RSQ OF THE LINEAR PROJECTION IS ABOVE      *
C   .70) THE EXPONENTIAL PROJECTIONS WITH AN EXPONENT NEAR 1.0 ARE       *
C   TRIED FIRST IN AN EFFORT TO SAVE THE NUMBER OF PROJECTIONS NEC-      *
C   CESSARY.                                                             *
C*************************************************************************
C                          SYMBOL DEFINITION                            *
C                                                                       *
C   LINRSQ = R SQUARE OF THE LINEAR REGRESSION.                         *
C                                                                       *
C   REGRES = SUBROUTINE WHICH PERFORMS THE FOLLOWING TYPES OF REGRES-   *
C            SION.  IREGTP IS THE KEY TO THE TYPE OF REGRESSION RE-     *
C            REGRESSION REGRES IS TO PERFORM.                           *
C            LINEAR       WHEN IREGTP = 1.0                             *
C            LOGARITHMIC               = 2.0                           *
C            EXPONENTIAL               = 3.0,                          *
C                                                                       *
C   EXPON  = VALUE PASSED IN COMMON TO SUBROUTINE REGRES TO DETER-      *
C            MINE THE EXPONENT USED IN EXPONENTIAL REGRESSION            *
C                                                                       *
C   ERSQX  = RSQ FOR AN EXPONENTIAL PROJECTION USING THE EXPONENT X     *
C            EX:  ERSQ12 MEANS THE EXPONENT 1.2.                        *
C                                                                       *
C*************************************************************************
      SUBROUTINE EXPFLW
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER XYEAR,PRTDAT,PRGNAM,XSYEAR
      REAL*4 BESPRO,DCPMAT,LINRSQ,LOGRSQ
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DCPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRGNAM,LINRSQ,LOGRSQ
      IF(LINRSQ.LT.0.70) GO TO 500
      EXPON=.90000000000
      CALL REGRES
      ERSQ9=RSQ
      IF(ERSQ9.LT.LINRSQ) GO TO 501
      EXPON=.70000000
      CALL REGRES
      ERSQ7=RSQ
      IF(ERSQ7.LT.ERSQ9) GO TO 502
      EXPON=.50000000
      CALL REGRES
      ERSQ5=RSQ
      IF(ERSQ5.LT.ERSQ7) GO TO 503
  505 EXPON=.40000000
      CALL REGRES
      GO TO 888
  503 EXPON=.60000000
      CALL REGRES
      GO TO 888
  510 EXPON=1.10000000
      CALL REGRES
      ERSQ11=RSQ
```

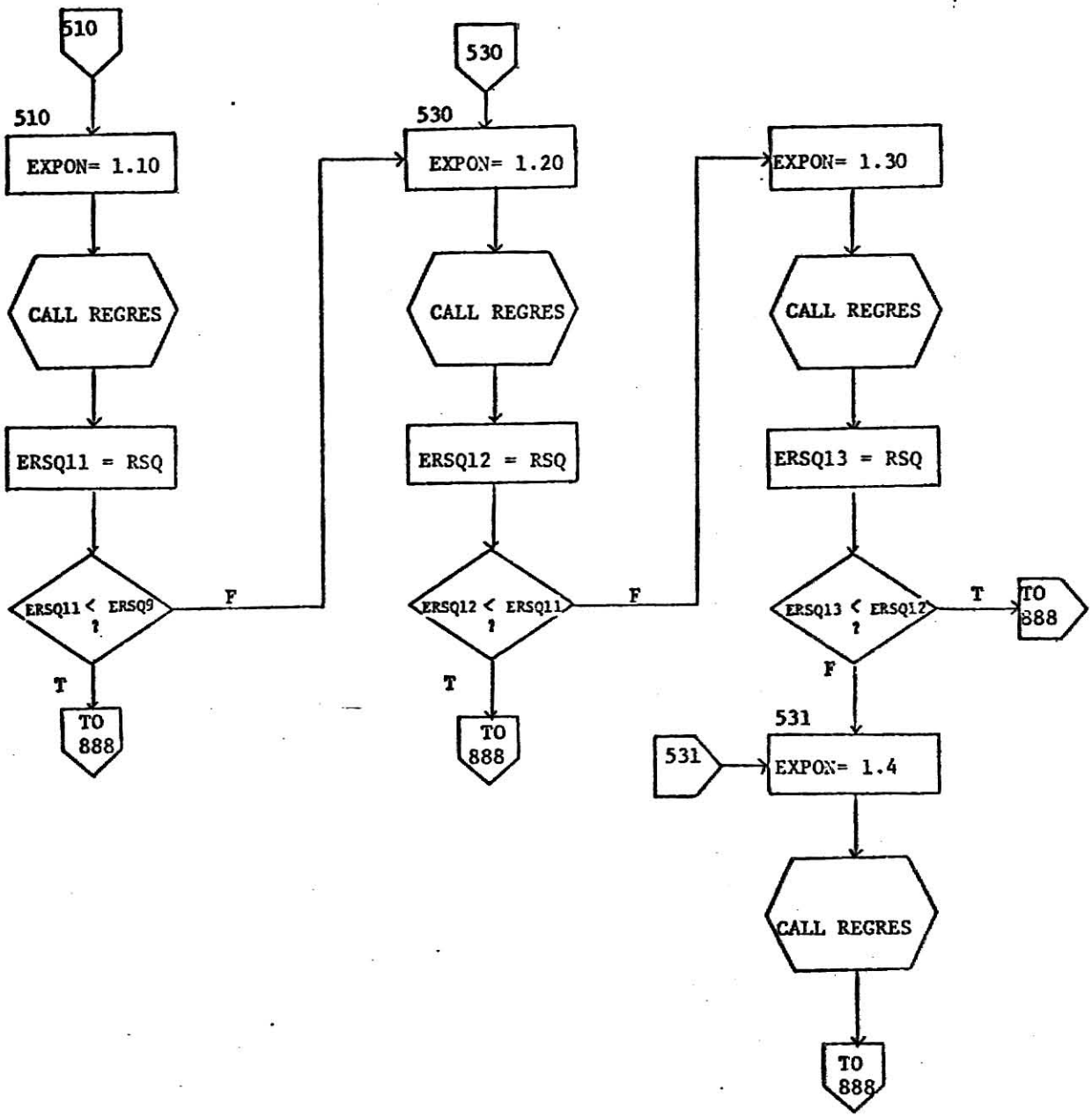
```
      IF(ERSQ11.LT.ERSQ9) GO TO 888
      GO TO 530
  502 EXPON=.800000000
      CALL REGRES
      ERSQ8=RSQ
      IF(ERSQ8.LT.ERSQ9) GO TO 510
  501 EXPON=1.100000000
      CALL REGRES
      ERSQ11=RSQ
      IF(ERSQ11.LT.LINRSQ) GO TO 888
  530 EXPON=1.200000000
      CALL REGRES
      ERSQ12=RSQ
      IF(ERSQ12.LT.ERSQ11) GO TO 888
      EXPON=1.300000000
      CALL REGRES
      ERSQ13=RSQ
      IF(EPSQ13.LT.ERSQ12) GO TO 888
  531 EXPON=1.400000000
      CALL REGRES
      GO TO 888
  500 EXPON=.700000000
      CALL REGRES
      ERSQ7=RSQ
      IF(ERSQ7.LT.LINRSQ) GO TO 516
      EXPON=.600000000
      CALL REGRES
      ERSQ6=RSQ
      IF(ERSQ6.LT.ERSQ7) GO TO 517
      EXPON=.500000000
      CALL REGRES
      ERSQ5=RSQ
      IF(ERSQ5.LT.ERSQ6) GO TO 888
      GO TO 505
  517 EXPON=.800000000
      CALL REGRES
      ERSQ8=RSQ
      IF(ERSQ8.LT.EPSQ7) GO TO 888
      EXPON=.900000000
      CALL REGRES
      GO TO 888
  516 EXPON=.800000000
      CALL REGRES
      ERSQ8=RSQ
      IF(ERSQ8.LT.LINRSQ) GO TO 522
      EXPON=.900000000
      CALL REGRES
      GO TO 888
  522 EXPON=1.200000000
      CALL REGRES
      ERSQ12=RSQ
      IF(ERSQ12.LT.LINPSQ) GO TO 888
      EXPON=1.300000000
      CALL REGRES
      ERSQ13=RSQ
      IF(ERSQ13.GE.ERSQ12) GO TO 531
      EXPON=1.100000000
      CALL REGRES
  888 RETURN
      END
```

Flowchart for Subroutine Expflw




516
516
EXPON= .80
CALL REGRES
ERSQ8 = RSQ
ERSQ8 < LINRSQ ?
T
F
EXPON= .90
CALL REGRES
888
522
EXPON= 1.20
CALL REGRES
ERSQ12 = RSQ
ERSQ12 < LINRSQ ?
T
F
888
RETURN
EXPON= 1.30
CALL REGRES
ERSQ13 = RSQ
ERSQ13 ≥ ERSQ12 ?
T
TO 531
F
EXPON= 1.10
CALL REGRES
888


3
EXPON= .90
CALL REGRES
ERSQ9 = RSQ
ERSQ9 < LINRSQ ?
T
F
501
501
EXPON= 1.10
CALL REGRES
ERSQ11 = RSQ
ERSQ11 < LINRSQ ?
F
TO 530
T
TO 888
EXPON= .70
CALL REGRES
ERSQ7 = RSQ
ERSQ7 < ERSQ9 ?
T
F
502
EXPON= .80
CALL REGRES
ERSQ8 = RSQ
ERSQ8 < ERSQ9 ?
T
TO 510
F
TO 501
EXPON= .50
CALL REGRES
ERSQ5 = RSQ
ERSQ5 < ERSQ7 ?
T
F
503
EXPON= .60
CALL REGRES
TO 188
505
505
EXPON= .40
CALL REGRES
TO 888


510
510
EXPON= 1.10
CALL REGRES
ERSQ11 = RSQ
ERSQ11 < ERSQ9 ?
F
T
TO 888
530
530
EXPON= 1.20
CALL REGRES
ERSQ12 = RSQ
ERSQ12 < ERSQ11 ?
F
T
TO 888
EXPON= 1.30
CALL REGRES
ERSQ13 = RSQ
ERSQ13 < ERSQ12 ?
T
TO 888
F
531
531
EXPON= 1.4
CALL REGRES
TO 888

VII. Subroutine Prosto

This subroutine stores the projections and holds them until time to print the best three. The first three projections and their associated statistics (RSQ,XBAR, etc.) are automatically stored in order by RSQ with the projection with the highest RSQ going in the 1st column of matrix BESPRO and so on. (The statistics going in matrix DOPMAT.) Since the output matrix and the statistic matrix are initialized to zero before each data set is input, if less than three projections are run (due to the goodness of fit of the first one or two projections) then zeros will be printed for each projection (up to two) not run, when the best three projections are printed.

After the first three projections are stored, if subroutine EXPFLW requests more projections, then as each projection is made, its r^2 is tested against the r^2 of the three projections already stored in BESPRO. (This also was done as each of the first 3 projections were stored.) If the RSQ of the latest projection (YPROJ) is less than the RSQ of the projection already stored in column 3 of BESPRO (the third best projection, as measured by r^2, so far made) then nothing happens and the subroutine returns.

If the RSQ of the latest projection is better than the r^2 of the projection in the third column of BESPRO, the last projection's RSQ is tested against the 2nd column's associated RSQ. If found better, it is tested against the 1st column projection's r^2.

From the above, the subroutine decides if the r^2 of the latest projection is between the 1st and 2nd or 2nd and 3rd best projections' r^2's. If it fits between the 1st and 2nd projection, then the 2nd column of BESPRO and DOPMAT is put into the 3rd column displacing what formerly was the 3rd best projection. Then the latest projection is stored in the 2nd

column of BESPRO and its associated statistics are put into the second column of DOPMAT.

If the newest projections RSQ is better than the formerly best projection's RSQ (in the 1st column of BESPRO) then the newest projection is put there with the existing 1st and 2nd best projection being shuffled one column to the right--becoming the 2nd and 3rd best projections and displacing what was formerly the third best projection.

If the newest projection is only a better projection than the 3rd best projection, the newest projection replaces it directly and no shuffling of the other existing projections occurs. In this case and the others above, when data in a column of BESPRO is shifted, the data in the same column number in DOPMAT is also shifted by an equal amount.

The above processes occur each time a projection is made. Thus, after the last projection is made, BESPRO contains the best three projections from among all the projections made for a given data set.

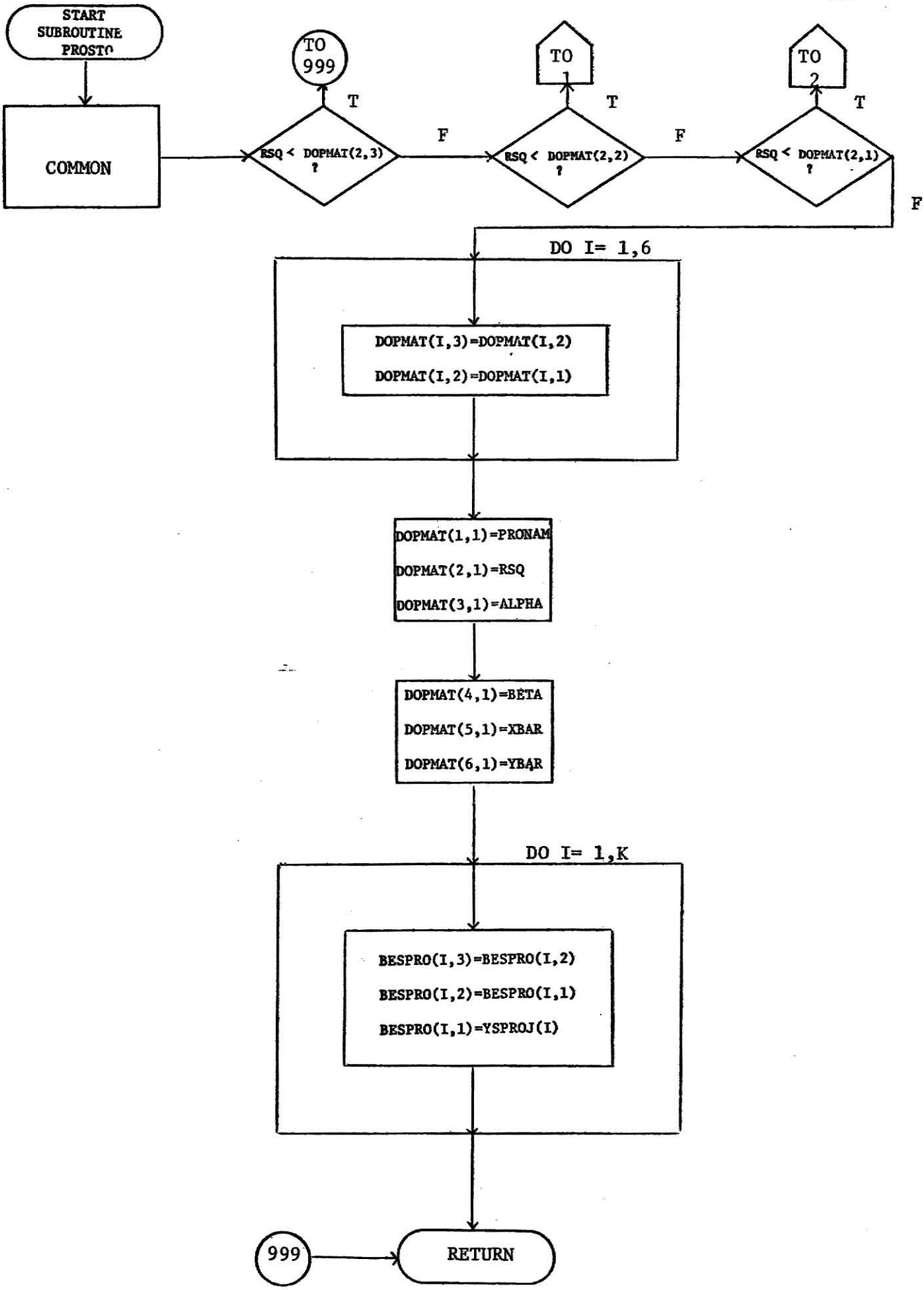
A flowchart of subroutine PROSTO follows a listing of it.

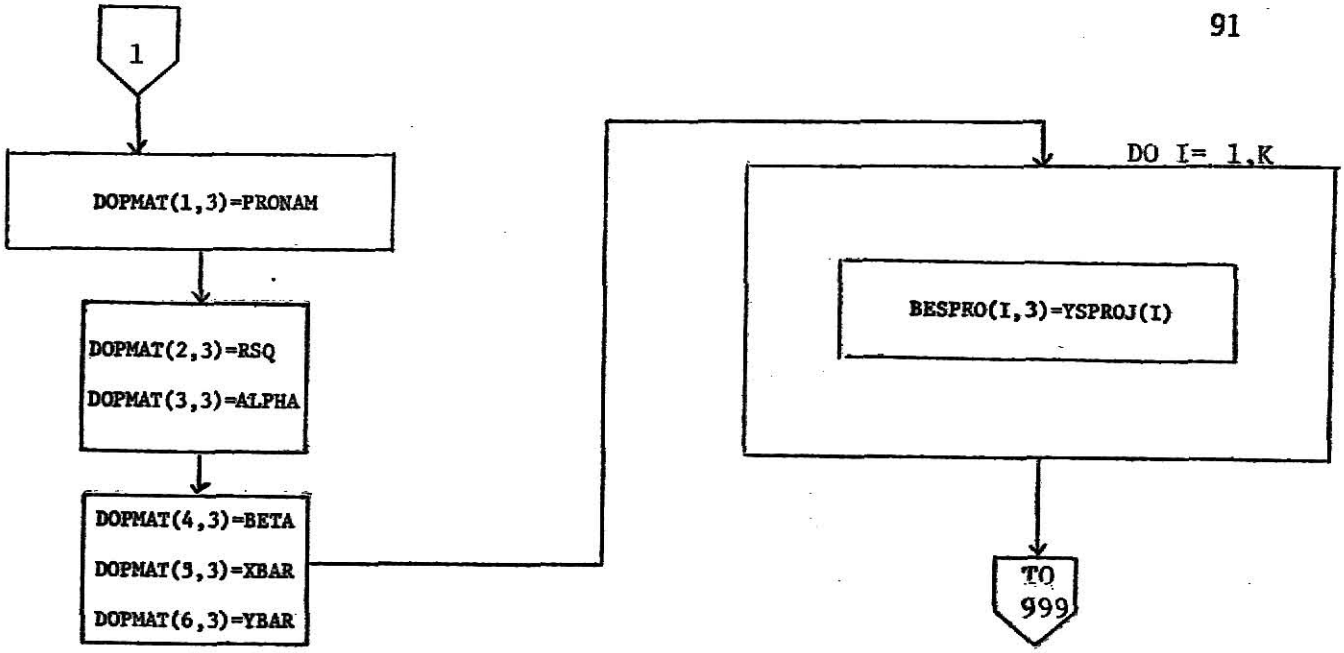
Subroutine Prosto Listing

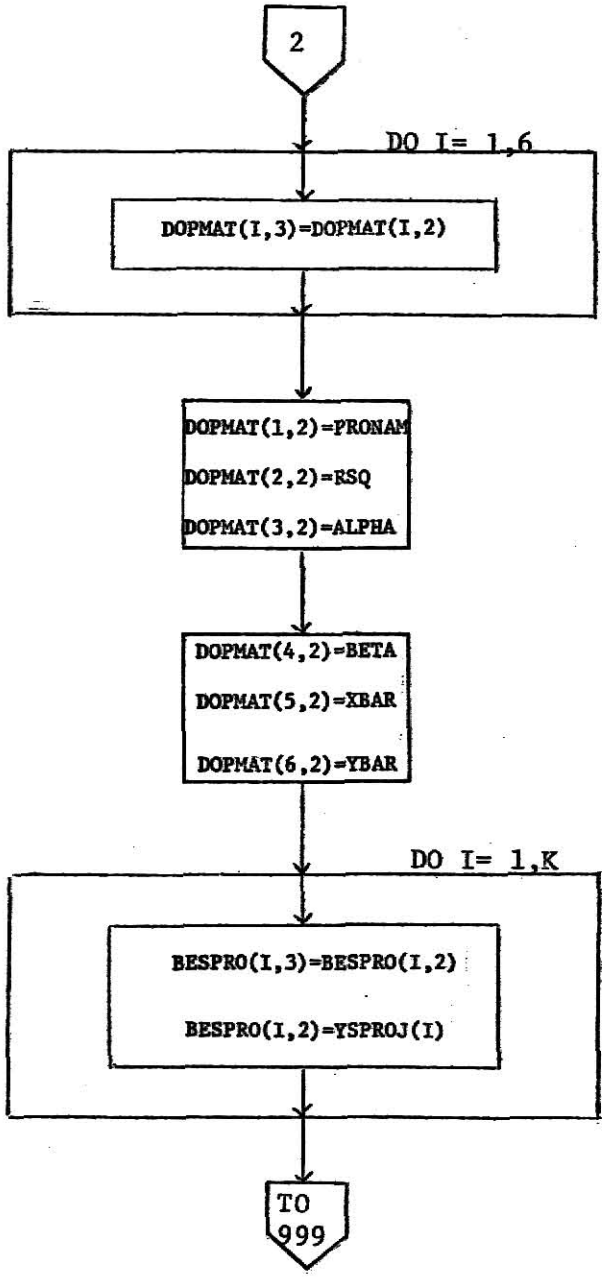
```
C*****************************************************************************
C                                                                              *
C   THIS SUBROUTINE KEEPS THE BEST THREE PROJECTIONS AND THEIR                 *
C   ASSOCIATED DOPE VECTORS (CONTAINING RSQ, ALPHA, BETA, YBAR, XBAR,          *
C   AND PRONAM.)  THEY ARE STORED IN MATRIX DOPMAT (FOR THE STATISTICS)        *
C   AND IN MATRIX BESPRO FOR THE BEST THREE PROJECTIONS.  THE                  *
C   PROJECTIONS ARE FILED IN THE MATRIX IN ORDER OF THEIR RSQ, WITH            *
C   THE PROJECTION WITH THE HIGHEST RSQ GOING INTO THE FIRST COLUMN            *
C   OF THE MATRIX AND SO ON.  NEW PROJECTIONS ARE COMPARED TO THOSE            *
C   ON FILE IN THIS SUBROUTINE AND IF THEIR RSQ EXCEEDS THAT OF THE            *
C   LOWEST RSQ-PROJECTION ON FILE, THE NEW PROJECTION IS STORED IN             *
C   THE COLUMN BEFITTING ITS RSQ RANK AND ONE OF THE OTHER PROJEC-             *
C   TIONS IS DROPPED AND THE OTHER PROJECTIONS AND THEIR ASSOCIATED            *
C   DOPE VECTORS ARE MOVED OVER TO MAKE ROOM FOR THE NEW PROJECTION            *
C   AND ITS DOPE VECTOR, WHERE IT, AS DICTATED BY ITS RSQ RANK,                *
C   SHOULD GO.                                                                 *
C   **THUS THE PRINTING OF BESPRO WOULD GIVE THE BEST THREE PROJEC-            *
C   TIONS FOUND FOR A GIVEN SET OF INPUT DATA.                                 *
C                                                                              *
C   PRONAM = NAME OF THE TYPE OF PROJECTION JUST MADE BY REGRES.               *
C            PRONAM IS REPRESENTED BY A NUMBER, THE KEY TO WHICH IS:           *
C                                                                              *
C   -------KEY------------------PRONAM--------------------KEY--------         *
C                                                                              *
C               99 = LINEAR                                                    *
C               98 = LOGARITHMIC                                               *
C               OTHERWISE :  10 *EXPONENT                                      *
C                            EX: 1.4->PRONAM=14                                *
C                                                                              *
C*****************************************************************************
      SUBROUTINE PROSTO
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 BESPRO,DOPMAT,LINRSQ,LOGRSQ
      INTEGER PRTDAT,XYEAR,XSYEAR,PRONAM
      COMMON BESPRO(100,3),YSPROJ(100),XSBASE(100),YBASE(100)
      COMMON XBASE(100),DOPMAT(6,3),EXPON,RSQ,ALPHA,BETA,YBAR,XBAR
      COMMON XYEAR(100),N,K,M,IREGTP,PRTDAT,PRONAM,LINRSQ,LOGRSQ
      IF(RSQ.LE.DOPMAT(2,3)) GO TO 999
      IF(RSQ.LE.DOPMAT(2,2)) GO TO 1
      IF(RSQ.LE.DOPMAT(2,1)) GO TO 2
      DO 3 I=1,6
      DOPMAT(I,3)=DOPMAT(I,2)
    3 DOPMAT(I,2)=DOPMAT(I,1)
      DOPMAT(1,1)=PRONAM
      DOPMAT(2,1)=RSQ
      DOPMAT(3,1)=ALPHA
      DOPMAT(4,1)=BETA
      DOPMAT(5,1)=XBAR
      DOPMAT(6,1)=YBAR
      DO 4 I=1,K
      BESPRO(I,3)=BESPRO(I,2)
      BESPRO(I,2)=BESPRO(I,1)
    4 BESPRO(I,1)=YSPROJ(I)
      GO TO 999
    2 DO 5 I=1,6
    5 DOPMAT(I,3)=DOPMAT(I,2)
      DOPMAT(1,2)=PRONAM
      DOPMAT(2,2)=RSQ
      DOPMAT(3,2)=ALPHA
      DOPMAT(4,2)=BETA
```

```
      DOPMAT(5,2)=XBAR
      DOPMAT(6,2)=YBAR
      DO 6 I=1,K
      BESPRO(I,3)=BESPRO(I,2)
    6 BESPRC(I,2)=YSPRCJ(I)
      GO TO 999
    1 DOPMAT(1,3)=PRONAM
      DOPMAT(2,3)=RSQ
      DOPMAT(3,3)=ALPHA
      DOPMAT(4,3)=BETA
      DOPMAT(5,3)=XBAR
      DOPMAT(6,3)=YBAR
      DO 7 I=1,K
    7 BESPRO(I,3)=YSPRCJ(I)
  999 RETURN
      END
```

Flowchart for Subroutine Prosto




1
DOPMAT(1,3)=PRONAM
DOPMAT(2,3)=RSQ
DOPMAT(3,3)=ALPHA
DOPMAT(4,3)=BETA
DOPMAT(5,3)=XBAR
DOPMAT(6,3)=YBAR
DO I= 1,K
BESPRO(I,3)=YSPROJ(I)
TO 999


2
DO I= 1,6
DOPMAT(I,3)=DOPMAT(I,2)
DOPMAT(1,2)=PRONAM
DOPMAT(2,2)=RSQ
DOPMAT(3,2)=ALPHA
DOPMAT(4,2)=BETA
DOPMAT(5,2)=XBAR
DOPMAT(6,2)=YBAR
DO I= 1,K
BESPRO(I,3)=BESPRO(I,2)
BESPRO(I,2)=YSPROJ(I)
TO 999

IV. AUTOREG MULTI USER'S GUIDE

I. Program Features

Autoreg Multi is designed to project county agricultural data and provide a sum of the counties under the title "DISTRICT." Obviously, if the data for each county in a crop reporting district were input then the numbers in the column entitled "DISTRICT" would serve as a projection for that crop reporting district.

Autoreg Multi contains the following features, some of which are discussed in section II:

1) The program automatically centers titles.
2) If a title or footnote is only one card in length instead of two, no blank second card must be input.
3) The names of the counties are automatically truncated to eight characters and they are centered above their associated data in the output table.
4) The input data format is entirely compatable with that used by the Kansas State University Master Projection program.
5) The best projection models to try are selected by an internal decision tree, thus one run guarantees the best projection from among the various models available within the program.
6) Each page of output has a title block that also includes the district number.
7) All paging of output is determined automatically.
8) Three projection sets are generated containing the best, 2nd best, and 3rd best projections based on r^2.. Thus, if other constraints preclude the use of the first projection set, two others are available to select from.
9) At the beginning of each district's county projections, a statistical summary is provided.

10) Projections less than zero are automatically set to zero as negative crop production, yield etc. is not feasible.

11) Scaling of the input years allows correct regressions although years of data are missing. If only some counties are missing data for a given year (as in the sample input data in section II) then an option, when turned on, will calculate the values of the missing data prior to performing the projections.

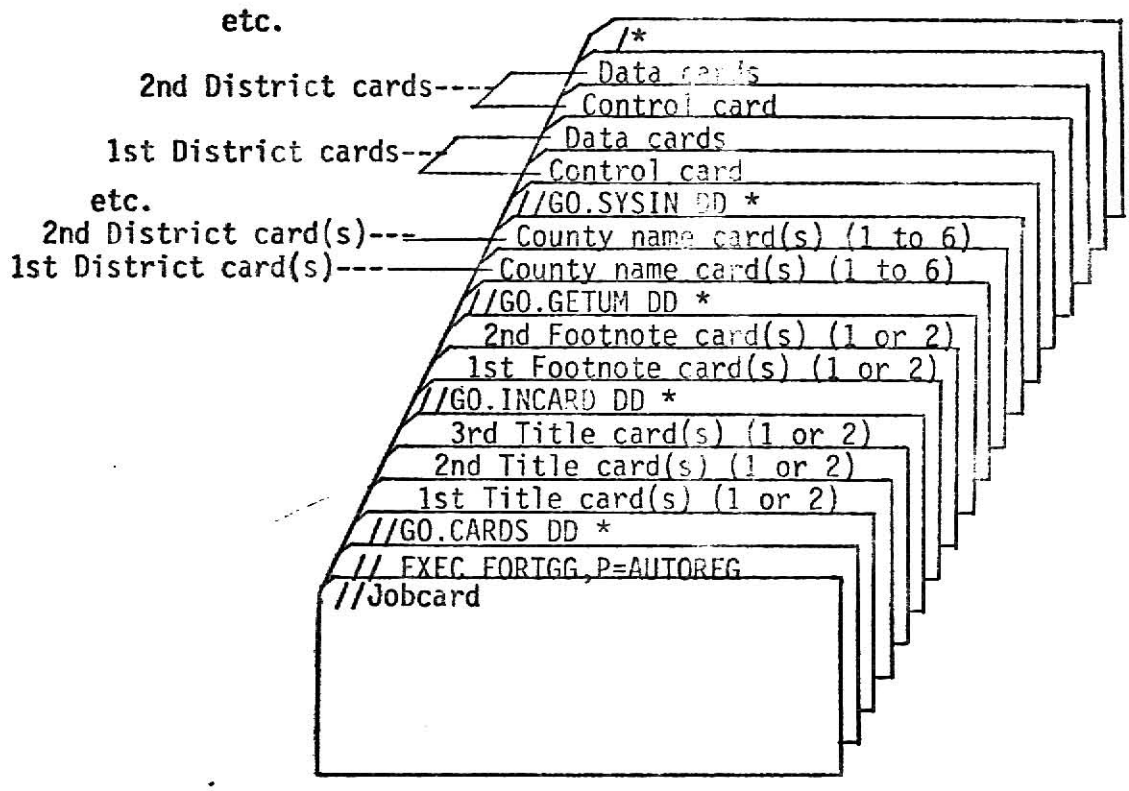
12) Due to the program's taking one county's base data through all the projection models and then starting on the next, rather than all at the same time, some of the projections within a projection set can be going up while others are going down. This allows for more realism, as in the real world, part of the counties in a crop production district may be increasing a given crop's production while others are decreasing it.

13) Multiple districts can be projected in one run. (See sample deck setup in section II.)

II. Program Use

In order to use Autoreg Multi, the user must supply the following items: titles, footnotes, county names, control cards, and data cards. The order these must be input in is shown in the diagram below.

Sample Deck Setup



NOTE: Include county name cards, control card, and data cards once for each district in the run. All other cards should be included only once per run.

The actual appearance of the preceding cards is shown below on a sample input listing. Exactly where the titles, footnotes, and county names are printed can be seen in the sample output listing later in this section. The output listing is the output resulting from the input cards shown below.

SAMPLE INPUT LISTING

```
// EXEC FORTGG,P=AUTOREG
//STEPLIB DD DSN=OSKNG.STATIST.ANALYSIS,DISP=SHR
//GO.CARDS DD *
+TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO
 1975 AND PROJECTION THROUGH 2035
YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)
KANSAS
//GO.INCARD DD *
KANSAS CROP AND LIVESTOCK REPORTING SERVICE,MIMEO. REPORTS, 1962 - 1976.
+TEST RUN MADE BY WALTER TURNER.  APRIL 13, 1977.  TEST SHOWS THE REPLACEMENT OF
 MISSING DATA.
//GO.GETU4 DD *
2BARBER,COMANCHE,EDWARDS,HARPER,HARVEY,KINGMAN,KIOWA,PAWNEE,PRATT,RENO,SEDGWICK,
STAFFORD,SUMNER,
//GO.SYSIN DD *
  15  60  13    0 YES 8
1961        300   220   280   300   320   250   280   260   270   290   310   280
1961        340
1962        200   175   220   230   170   190   220   240   210   230   220   220
1962        220
1963        150   120   150   190   300   190   150   160   150   240   240   170
1963        230
1964        210   100   180   260   360   240   150   170   130   270   310   190
1964        290
1965        290   210   250   300   340   290   250   190   290   330
1965        320
1966        240   150   160   240   250   220   170   150   130   220   250   200
1966        250
1967        160   140   180   130   130   150   130   180   220   200   140   220
1967        180
1968        283   135   174   334   382   330   162   164   212   345   375   292
1968        304
1969        332   293   317   321   306   321   316   290   299   321   335   329
1969        351
1970        313   233   306   320   369   311   307   321   310   326   352   327
1970        354
1971        269   232   275   284   382   311   283   362   312   357   363   362
1971        368
1972        292   253   355   312   387   348   326   326   344   363   364   357
1972        348
1973        376   253   348   403   366   329   339   368   343   344   379   377
1973        409
1974        251   223   226   278   238   261   224   250   227   254   273   256
1974        291
1975        274   208   316         274         224   306   280   278   266   327
1975        279
/*
```

The following sections explain the input card requirements in greater detail.

A. Title Cards

The title cards should be input behind the system card: //GO.CARDS DD *. The title cards should be included only once for each run. If you wish a district to have different titles (or footnotes), you must run it separately. (For an example of the input title cards, see the sample input card listing on p. 4 .)

This program contains provisions for the reading and printing of three titles, each of which may be either one or two cards in length up to a maximum of 132 characters. (For an example of where the titles are printing see the sample program output starting on p. 11.) If the title is 80 characters in length or less, insert only one card. If the title exceeds 80 characters in length, continue it on a second card. If you do continue it on a second card, any characters past column 53 will be ignored. When a title is continued on a second card you must put a '+' in the first column of the first card. This signals the computer that the title is two cards long. (This also applies to footnotes.) Omit the '+' if the title is only one card in length.

The title cards are included only once for each run. The title cards should be input in the order you desire the titles to appear in an output. The titles will be automatically centered by the program. If you wish to omit the printing of a title, insert one blank card in the input deck where the title card should go.

B. Footnote Cards

The footnote cards should be put behind the following system card:

//GO.INCARD DD *

One or two cards must be input for each of the two allowable footnotes. If you wish for a run to include only one footnote, insert a blank card for the second footnote. If you wish for a run to contain no footnotes, include two blank footnote cards.

A footnote can be either one or two cards in length, up to a maximum of 124 characters. If the footnote is 80 characters in length or less, insert only one card. If the footnote exceeds 80 characters, continue it onto a second card. If you do continue it on a second card, any characters past column 45 will be ignored. When a footnote is continued on a second card you must put a '+' in the first column of the first card. This signals the computer that the title is two cards long. (This also applies to titles.) Omit the '+' if the footnote is only one card in length. The footnote cards are included only once for each time the program is run. (An example of footnote cards can be seen in the sample input listing on p. 4 .)

C. County Name Cards

The county name card (s) should be input behind the system card:

//GO.GETUM DD *

A set of county name card (s) should be included for each district. The first column of the first card of each set of county cards should contain the number of cards in that set of county cards. The number of cards can be from one up to and including six.

The names of the counties can be any length just so they all are contained within six cards. The county names may contain up to two imbedded blanks. Each county name should be separated from the next by a comma, leaving no spaces

between names. The last county name must be followed by a comma. (For those of you used to using the Master Projection program, in this program it is not necessary to include the word DISTRICT with the county names.)

D. Control Card

The control card for the 1st district should be input immediately behind the system card: //GO.SYSIN DD * . The control card for any other districts should be input immediately following the data for the preceding district. The format for the control card (s) is (2X,I2,2X,I2,2X,I2,2X,I2,1X,A3,A3). (See sample input listing on p. 4 for an example of a control card. The variables read in from the control card are, in order: N, K, L, IGAP, IZERO, IDISTR. Their functions are as follows:

N = the number of base years of data for this district. The maximum value for N is 40. N should be equal to the number of different years of data cards.

K = the number of years you wish the program to make projections for this district, up to a maximum of 60.

L = the number of counties for this district. The number of columns of data, excluding the year, on the data cards.

IGAP = This controls the minimum difference in RSQ a projection must have to replace another in the set of best projections. For example, if you wish a projection's RSQ to exceed that of the next best projection by at least .05 you would set IGAP to 5. Unless the first three projections made are the best three, then all projections would be separated by at least .05. If no value is entered for IGAP it defaults to 0.

IZERO = This variable controls the feature that replaces missing data. This feature is normally turned off, but the insertion of YES for IZERO will turn it on. When IZERO is turned on, it searches the input data for missing data or zeros which would make the regression routine work incorrectly. If missing data or zeros are found, they are replaced with another value. (Thus, if the

input data should contain zeros, leave IZERO blank.) If the missing data is in the first year, it is replaced with a value that is as far from the value for the second year as the second year is from the value for the third year. For example, in the following cards: 1974 0

1975 57

1976 63

the value inserted for 1974 would be 51. If the value that would be inserted would be less than zero, it is replaced by a zero as negative production is not realistic.

If the missing data is in the last year, it is replaced with a value that is equal to the difference between the year preceding the last and the value for the year two years prior to the last, added to the value of the year prior to the last. For example, in the following cards:

1973 16

1974 18

1975 16

1976 0

1976 would be replaced with the value 14 calculated by $(t_{-1}-t_{-2})+t_{-1} = -2+16=14$.

If the missing data is in any year except the first or last, then it is replaced with the average of the values for the preceding and the following year. Because of this method of replacement and the above ones, the program won't replace data correctly if two consecutive years data for the same county are missing or if the 1st and 3rd or last and 2nd to last years are both missing data for the same county. In that case, the user must calculate one of the missing data items.

IDISTR = This is a three character identifier that contains the number of the control card's district. This number is printed on each page of output so that all of a district's output can be identified, even if the output is separated.

E. Data Cards

The data cards contain the information for each county for each base year. If the number of counties is greater than twelve, two cards must be included for each year. If the number of counties is greater than twenty-five, three cards must be included for each year. The year should be put on all cards for a given year for ease in identification. When more than one card is necessary for a given year, the cards should be put one after each other.

The data should be put on the data cards in the format (I4,4X,12F6.0). The year should be put in the first four columns of each card, followed by four blank columns, and then twelve fields of six columns each. (See the sample input listing on p. 4 for an example of this.) The numbers in the F fields should be right justified unless a decimal is punched. The data for the counties can be put on the cards in any order, but it must be consistent for each year. For example, if the first F field for the first card of data for the year 1954 contains the data for Trego county, then so should the first F field for the first card of data for the year 1955 and so on.

The data cards should be included once for each district with the data cards containing the base year information for each county in a given district. The data cards for a district should immediately follow the control card for that district.

III. Sample Program Output

This section contains a listing of output from a run made with the sample input cards shown in the previous section. Note that the base data for 1965 for Sedgwick and Stafford counties and for 1975 for Harper and Kingman counties that was missing in the input cards was filled in, with the replacement numbers now showing in the output listing.

The run that produced the following output took 4.55 seconds to execute. The cost was $1.52. If a run were made with the data from multiple districts, the output for each district would appear similar to that which follows. If a run contains more than one district, allow about 6-7 seconds job time per district (when run at K-State).

KANSAS

DISTRICT 8 13 COUNTIES 15 BASE YEARS PROJECTED 60 YEARS

SUMMARY STATISTICS FOR DISTRICT 8

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

KANSAS

COUNTY

	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
1ST PROJECTION SET													
RSQ	0.2221	0.2922	0.3468	0.0354	0.0509	0.1766	0.1925	0.4241	0.3226	0.1688	0.1373	0.5042	0.1780
ALPHA	216.73	153.08	190.52	238.56	238.50	164.04	157.20	178.17	202.60	239.47	248.61	197.86	260.67
BETA	3.68	2.17	2.94	7.69	30.27	44.80	2.08	3.58	3.32	12.53	7.58	3.59	3.31
YEAR	263.00	196.33	249.13	270.33	304.87	262.27	238.73	249.53	255.13	291.20	297.13	269.47	302.27
PRC TYPE	1.2 EXP	1.4 EXP	1.4 EXP	0.7 EXP	0.4 EXP	0.4 EXP	1.4 EXP	1.4 EXP	1.3 EXP	0.7 EXP	0.9 EXP	1.4 EXP	1.2 EXP
2ND PROJECTION SET													
RSQ	0.2218	0.2914	0.3414	0.0353	0.0508	0.1759	0.1915	0.4181	0.3226	0.1686	0.1371	0.5023	0.1776
ALPHA	219.24	151.00	187.99	241.71	250.00	181.10	155.18	175.03	199.63	232.79	244.15	194.32	262.04
BETA	2.76	2.86	3.86	5.58	20.34	30.08	2.75	4.71	4.41	17.53	10.32	4.75	2.48
YEAR	263.00	196.33	249.13	270.33	304.87	262.27	238.73	249.53	255.13	291.20	297.13	269.47	302.27
PRO TYPE	1.3 EXP	1.3 EXP	1.3 EXP	0.8 EXP	0.5 EXP	0.5 EXP	1.3 EXP	1.3 EXP	1.2 EXP	0.6 EXP	0.8 EXP	1.3 EXP	1.3 EXP
3RD PROJECTION SET													
RSQ	0.2215	0.2884	0.3348	0.0351	0.0503	0.1741	0.1895	0.4106	0.3215	0.1682	0.1367	0.4989	0.1776
ALPHA	213.88	148.67	185.15	234.56	257.82	192.71	152.94	171.51	209.26	244.62	252.30	190.33	259.10
BETA	4.90	3.79	5.08	10.73	14.12	20.87	3.64	6.20	7.50	9.07	5.60	6.29	4.41
YEAR	263.00	196.33	249.13	270.33	304.87	262.27	238.73	249.53	255.13	291.20	297.13	269.47	302.27
PRO TYPE	1.1 EXP	1.2 EXP	1.2 EXP	0.6 EXP	0.6 EXP	0.6 EXP	1.2 EXP	1.2 EXP	1.4 EXP	0.8 EXP	LINEAR	1.2 EXP	1.1 EXP

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

1ST PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
1961	300.	220.	280.	300.	320.	250.	280.	260.	270.	290.	310.	280.	340.
1962	200.	175.	220.	230.	170.	190.	220.	240.	210.	230.	220.	220.	220.
1963	150.	120.	150.	190.	300.	190.	150.	160.	150.	240.	240.	170.	230.
1964	210.	100.	180.	260.	360.	240.	150.	170.	180.	270.	310.	190.	290.
1965	290.	210.	250.	300.	340.	290.	250.	190.	290.	330.	280.	195.	320.
1966	240.	150.	160.	240.	250.	220.	170.	150.	180.	220.	250.	200.	250.
1967	160.	140.	180.	130.	130.	150.	180.	180.	220.	200.	140.	220.	180.
1968	289.	135.	174.	334.	382.	330.	162.	164.	212.	345.	375.	232.	304.
1969	332.	293.	317.	321.	306.	321.	316.	290.	299.	321.	335.	329.	351.
1970	313.	233.	206.	320.	368.	311.	307.	321.	310.	326.	352.	327.	354.
1971	268.	232.	275.	284.	382.	211.	283.	368.	312.	357.	363.	362.	368.
1972	292.	253.	355.	312.	387.	348.	326.	326.	344.	363.	364.	357.	348.
1973	376.	253.	348.	402.	366.	329.	339.	368.	343.	344.	379.	377.	409.
1974	251.	223.	226.	278.	238.	261.	224.	250.	227.	254.	273.	256.	291.
1975	274.	208.	316.	153.	274.	193.	224.	306.	280.	278.	266.	327.	279.
1976	319.	258.	333.	292.	330.	300.	298.	352.	325.	327.	341.	372.	353.
1977	327.	267.	346.	294.	333.	303.	307.	367.	335.	330.	346.	387.	360.
1978	335.	277.	358.	297.	335.	306.	316.	383.	345.	334.	351.	403.	367.
1979	343.	287.	372.	299.	337.	310.	326.	399.	355.	338.	356.	419.	374.
1980	351.	297.	385.	301.	339.	313.	335.	415.	366.	341.	361.	436.	381.
1981	359.	307.	299.	303.	341.	315.	345.	432.	376.	345.	366.	452.	388.
1982	367.	317.	413.	306.	343.	318.	355.	449.	387.	348.	371.	470.	396.
1983	375.	328.	427.	308.	345.	321.	365.	466.	398.	352.	376.	487.	403.
1984	393.	338.	442.	310.	346.	324.	375.	484.	409.	355.	381.	505.	410.
1985	392.	349.	457.	312.	348.	326.	386.	502.	420.	359.	386.	523.	418.
1986	400.	360.	472.	314.	350.	329.	396.	520.	432.	362.	391.	541.	426.
1987	409.	372.	487.	316.	352.	331.	407.	539.	443.	365.	396.	560.	433.
1988	417.	383.	502.	318.	353.	334.	418.	558.	455.	369.	401.	579.	441.
1989	426.	395.	518.	320.	355.	336.	429.	577.	467.	372.	406.	598.	449.
1990	434.	406.	534.	322.	356.	339.	441.	596.	479.	375.	410.	617.	456.
1991	443.	418.	550.	324.	358.	341.	452.	616.	491.	378.	415.	637.	464.
1992	452.	430.	566.	326.	360.	343.	464.	636.	503.	381.	420.	657.	472.
1993	461.	443.	583.	328.	361.	345.	475.	656.	515.	384.	425.	677.	480.
1994	470.	455.	600.	329.	363.	348.	487.	676.	527.	387.	430.	698.	488.
1995	479.	468.	617.	331.	364.	350.	499.	697.	540.	390.	435.	718.	496.
1996	488.	480.	634.	333.	365.	352.	511.	718.	553.	393.	439.	739.	504.
1997	497.	493.	651.	335.	367.	354.	524.	739.	565.	396.	444.	761.	512.
1998	506.	506.	669.	237.	368.	356.	536.	760.	578.	399.	449.	782.	521.
1999	515.	519.	686.	339.	370.	358.	549.	782.	591.	402.	454.	804.	529.
2000	524.	532.	704.	340.	371.	360.	561.	804.	604.	405.	458.	825.	537.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

1ST PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
2001	534.	545.	722.	342.	372.	362.	574.	826.	617.	408.	463.	847.	545.
2002	543.	559.	740.	344.	373.	364.	587.	848.	630.	411.	468.	870.	554.
2003	552.	573.	759.	346.	375.	366.	600.	870.	643.	414.	472.	892.	562.
2004	562.	586.	777.	347.	376.	368.	613.	893.	657.	417.	477.	915.	571.
2005	571.	600.	796.	349.	377.	369.	626.	916.	670.	419.	482.	938.	579.
2006	580.	614.	815.	351.	378.	371.	640.	939.	684.	422.	486.	961.	588.
2007	590.	628.	834.	352.	380.	373.	653.	962.	697.	425.	491.	984.	596.
2008	599.	642.	854.	354.	381.	375.	667.	985.	711.	428.	496.	1008.	605.
2009	609.	657.	873.	356.	382.	377.	681.	1009.	725.	430.	500.	1032.	613.
2010	619.	671.	893.	358.	383.	378.	695.	1033.	739.	433.	505.	1056.	622.
2011	628.	686.	912.	359.	384.	380.	709.	1057.	753.	436.	510.	1080.	631.
2012	638.	700.	932.	361.	386.	382.	723.	1081.	767.	439.	514.	1104.	639.
2013	648.	715.	952.	362.	387.	383.	737.	1106.	781.	441.	519.	1128.	648.
2014	658.	730.	972.	364.	388.	385.	751.	1130.	795.	444.	523.	1153.	657.
2015	667.	745.	993.	366.	389.	387.	766.	1155.	810.	447.	528.	1178.	666.
2016	677.	760.	1013.	367.	390.	388.	780.	1180.	824.	449.	532.	1203.	675.
2017	687.	775.	1034.	369.	391.	390.	795.	1205.	839.	452.	537.	1228.	684.
2018	697.	791.	1055.	371.	392.	391.	810.	1230.	853.	454.	542.	1254.	692.
2019	707.	806.	1076.	372.	393.	393.	824.	1256.	868.	457.	546.	1279.	701.
2020	717.	822.	1097.	374.	394.	394.	839.	1282.	882.	460.	551.	1305.	710.
2021	727.	837.	1118.	375.	395.	396.	854.	1307.	897.	462.	555.	1331.	719.
2022	737.	853.	1139.	377.	396.	398.	870.	1333.	912.	465.	560.	1357.	728.
2023	747.	869.	1161.	378.	397.	399.	885.	1360.	927.	467.	564.	1383.	738.
2024	757.	885.	1182.	380.	398.	400.	900.	1386.	942.	470.	569.	1410.	747.
2025	767.	901.	1204.	382.	399.	402.	915.	1412.	957.	472.	573.	1436.	756.
2026	778.	917.	1226.	383.	400.	403.	931.	1439.	972.	475.	578.	1463.	765.
2027	788.	934.	1248.	385.	401.	405.	947.	1466.	987.	477.	582.	1490.	774.
2028	798.	950.	1270.	386.	402.	406.	962.	1493.	1002.	480.	587.	1517.	783.
2029	808.	966.	1292.	388.	403.	408.	978.	1520.	1018.	482.	591.	1544.	793.
2030	819.	983.	1315.	389.	404.	409.	994.	1547.	1033.	485.	596.	1572.	802.
2031	829.	1000.	1337.	391.	405.	411.	1010.	1575.	1049.	487.	600.	1599.	811.
2032	839.	1016.	1360.	392.	406.	412.	1026.	1602.	1064.	489.	605.	1627.	820.
2033	850.	1033.	1383.	394.	407.	413.	1042.	1630.	1080.	492.	609.	1655.	830.
2034	860.	1050.	1406.	395.	408.	415.	1058.	1658.	1095.	494.	613.	1683.	839.
2035	871.	1067.	1429.	397.	409.	416.	1075.	1686.	1111.	497.	618.	1711.	849.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

1ST PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
RSQ	0.2221	0.2932	0.3468	0.0354	0.0509	0.1766	0.1925	0.4241	0.3226	0.1688	0.1373	0.5042	0.1780
ALPHA	216.73	153.08	190.52	238.56	238.50	164.04	197.20	178.17	202.60	239.47	248.61	197.96	260.67
BETA	3.66	2.17	2.94	7.69	30.27	44.80	2.08	3.58	3.32	12.53	7.58	3.59	3.31
YEAR	263.00	196.33	249.13	270.33	304.87	262.27	238.73	249.53	255.13	291.20	297.13	269.47	302.27
PRC TYPE	1.2 EXP	1.4 EXP	1.4 EXP	0.7 EXP	0.4 EXP	0.4 EXP	1.4 EXP	1.4 EXP	1.3 EXP	0.7 EXP	0.9 EXP	1.4 EXP	1.2 EXP

SOURCE:
KANSAS CROP AND LIVESTOCK REPORTING SERVICE, MIMEO. REPORTS, 1962 - 1976.
TEST RUN MADE BY WALTER TURNER. APRIL 13, 1977. TEST SHOWS THE REPLACEMENT OF MISSING DATA.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

1ST PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	DISTRICT
1961	3700.
1962	2745.
1963	2440.
1964	2910.
1965	3535.
1966	2680.
1967	2210.
1968	3438.
1969	4131.
1970	4148.
1971	4165.
1972	4375.
1973	4634.
1974	3252.
1975	3378.
1976	4199.
1977	4302.
1978	4406.
1979	4512.
1980	4620.
1981	4729.
1982	4839.
1983	4951.
1984	5064.
1985	5178.
1986	5293.
1987	5410.
1988	5527.
1989	5646.
1990	5766.
1991	5888.
1992	6010.
1993	6133.
1994	6258.
1995	6383.
1996	6510.
1997	6638.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

1ST PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	DISTRICT
2001	7158.
2002	7290.
2003	7424.
2004	7558.
2005	7694.
2006	7830.
2007	7967.
2008	8105.
2009	8244.
2010	8384.
2011	8524.
2012	8666.
2013	8808.
2014	8951.
2015	9095.
2016	9240.
2017	9385.
2018	9532.
2019	9679.
2020	9827.
2021	9975.
2022	10125.
2023	10275.
2024	10426.
2025	10577.
2026	10730.
2027	10883.
2028	11037.
2029	11191.
2030	11347.
2031	11503.
2032	11660.
2033	11817.
2034	11975.
2035	12134.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

2ND PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
1961	300.	220.	280.	300.	320.	250.	280.	260.	270.	290.	310.	280.	340.
1962	200.	175.	220.	230.	170.	190.	220.	240.	210.	230.	220.	220.	220.
1963	150.	120.	150.	190.	300.	190.	150.	160.	150.	240.	240.	170.	230.
1964	210.	100.	180.	260.	360.	240.	150.	170.	180.	270.	310.	190.	290.
1965	290.	210.	250.	300.	340.	290.	250.	190.	290.	330.	280.	195.	320.
1966	240.	150.	160.	240.	250.	220.	170.	150.	180.	220.	250.	200.	250.
1967	160.	140.	180.	130.	130.	150.	180.	190.	220.	200.	140.	220.	180.
1968	289.	135.	174.	334.	382.	330.	162.	164.	212.	345.	375.	232.	304.
1969	332.	293.	217.	321.	306.	321.	316.	290.	299.	321.	335.	329.	351.
1970	313.	233.	306.	320.	368.	311.	307.	321.	310.	326.	352.	327.	354.
1971	268.	232.	275.	284.	382.	311.	283.	369.	312.	357.	363.	362.	368.
1972	292.	253.	355.	312.	387.	248.	326.	326.	344.	363.	364.	357.	348.
1973	376.	253.	348.	402.	366.	329.	339.	368.	343.	344.	379.	377.	409.
1974	251.	223.	226.	278.	238.	261.	224.	250.	227.	254.	273.	256.	291.
1975	274.	208.	316.	153.	274.	193.	224.	306.	280.	278.	266.	327.	279.
1976	321.	256.	330.	293.	331.	301.	296.	348.	322.	325.	339.	369.	354.
1977	329.	265.	342.	295.	334.	305.	309.	362.	332.	329.	344.	383.	362.
1978	338.	274.	353.	298.	336.	309.	313.	377.	341.	332.	348.	398.	369.
1979	346.	283.	365.	300.	339.	312.	322.	391.	351.	335.	353.	412.	377.
1980	355.	292.	378.	303.	341.	316.	330.	406.	360.	339.	358.	427.	385.
1981	364.	301.	390.	305.	343.	319.	339.	421.	370.	342.	362.	443.	393.
1982	373.	310.	403.	308.	345.	322.	348.	437.	380.	345.	367.	458.	401.
1983	382.	320.	415.	310.	348.	325.	357.	452.	390.	348.	371.	474.	409.
1984	391.	329.	428.	313.	350.	329.	366.	468.	399.	351.	375.	490.	418.
1985	401.	339.	442.	315.	352.	332.	376.	484.	410.	354.	380.	506.	426.
1986	410.	349.	455.	317.	354.	334.	385.	500.	420.	357.	384.	522.	435.
1987	420.	359.	468.	320.	356.	337.	395.	516.	430.	359.	388.	539.	443.
1988	429.	369.	482.	322.	358.	340.	404.	533.	440.	362.	393.	555.	452.
1989	439.	379.	495.	324.	360.	343.	414.	550.	450.	365.	397.	572.	461.
1990	449.	389.	509.	326.	361.	346.	424.	567.	461.	368.	401.	589.	470.
1991	459.	400.	523.	329.	363.	349.	434.	584.	471.	370.	405.	606.	479.
1992	469.	410.	537.	331.	365.	351.	444.	601.	482.	373.	409.	624.	488.
1993	480.	421.	552.	333.	367.	354.	454.	618.	492.	376.	413.	641.	497.
1994	490.	431.	566.	335.	369.	357.	465.	636.	503.	378.	417.	659.	506.
1995	500.	442.	581.	338.	370.	359.	475.	653.	514.	381.	422.	677.	515.
1996	511.	453.	595.	340.	372.	362.	485.	671.	525.	383.	426.	695.	525.
1997	521.	464.	610.	342.	374.	364.	496.	688.	536.	386.	430.	713.	534.
1998	532.	475.	625.	344.	375.	367.	506.	707.	547.	388.	434.	731.	544.
1999	543.	486.	640.	346.	377.	369.	517.	726.	557.	391.	438.	750.	554.
2000	554.	497.	655.	348.	379.	371.	528.	744.	569.	393.	442.	768.	563.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

2ND PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
2001	564.	509.	670.	350.	390.	374.	539.	763.	580.	396.	445.	787.	573.
2002	575.	520.	686.	352.	382.	376.	550.	781.	591.	399.	449.	806.	583.
2003	586.	531.	701.	355.	383.	378.	561.	800.	602.	400.	453.	825.	593.
2004	598.	543.	717.	357.	385.	381.	572.	819.	613.	403.	457.	844.	603.
2005	609.	555.	732.	359.	386.	383.	583.	838.	625.	405.	461.	863.	613.
2006	620.	566.	748.	361.	388.	385.	594.	858.	636.	407.	465.	883.	623.
2007	632.	578.	764.	363.	389.	387.	606.	877.	647.	409.	469.	902.	633.
2008	643.	590.	780.	365.	391.	390.	617.	896.	659.	412.	473.	922.	644.
2009	654.	602.	796.	367.	392.	392.	628.	916.	670.	414.	476.	942.	654.
2010	666.	614.	812.	369.	394.	394.	640.	936.	682.	416.	480.	962.	664.
2011	678.	626.	829.	371.	395.	396.	651.	956.	693.	418.	484.	982.	675.
2012	689.	638.	845.	373.	397.	398.	663.	976.	705.	420.	488.	1002.	685.
2013	701.	650.	861.	375.	398.	400.	675.	996.	717.	423.	491.	1022.	696.
2014	713.	663.	878.	377.	399.	402.	687.	1016.	728.	425.	495.	1042.	707.
2015	725.	675.	895.	379.	401.	404.	699.	1036.	740.	427.	499.	1063.	717.
2016	737.	687.	911.	381.	402.	406.	710.	1056.	752.	429.	503.	1083.	728.
2017	749.	700.	928.	383.	404.	408.	722.	1077.	764.	431.	506.	1104.	739.
2018	761.	712.	945.	385.	405.	410.	735.	1098.	776.	433.	510.	1125.	750.
2019	773.	725.	962.	387.	406.	412.	747.	1118.	788.	435.	514.	1146.	761.
2020	786.	738.	979.	389.	408.	414.	759.	1139.	800.	437.	517.	1167.	772.
2021	798.	750.	996.	391.	409.	416.	771.	1160.	812.	439.	521.	1188.	783.
2022	810.	763.	1014.	393.	410.	418.	783.	1181.	824.	441.	524.	1209.	794.
2023	823.	776.	1031.	395.	411.	420.	796.	1202.	836.	443.	528.	1231.	805.
2024	835.	789.	1049.	397.	413.	422.	808.	1224.	848.	445.	532.	1252.	816.
2025	848.	802.	1066.	399.	414.	424.	821.	1245.	860.	447.	535.	1273.	828.
2026	860.	815.	1084.	401.	415.	426.	833.	1266.	872.	449.	539.	1295.	839.
2027	873.	828.	1101.	403.	416.	427.	846.	1288.	885.	451.	542.	1317.	850.
2028	886.	841.	1119.	405.	418.	429.	858.	1310.	897.	453.	546.	1339.	862.
2029	898.	855.	1137.	407.	419.	431.	871.	1331.	909.	455.	549.	1361.	873.
2030	911.	868.	1155.	409.	420.	433.	884.	1353.	922.	457.	553.	1383.	885.
2031	924.	881.	1173.	410.	421.	435.	897.	1375.	934.	459.	557.	1405.	896.
2032	937.	895.	1191.	412.	423.	436.	910.	1397.	946.	461.	560.	1427.	908.
2033	950.	908.	1209.	414.	424.	438.	923.	1419.	959.	463.	564.	1449.	920.
2034	963.	922.	1227.	416.	425.	440.	935.	1441.	971.	465.	567.	1472.	931.
2035	976.	935.	1246.	418.	426.	442.	949.	1464.	984.	467.	571.	1494.	943.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

2ND PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
RSQ	0.2218	0.2914	0.3414	0.0353	0.0508	0.1759	0.1915	0.4181	0.3226	0.1686	0.1371	0.5023	0.1776
ALPHA	219.24	151.00	187.99	241.71	250.00	181.10	195.19	175.03	199.63	232.79	244.15	194.32	262.94
BETA	2.76	2.86	3.86	5.58	20.34	30.08	2.75	4.71	4.41	17.53	10.32	4.75	2.48
YEAR	263.00	196.33	245.13	270.33	304.87	262.27	238.73	249.53	255.13	291.20	297.13	269.47	302.27
PRC TYPE	1.3 EXP	1.3 EXP	1.3 EXP	0.8 EXP	0.5 EXP	0.5 EXP	1.3 EXP	1.3 EXP	1.2 EXP	0.6 EXP	0.8 EXP	1.3 EXP	1.3 EXP

SOURCE:
KANSAS CROP AND LIVESTOCK REPORTING SERVICE, MIMEO. REPORTS, 1962 - 1976.
TEST RUN MADE BY WALTER TURNER. APRIL 13, 1977. TEST SHOWS THE REPLACEMENT OF MISSING DATA.

TABLE Z. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

2ND PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	DISTRICT
1961	3700.
1962	2745.
1963	2440.
1964	2910.
1965	3535.
1966	2680.
1967	2210.
1968	3438.
1969	4131.
1970	4148.
1971	4165.
1972	4375.
1973	4634.
1974	3252.
1975	3378.
1976	4187.
1977	4286.
1978	4386.
1979	4487.
1980	4589.
1981	4692.
1982	4796.
1983	4901.
1984	5007.
1985	5114.
1986	5222.
1987	5330.
1988	5439.
1989	5550.
1990	5661.
1991	5772.
1992	5885.
1993	5998.
1994	6112.
1995	6227.
1996	6342.
1997	6459.
1998	6576.
1999	6693.
2000	6811.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

2ND PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	DISTRICT
2001	6930.
2002	7050.
2003	7170.
2004	7291.
2005	7412.
2006	7534.
2007	7657.
2008	7780.
2009	7904.
2010	8029.
2011	8154.
2012	8279.
2013	8406.
2014	8532.
2015	8660.
2016	8787.
2017	8916.
2018	9045.
2019	9174.
2020	9304.
2021	9435.
2022	9566.
2023	9697.
2024	9829.
2025	9962.
2026	10095.
2027	10228.
2028	10362.
2029	10497.
2030	10632.
2031	10767.
2032	10903.
2033	11039.
2034	11176.
2035	11313.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

3RD PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
1961	300.	220.	280.	300.	320.	250.	280.	260.	270.	290.	310.	280.	340.
1962	200.	175.	220.	230.	170.	190.	220.	240.	210.	230.	220.	220.	220.
1963	150.	120.	150.	190.	300.	190.	150.	160.	150.	240.	240.	170.	230.
1964	210.	100.	180.	260.	360.	240.	150.	170.	180.	270.	310.	190.	290.
1965	290.	210.	250.	200.	340.	290.	250.	190.	290.	330.	280.	195.	320.
1966	240.	150.	160.	240.	250.	220.	170.	150.	190.	220.	250.	200.	250.
1967	160.	140.	180.	130.	130.	150.	180.	180.	220.	200.	140.	220.	180.
1968	289.	135.	174.	334.	382.	330.	162.	164.	212.	345.	375.	232.	304.
1969	332.	293.	317.	321.	336.	321.	316.	290.	299.	321.	335.	329.	351.
1970	313.	233.	306.	320.	360.	311.	307.	321.	310.	326.	352.	327.	354.
1971	268.	232.	275.	284.	302.	311.	283.	368.	312.	357.	363.	362.	368.
1972	292.	253.	355.	312.	387.	348.	326.	326.	344.	363.	364.	357.	348.
1973	376.	253.	348.	403.	366.	329.	339.	368.	343.	344.	379.	377.	409.
1974	251.	223.	235.	278.	238.	261.	224.	250.	227.	254.	273.	256.	291.
1975	274.	208.	316.	153.	274.	193.	224.	306.	200.	278.	266.	327.	279.
1976	317.	254.	327.	291.	332.	303.	294.	344.	326.	328.	342.	365.	351.
1977	324.	262.	337.	293.	335.	307.	302.	357.	337.	332.	348.	379.	359.
1978	332.	270.	348.	295.	338.	311.	310.	370.	348.	336.	353.	392.	364.
1979	339.	278.	359.	297.	340.	315.	318.	384.	359.	340.	359.	406.	370.
1980	346.	287.	370.	299.	343.	319.	325.	397.	371.	344.	364.	419.	377.
1981	353.	295.	381.	301.	346.	322.	333.	411.	383.	348.	370.	433.	384.
1982	361.	303.	393.	302.	348.	326.	341.	425.	395.	352.	376.	447.	390.
1983	368.	312.	404.	305.	350.	330.	350.	438.	407.	356.	381.	461.	397.
1984	375.	320.	416.	307.	353.	333.	358.	452.	419.	360.	387.	475.	403.
1985	383.	329.	427.	309.	355.	337.	366.	467.	432.	364.	392.	490.	410.
1986	390.	338.	439.	310.	358.	340.	374.	481.	444.	368.	398.	504.	417.
1987	398.	346.	450.	312.	360.	344.	383.	495.	457.	371.	404.	519.	424.
1988	405.	355.	462.	314.	362.	347.	391.	510.	471.	375.	409.	533.	430.
1989	413.	364.	474.	316.	364.	350.	400.	524.	484.	379.	415.	548.	437.
1990	420.	373.	486.	317.	366.	353.	408.	539.	497.	382.	420.	563.	444.
1991	428.	382.	498.	319.	369.	357.	417.	553.	511.	386.	426.	578.	451.
1992	436.	391.	511.	320.	371.	360.	426.	568.	525.	390.	432.	593.	458.
1993	443.	400.	523.	322.	373.	364.	435.	583.	539.	393.	437.	608.	464.
1994	451.	409.	535.	324.	375.	366.	443.	598.	553.	397.	443.	623.	471.
1995	459.	419.	547.	325.	377.	369.	452.	613.	569.	401.	448.	638.	478.
1996	466.	428.	560.	327.	379.	372.	461.	629.	583.	404.	454.	654.	485.
1997	474.	437.	572.	328.	381.	375.	470.	644.	597.	408.	460.	669.	492.
1998	482.	447.	585.	330.	383.	378.	479.	659.	612.	411.	465.	685.	499.
1999	490.	456.	598.	331.	385.	381.	488.	675.	627.	415.	471.	701.	506.
2000	497.	465.	610.	333.	387.	384.	497.	690.	642.	418.	476.	716.	513.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

3RD PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
2001	505.	475.	623.	334.	389.	386.	506.	706.	658.	422.	482.	732.	520.
2002	513.	485.	636.	336.	391.	389.	516.	721.	673.	425.	488.	748.	527.
2003	521.	494.	649.	337.	393.	392.	525.	737.	689.	428.	493.	764.	534.
2004	529.	504.	662.	339.	395.	395.	534.	753.	705.	432.	499.	780.	541.
2005	537.	514.	675.	340.	396.	398.	543.	769.	721.	435.	504.	796.	548.
2006	544.	523.	688.	341.	398.	401.	553.	785.	737.	439.	510.	812.	555.
2007	552.	533.	701.	343.	400.	403.	562.	801.	753.	442.	516.	829.	562.
2008	560.	543.	714.	344.	402.	406.	572.	817.	769.	445.	521.	845.	570.
2009	568.	553.	728.	345.	404.	408.	581.	833.	786.	449.	527.	861.	577.
2010	576.	563.	741.	347.	405.	411.	591.	849.	803.	452.	532.	878.	584.
2011	584.	573.	754.	348.	407.	414.	601.	866.	819.	455.	538.	894.	591.
2012	592.	583.	768.	349.	409.	416.	610.	882.	836.	459.	544.	911.	598.
2013	600.	593.	781.	351.	411.	419.	620.	898.	853.	462.	549.	928.	605.
2014	608.	603.	795.	352.	412.	421.	629.	915.	871.	465.	555.	944.	613.
2015	616.	613.	808.	353.	414.	424.	639.	931.	888.	469.	561.	961.	620.
2016	624.	623.	822.	355.	416.	426.	649.	948.	905.	472.	566.	978.	627.
2017	632.	633.	836.	356.	418.	429.	658.	965.	923.	475.	572.	995.	634.
2018	640.	643.	849.	357.	419.	431.	668.	981.	941.	478.	577.	1012.	642.
2019	649.	654.	863.	359.	421.	434.	678.	998.	958.	481.	583.	1029.	649.
2020	657.	664.	877.	360.	423.	436.	688.	1015.	976.	485.	589.	1046.	656.
2021	665.	674.	891.	361.	424.	439.	698.	1032.	994.	488.	594.	1063.	664.
2022	673.	685.	905.	362.	426.	441.	708.	1049.	1013.	491.	600.	1080.	671.
2023	681.	695.	919.	364.	427.	443.	718.	1066.	1031.	494.	605.	1097.	678.
2024	689.	705.	933.	365.	429.	446.	728.	1083.	1049.	497.	611.	1115.	686.
2025	697.	716.	947.	366.	431.	448.	738.	1100.	1068.	501.	617.	1132.	693.
2026	706.	726.	961.	367.	432.	451.	748.	1117.	1086.	504.	622.	1150.	700.
2027	714.	737.	975.	368.	434.	453.	758.	1135.	1105.	507.	628.	1167.	708.
2028	722.	747.	989.	370.	435.	455.	768.	1152.	1124.	510.	633.	1185.	715.
2029	730.	758.	1003.	371.	437.	457.	778.	1169.	1143.	513.	639.	1202.	722.
2030	739.	769.	1017.	372.	438.	460.	789.	1186.	1162.	516.	645.	1220.	730.
2031	747.	779.	1032.	373.	440.	462.	799.	1204.	1181.	519.	650.	1237.	737.
2032	755.	790.	1046.	374.	442.	464.	809.	1221.	1201.	522.	656.	1255.	745.
2033	763.	801.	1060.	375.	443.	467.	819.	1239.	1220.	525.	661.	1273.	752.
2034	772.	811.	1075.	377.	445.	469.	830.	1256.	1240.	528.	667.	1291.	760.
2035	780.	822.	1089.	378.	446.	471.	840.	1274.	1259.	532.	673.	1309.	767.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

3RD PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

	BARBER	COMANCHE	EDWARDS	HARPER	HARVEY	KINGMAN	KIOWA	PAWNEE	PRATT	RENO	SEDGWICK	STAFFORD	SUMNER
RSQ	0.2215	0.2884	0.3348	0.0351	0.0503	0.1741	0.1895	0.4106	0.3215	0.1682	0.1367	0.4989	0.1776
ALPHA	213.88	148.67	185.15	234.56	257.82	192.71	192.94	171.51	205.26	244.62	252.30	190.33	258.10
BETA	4.70	3.79	5.08	10.72	14.12	20.87	3.64	6.20	7.50	9.07	5.60	6.29	4.41
YEAR	263.00	196.33	249.13	270.23	304.87	262.27	239.73	249.53	255.13	291.20	297.13	269.47	302.27
PRO TYPE	1.1 EXP	1.2 EXP	1.2 EXP	0.6 EXP	0.6 EXP	0.6 EXP	1.2 EXP	1.2 EXP	1.4 EXP	0.8 EXP	LINEAR	1.2 EXP	1.1 EXP

SOURCE:
KANSAS CROP AND LIVESTOCK REPORTING SERVICE, MIMEO. REPORTS, 1962 - 1976.
TEST RUN MADE BY WALTER TURNER. APRIL 13, 1977. TEST SHOWS THE REPLACEMENT OF MISSING DATA.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

3RD PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	DISTRICT
1961	3700.
1962	2745.
1963	2440.
1964	2910.
1965	3535.
1966	2680.
1967	2210.
1968	3438.
1969	4131.
1970	4148.
1971	4165.
1972	4375.
1973	4634.
1974	3252.
1975	3378.
1976	4176.
1977	4272.
1978	4368.
1979	4465.
1980	4562.
1981	4661.
1982	4759.
1983	4859.
1984	4959.
1985	5059.
1986	5161.
1987	5262.
1988	5365.
1989	5467.
1990	5571.
1991	5675.
1992	5779.
1993	5884.
1994	5989.
1995	6095.
1996	6201.
1997	6307.
1998	6415.
1999	6522.
2000	6630.

TABLE 2. REPORTED AVERAGE YIELD OF WHEAT HARVESTED FOR GRAIN BY COUNTY, 1961 TO 1975 AND PROJECTION THROUGH 2035

YIELD PER HARVESTED ACRE (UNIT=TENTHS OF BUSHELS)

3RD PROJECTION SET — DISTRICT 8

KANSAS

COUNTY

YEAR	DISTRICT
2001	6738.
2002	6847.
2003	6956.
2004	7066.
2005	7176.
2006	7287.
2007	7397.
2008	7509.
2009	7620.
2010	7732.
2011	7844.
2012	7957.
2013	8070.
2014	8183.
2015	8297.
2016	8411.
2017	8526.
2018	8640.
2019	8755.
2020	8871.
2021	8987.
2022	9103.
2023	9219.
2024	9336.
2025	9453.
2026	9570.
2027	9688.
2028	9805.
2029	9924.
2030	10042.
2031	10161.
2032	10280.
2033	10400.
2034	10519.
2035	10639.

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

IV. PROGRAM LISTING

```
//STEP1 EXEC ASMGC
//ASM.SYSIN DD *
***********************************************************************
***********************************************************************
*    THIS PROGRAM CENTERS A TITLE PASSED IN WITHIN A 132 SPACE LINE AND
*    PASSES THE RESULTING CENTERED LINE BACK TO THE CALLING PROGRAM.
*    THE TITLE CAN BE ONE OR TWO CARDS IN LENGTH.  IF THE TITLE TO BE
*    PASSED TO THIS ROUTINE IS MORE THAN ONE CARD IN LENGTH, THE FIRST
*    CHARACTER OF THE FIRST CARD SHOULD BE A '+' TO TELL THIS ROUTINE
*    THAT IT WILL BE CENTERING A TWO CARD TITLE.
***********************************************************************
***********************************************************************
*    R2 = BEGINNING ADDRESS OF TEXT.
*    R5 = ENDING ADDRESS OF TEXT.
*    R6 = TOP HALF OF R7 FOR DIVISION.
*    R7 = # PADDING CHARACTERS FOR FRONT.
*    R9 = ADDRESS OF FORTRAN'S CHAR. STRING OUTPUT VECTOR.
***********************************************************************
CENTER   CSECT
         PRINT NOGEN
         STM   14,12,12(13)        ST REGS. IN AREA GIVEN BY FORTRAN.
         BALR  12,0
         USING *,12
         LA    14,SAV              LOAD R14 WITH ADDR. OF SUB SAVE AREA
         ST    13,4(14)            PUT R13 INTO SUB SAVE AREA.
         ST    14,8(13)            PUT R14 (KEY TO SUB SAVE AREA) INTO
*                                    FORTRAN SAVE AREA.
         LR    13,14               PUT KEY TO SUB SAVE AREA IN R13.
         L     9,0(1)              LOAD 9 WITH ADDRESS OF LINE IN MAIN.
         MVC   0(132,9),=CL132' '  BLANK OUT OUTPUT VECTOR.
         CLI   FLAG,C'/'           TEST FOR FLAG /. IF IT OCCURS THE
         BE    AGIN                  INPUT HAS ALREADY BEEN OPENED.
         OPEN  (CARD,INPUT)        PREPARE FOR INPUT.
         MVI   FLAG,C'/'           SET FLAG THAT INPUT HAS BEEN OPENED.
AGIN     GET   CARD,WAIT           READ A CARD INTO WAIT.
         CLI   WAIT,C'+'           TEST 1ST CHAR OF INPUT FOR A '+'.
         BE    PLUS                IF PLUS IS FOUND, BRANCH TO PLUS.
***** FOR A 1 CARD TITLE.
         LA    2,WAIT              LOAD R2 WITH ADDR. OF 1ST CHAR INPUT
         LA    5,WAIT+79           LOAD R5 WITH LAST CHAR (INPUT) ADDR.
         B     CONTIN              BRANCH TO CONTIN.
***** FOR A 2 CARD TITLE.
PLUS     GET   CARD,WAIT+80        READ CARD INTO WAIT AND PUT IT AT THE
*                                    END OF THE FIRST CARD IMAGE .
         LA    5,WAIT+132          LOAD R5 WITH LAST CHAR (INPUT) ADDR.
         LA    2,WAIT+1            LOAD R2 WITH ADDR. OF 2ND CHAR IN
*                                    WAIT, SO '+' WILL BE DISCARDED.
***** FIND LAST NON-BLANK CHARACTER.
CONTIN   CLI   0(5),C' '           TEST FOR BLANK CHARACTER.
         BNE   FOUND               IF BLANK CHAR NOT FOUND (=LAST NON-
*                                    BLANK CHAR),GO TO FOUND.
         BCTR  5,0                 OTHERWISE,SET PTR.R5 BACK 1,CONTINUE
*****IF CARD(S) BLANK, RETURN BLANK.
         CR    5,2                 COMPARE PTR.R5 TO PTR.R2.
         BE    RETURN              IF THEY ARE =,INPUT CARD WAS BLANK,
*                                    SO RETURN.
         B     CONTIN              OTHERWISE, CONTINUE TEST FOR BLANK.
FOUND    SR    5,2                 SUBTRACT PTR.R2 FROM PTR.R5, THE
         LA    5,1(5)                DIFFERENCE+1 = # CHARS IN TEXT.
***** R5 = # CHARACTERS IN THE TEXT.
```

THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH ILLEGIBLE
PAGE NUMBERS
THAT ARE CUT OFF,
MISSING OR OF POOR
QUALITY TEXT.

THIS IS AS RECEIVED
FROM THE
CUSTOMER.

```
         LA    7,132                LOAD R7 WITH # CHARS IN OUT VECTOR.    AUT00590
         SR    7,5                  CALC. # PADDING BLANKS (R7-R5).        AUT00600
         SR    6,6                  ZERO R6.                               AUT00610
         D     6,=F'2'              DIVIDE # BLANKS BY 2 TO FIND # TO      AUT00620
*                                     USE IN CENTERING THE INPUT TITLE.    AUT00630
         AR    9,7                  ADD R7 (# PADDING BLANKS) TO R9.       AUT00640
         LA    6,1                  SUBTRACT 1 FROM # CHARS TO MOVE AS     AUT00650
         SR    5,6                    EX MOVES 1 EXTRA CHAR.               AUT00660
         EX    5,MOVE               MOVE THE NUMBER OF CHARS SPECIFIED     AUT00670
*                                     BY R5 ACCORDING TO STMT. MOVE.       AUT00680
RETURN   L     13,4(13)             RETR. KEY TO FORTRAN'S REG. STORAGE.   AUT00690
         LM    14,12,12(13)         RESTORE FORTRAN'S REGISTERS.           AUT00700
         MVI   12(13),X'FF'         INDICATE TO FORTRAN WE ARE DONE.       AUT00710
         BR    14                   RETURN.                                AUT00720
FLAG     DC    CL1' '                                                      AUT00730
SAV      DS    18F                                                         AUT00740
CARD     DCB   MACRF=GM,DDNAME=CARDS,EODAD=RETURN,DSORG=PS                 AUT00750
WAIT     DS    CL160                                                       AUT00760
MOVE     MVC   0(*-*,9),0(2)        MOVE CHARS FROM LOC(R2) TO LOC(R9).    AUT00770
         END                                                               AUT00780
/*                                                                         AUT00790
//STEP2 EXEC ASMGC                                                         AUT00800
//ASM.SYSIN DD *                                                           AUT00810
************************************************************************   AUT00820
************************************************************************   AUT00820
*   THIS PROGRAM READS A TITLE AND PASSES IT BACK TO THE CALLING           AUT00830
*   ROUTINE.  THE TITLE CAN BE ONE OR TWO CARDS IN LENGTH.  IF THE         AUT00840
*   TITLE TO BE READ IS ONE CARD IN LENGTH, JUST PUT THE CARD IN NOR-      AUT00850
*   MALLY.  IF THE TITLE TO BE READ IN TAKES TWO CARDS, THEN PUT A '+'     AUT00860
*   IN THE FIRST SPOT ON THE FIRST CARD.                                   AUT00870
************************************************************************   AUT00880
*   R2 = BEGINNING ADDRESS OF TEXT.                                        AUT00890
*   R5 = ENDING ADDRESS OF TEXT.                                           AUT00900
*   R9 = ADDRESS OF FORTRAN'S CHAR. STRING OUTPUT VECTOR.                  AUT00910
************************************************************************   AUT00920
READ     CSECT                                                             AUT00930
         PRINT NOGEN                                                       AUT00940
         STM   14,12,12(13)         ST REGS. IN AREA GIVEN BY FORTRAN.     AUT00950
         BALR  12,0                                                        AUT00960
         USING *,12                                                        AUT00970
         LA    14,SAV               LOAD R14 WITH ADDR. OF SUB SAVE AREA   AUT00980
         ST    13,4(14)             PUT R13 INTO SUB SAVE AREA.            AUT00990
         ST    14,8(13)             PUT R14 (KEY TO SUB SAVE AREA) INTO    AUT01000
*                                     FORTRAN SAVE AREA.                   AUT01010
         LR    13,14                PUT KEY TO SUB SAVE AREA IN R13.       AUT01020
         L     9,0(1)               LOAD 9 WITH ADDRESS OF LINE IN MAIN.   AUT01030
         MVC   0(124,9),=CL124' '   BLANK OUT OUTPUT VECTOR.               AUT01040
         CLI   FLAG,C'/'            TEST FOR FLAG /. IF IT OCCURS THE      AUT01050
         BE    AGIN                   INPUT HAS ALREADY BEEN OPENED.       AUT01060
         OPEN  (CARD,INPUT)         PREPARE FOR INPUT.                     AUT01070
         MVI   FLAG,C'/'            SET FLAG THAT INPUT HAS BEEN OPENED.   AUT01080
AGIN     GET   CARD,WAIT            READ A CARD INTO WAIT.                 AUT01090
         CLI   WAIT,C'+'            TEST 1ST CHAR OF INPUT FOR A '+'.      AUT01100
         BE    PLUS                 IF PLUS IS FOUND, BRANCH TO PLUS.      AUT01110
***** FOR A 1 CARD TITLE.                                                  AUT01120
         LA    2,WAIT               LOAD R2 WITH ADDR. OF 1ST CHAR INPUT   AUT01130
         LA    5,WAIT+79            LOAD R5 WITH LAST CHAR (INPUT) ADDR.   AUT01140
         B     FOUND                BRANCH TO FOUND.                       AUT01150
***** FOR A 2 CARD TITLE.                                                  AUT01160
PLUS     GET   CARD,WAIT+80         READ CARD INTO WAIT AND PUT IT AT THE  AUT01170
```

```
         LA    5,WAIT+124          LOAD R5 WITH LAST CHAR (INPUT) ADDR.   AUT01180
         LA    2,WAIT+1            LOAD R2 WITH ADDR. OF 2ND CHAR IN      AUT01190
*                                    WAIT, SO '+' WILL BE DISCARDED.      AUT01200
FOUND    SR    5,2                 SUBTRACT PTR.R2 FROM PTR.R5, = THE     AUT01210
*                                    ACTUAL NUMBER-1.                     AUT01220
***** R5 = # CHARACTERS IN THE TEXT TO MOVE (=ACTUAL-1).                   AUT01230
         EX    5,MOVE              MOVE THE NUMBER OF CHARS SPECIFIED     AUT01240
*                                    BY R5 ACCORDING TO STMT. MOVE.       AUT01250
RETURN   L     13,4(13)            RETR. KEY TO FORTRAN'S REG. STORAGE.   AUT01260
         LM    14,12,12(13)        RESTORE FORTRAN'S REGISTERS.           AUT01270
         MVI   12(13),X'FF'        INDICATE TO FORTRAN WE ARE DONE.       AUT01280
         BR    14                  RETURN.                                AUT01290
FLAG     DC    CL1' '                                                     AUT01300
SAV      DS    18F                                                        AUT01310
CARD     DCB   MACRF=GM,DDNAME=INCARD,EODAD=RETURN,DSORG=PS               AUT01320
WAIT     DS    CL160                                                      AUT01330
MOVE     MVC   0(*-*,9),0(2)       MOVE CHARS FROM LOC(R2) TO LOC(R9).    AUT01340
         END                                                              AUT01350
/*                                                                        AUT01360
//STEP3 EXEC ASMGC                                                       AUT01370
//ASM.SYSIN DD *                                                         AUT01380
COUNTY   CSECT                                                          AUT01390
*********************************************************************** AUT01400
*   THIS SUBROUTINE AUTOMATICALLY CENTERS THE COUNTY NAMES ABOVE THEIR  AUT01410
*   COLUMN IN THE OUTPUT TABLES.  THE INPUT TO THIS PROGRAM CAN BE UP   AUT01420
*   TO 6 CARDS IN LENGTH.  THE NUMBER OF CARDS OF INPUT MUST BE PUT     AUT01430
*   IN THE FIRST COLUMN OF CARD ONE.  THE COUNTY NAMES CAN BE ANY       AUT01440
*   LENGTH, BUT THIS ROUTINE WILL TRUNCATE THEM TO 8 CHARACTERS.  THE   AUT01450
*   COUNTY NAMES MUST BE SEPARATED BY A COMMA AND CANNOT CONTAIN MORE   AUT01460
*   THAN TWO CONSECUTIVE IMBEDDED BLANKS.  THE LAST COUNTY NAME MUST    AUT01470
*   BE FOLLOWED BY A COMMA.  THE TITLE 'DISTRICT' SHOULD NOT BE INPUT,  AUT01480
*   AS THIS ROUTINE SUPPLIES IT.                                        AUT01490
*********************************************************************** AUT01500
*********************************************************************** AUT01500
         PRINT NOGEN                                                    AUT01510
         STM   14,12,12(13)        ST REGS. IN AREA GIVEN BY FORTRAN.   AUT01520
         BALR  12,0                                                     AUT01530
         USING *,12                                                     AUT01540
         LA    14,SAV              LOAD R14 WITH ADDR. OF SUB SAVE AREA AUT01550
         ST    13,4(14)            PUT R13 INTO SUB SAVE AREA.          AUT01560
         ST    14,8(13)            PUT R14 (KEY TO SUB SAVE AREA) INTO  AUT01570
*                                    FORTRAN SAVE AREA.                   AUT01580
         LR    13,14               PUT KEY TO SUB SAVE AREA IN R13.     AUT01590
*********************************************************************** AUT01600
*********************************************************************** AUT01600
*   R2  = COUNTER OF THE NUMBER OF CHARS IN EACH COUNTY NAME.           AUT01610
*   R4  = PTR TO BEGIN OF COUNTY NAME.  USED AS REFERENCE TO MOVE IT.   AUT01620
*   R5  = POINTER TO CURRENT POSITION IN INPUT LINE.                    AUT01630
*   R8  = COUNTER OF THE NUMBER OF COUNTIES IN THE OUTPUT VECTOR.       AUT01640
*   R9  = POINTER TO END OF INPUT LINE.                                 AUT01650
*   R10 = POINTER TO LAST CHAR WRITTEN INTO THE OUTPUT VECTOR.          AUT01660
*********************************************************************** AUT01670
         L     10,0(1)             SET PTR R10 TO BEGIN OF OUTPUT LINE. AUT01680
         LA    8,0                 SET COUNTY COUNTER TO ZERO.          AUT01690
         MVC   0(240,10),=CL240' '  BLANK OUT OUTPUT VECTOR.            AUT01700
         LA    10,1(10)            MOVE PTR 1 TO RIGHT TO LEAVE 1ST     AUT01710
*                                    SPOT IN OUTPUT VECTOR BLANK.         AUT01720
         LA    4,WAIT+1            SET PTR R4 TO BEGIN OF INPUT CHARS.  AUT01730
         LA    5,WAIT+1            SET PTR R5 TO BEGIN OF INPUT CHARS.  AUT01740
         CLI   OPENER,C'/'         TEST FOR FLAG /.  IF IT OCCURS, THE  AUT01750
```

```
         BE    AGIN                  INPUT HAS ALREADY BEEN OPENED.      AUT01760
         OPEN  (CARD,INPUT)          PREPARE FOR INPUT.                  AUT01770
         MVI   OPENER,C'/'           SET FLAG THAT INPUT HAS BEEN OPENED. AUT01780
*** FOR 1 CARD OF INPUT.                                                 AUT01790
AGIN     GET   CARD,WAIT             READ 1 CARD INTO WAIT.              AUT01800
         CLI   WAIT,C'1'             TEST FOR NO. CARDS = 1.             AUT01810
         BNE   AGIN2                 IF NOT = 1, READ ANOTHER CARD.      AUT01820
         LA    9,WAIT+79             SET PTR R9 TO END OF INPUT LINE.    AUT01830
         B     CONTIN                GO TO THE NEXT SECTION.             AUT01840
*** FOR 2 CARDS OF INPUT.                                                AUT01850
AGIN2    GET   CARD,WAIT+80          PUT 2ND CARD INTO WAIT AT END OF 1ST AUT01860
         CLI   WAIT,C'2'             TEST FOR NO. CARDS = 2.             AUT01870
         BNE   AGIN3                 IF NOT = 2,READ ANOTHER CARD.       AUT01880
         LA    9,WAIT+159            SET PTR R9 TO END OF INPUT LINE.    AUT01890
         B     CONTIN                GO TO THE NEXT SECTION.             AUT01900
*** FOR 3 CARDS OF INPUT.                                                AUT01910
AGIN3    GET   CARD,WAIT+160         PUT 3RD CARD INTO WAIT AT END OF 2ND AUT01920
         CLI   WAIT,C'3'             TEST FOR NO. CARDS = 3.             AUT01930
         BNE   AGIN4                 IF NOT = 3,READ ANOTHER CARD.       AUT01940
         LA    9,WAIT+239            SET PTR R9 TO END OF INPUT LINE.    AUT01950
         B     CONTIN                GO TO THE NEXT SECTION.             AUT01960
*** FOR 4 CARDS OF INPUT.                                                AUT01970
AGIN4    GET   CARD,WAIT+240         PUT 4TH CARD INTO WAIT AT END OF 3RD AUT01980
         CLI   WAIT,C'4'             TEST FOR NO. CARDS = 4.             AUT01990
         BNE   AGIN5                 IF NOT = 4,READ ANOTHER CARD.       AUT02000
         LA    9,WAIT+319            SET PTR R9 TO END OF INPUT LINE.    AUT02010
         B     CONTIN                GO TO THE NEXT SECTION.             AUT02020
*** FOR 5 CARDS OF INPUT.                                                AUT02030
AGIN5    GET   CARD,WAIT+320         PUT 5TH CARD INTO WAIT AT END OF 4TH AUT02040
         CLI   WAIT,C'5'             TEST FOR NO. CARDS = 5.             AUT02050
         BNE   AGIN6                 IF NOT = 5,READ ANOTHER CARD.       AUT02060
         LA    9,WAIT+399            SET PTR R9 TO END OF INPUT LINE.    AUT02070
         B     CONTIN                GO TO THE NEXT SECTION.             AUT02080
*** FOR 6 CARDS OF INPUT.                                                AUT02090
AGIN6    GET   CARD,WAIT+400         PUT 6TH CARD INTO WAIT AT END OF 5TH AUT02100
         LA    9,WAIT+479            SET PTR R9 TO END OF INPUT LINE.    AUT02110
*************************************************************************AUT02120
CONTIN   LA    2,0                   SET COUNTER TO 0.                   AUT02130
         LA    11,1                  LOAD R11 WITH 1.                    AUT02140
         LA    3,8                   SET LIMIT TO 8.                     AUT02150
NEXT     CLI   0(5),C','             SEARCH FOR A COMMA.                 AUT02160
         BE    COMMA                 IF A COMMA IS FOUND, GO TO COMMA.   AUT02170
         CR    3,2                   TEST FOR # CHARS = 8.               AUT02180
         BE    STILL                 IF # CHARS = 8 DON'T INCREMENT CTR. AUT02190
         LA    2,1(2)                INCREMENT COUNTER.                  AUT02200
STILL    LA    5,1(5)                INCREMENT PTR R5.                   AUT02210
         B     NEXT                  CONTINUE LOOP NEXT.                 AUT02220
COMMA    LA    5,1(5)                INCREMENT PTR R5 SO IT WILL POINT   AUT02230
*                                      TO BEGIN OF NEXT COUNTY NAME.     AUT02240
         LA    8,1(8)                INCREMENT COUNTER.                  AUT02250
         SR    3,11                  SET LIMIT TO 7.                     AUT02260
         CR    3,2                   TEST # CHARS IN COUNTY NAME AGAINST AUT02270
*                                      LIMIT (R3).                       AUT02280
         BH    CENTER                IF R2 < 7 GO TO CENTER.             AUT02290
***COUNTY NAME TO PUT IN OUTPUT IS 7 OR 8 CHARS.                         AUT02300
         SR    7,7                   ZERO R7 (# BLANKS TO PAD FRONT WITH) AUT02310
         B     MOVEIT                GO TO MOVEIT.                       AUT02320
***COUNTY NAME IS LESS THAN 7 CHARS LONG, MUST BE CENTERED.              AUT02330
CENTER   LA    7,8                   LOAD R7 WITH 8.                     AUT02340
         SR    7,2                   FIND NUMBER OF PADDING BLANKS.      AUT02350
```

```
         SR    6,6                 ZERO R6 FOR DIVISION.                    AUT02360
         D     6,=F'2'             FIND # OF BLANKS TO PAD FRONT WITH.      AUT02370
*****************************************************************         AUT02380
*** R6  = # PADDING BLANKS IN FRONT OF COUNTY NAME.           ***         AUT02390
*** R4  = START OF COUNTY NAME.                               ***         AUT02400
*** R2  = LENGTH OF COUNTY NAME.                              ***         AUT02410
*** R10 = DESTINATION OF COUNTY NAME.                         ***         AUT02420
*****************************************************************         AUT02430
MOVEIT   AR    10,7                MOVE PTR TO OUTPUT VECTOR FORWARD.       AUT02440
*                                    TO SKIP PADDING BLANKS.                AUT02450
         SR    2,11                SUBTRACT 1 FROM # CHARS TO MOVE AS       AUT02460
*                                    EXEC MOVES 1 TOO MANY CHARS.           AUT02470
         EX    2,MOVE              MOVE # CHARS SPECIFIED BY R2 ACCORD-     AUT02480
*                                    ING TO STMT. MOVE.                     AUT02490
         LA    2,8                 LOAD R2 WITH 8.                          AUT02500
         SR    2,7                 FIND DIFFERENCE OF R2-NUMBER OF          AUT02510
*                                    SPACES PTR R10 HAS ALREADY BEEN        AUT02520
*                                    MOVED TO ALLOW FOR PADDING BLANKS.     AUT02530
         AR    10,2                ADD THIS NUMBER TO PTR 10 SO IT          AUT02540
*                                    NOW POINTS TO SPOT IN OUTPUT           AUT02550
*                                    VECTOR WHERE NEXT NAME GOES.           AUT02560
         LA    10,1(10)            MOVE OUTPUT PTR TO LEAVE SPACE           AUT02570
*                                    BETWEEN COUNTY NAMES IN OUTPUT.        AUT02580
***TEST FOR SPOT IN OUTPUT VECTOR THAT = THE END OF TITLE FOR ONE            AUT02590
***  PAGE HEADING IN MAIN.  IF FOUND PUT IN TWO SPACES.                     AUT02600
         C     8,=F'13'            TEST FOR 13 COUNTIES,TIME TO SPACE.      AUT02610
         BNE   NOSPACE             IF NOT = 118 DON'T SPACE.                AUT02620
         LA    10,3(10)            PUT IN THREE BLANK SPACES.               AUT02630
***SEARCH FOR 3 CONSECUTIVE BLANKS IN INPUT OR INPUT END.                    AUT02640
NOSPACE  LR    6,5                 LOAD R6 WITH ADDRESS OF INPUT PTR.       AUT02650
         LA    3,3                 LOAD R3 WITH 3.                          AUT02660
         SR    7,7                 ZERO R7 (=COUNTER).                      AUT02670
LOOP     CLI   0(6),C' '           SEARCH FOR A BLANK CHAR.                 AUT02680
         BNE   GOMORE              IF BLANK NOT FOUND, GO TO GOMORE.        AUT02690
         LA    7,1(7)              INCREMENT COUNTER R7 BY 1.               AUT02700
         CR    7,3                 TEST FOR COUNTER=3 (3 BLANKS).           AUT02710
         BE    RETURN              IF 3 BLANKS, GO TO RETURN.               AUT02720
         LA    6,1(6)              INCREMENT POINTER BY 1.                  AUT02730
         B     LOOP                GO TO LOOP.                              AUT02740
GOMORE   CR    6,9                 TEST FOR END OF INPUT VECTOR, USING      AUT02750
*                                  R6 IN CASE THE LAST OF THE INPUT         AUT02760
*                                    CONSISTS OF LESS THAN 3 BLANKS.        AUT02770
         BE    RETURN              GO TO RETURN IF END OF INPUT.            AUT02780
         BH    RETURN                                                       AUT02790
         LR    4,5                 SET PTR R4 TO BEGINNING OF NEXT          AUT02800
*                                    COUNTY NAME.                           AUT02810
         B     CONTIN              GO TO CONTIN.                            AUT02820
RETURN   MVC   0(8,10),DIST        PUT 'DISTRICT' AT END OF OUTPUT.         AUT02830
         L     13,4(13)            RETR. KEY TO FORTRAN'S REG. STORAGE.     AUT02840
         LM    14,12,12(13)        RESTORE FORTRAN'S REGISTERS.             AUT02850
         MVI   12(13),X'FF'        INDICATE TO FORTRAN WE ARE DONE.         AUT02860
         BR    14                  RETURN.                                  AUT02870
SAV      DS    18F                                                          AUT02880
CARD     DCB   MACRF=GM,DDNAME=GETUM,EODAD=RETURN,DSORG=PS                  AUT02890
WAIT     DS    CL480               RESERVE STORAGE FOR 6 CARDS OF INPUT     AUT02900
OPENER   DC    CL1' '                                                       AUT02910
MOVE     MVC   0(*-*,10),0(4)      MOVE CHARS FROM LOC(R4) TO LOC(R10).     AUT02920
DIST     DC    CL8'DISTRICT'                                                AUT02930
         END                                                                AUT02940
/*                                                                          AUT02950
```

```
//STEP4 EXEC FCRTHCL,PARM.FORT='OPT=2'                                  AUT02960
//FORT.SYSIN DD *                                                       AUT02970
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT02980
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR             AUT02990
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT03000
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT03010
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                AUT03020
      COMMON/$PROST/IGAP,PROMAM,DOPMAT(5,25,3),BESPRO(60,26,3)          AUT03030
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                AUT03040
      COMMON/C/IDISTR,IPRTCT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT03050
     1E5(31),LINE6(31),IQUIT                                            AUT03060
      INTEGER*4 XYEAR,XSYEAR                                            AUT03070
C-----------------------------------------------------------------------AUT03080
C   THIS SECTION CALLS ASSEMBLER SUBROUTINES WHICH READ AND CENTER THE  |AUT03090
C   3 TITLES FOR THE OUTPUT PAGES AND READ THE 2 LINES OF FOOTNOTES     |AUT03100
C   USED ON EACH OUTPUT PAGE.  (SEE SUBROUTINES READ AND CENTER FOR     |AUT03110
C   FURTHER DETAILS.)  THE INPUT CARDS FOR THIS SECTION ARE INSERTED    |AUT03120
C   ONLY ONCE PER RUN.                                                  |AUT03130
C-----------------------------------------------------------------------AUT03140
   77 CALL CENTER(LINE1)                                                AUT03150
      CALL CENTER(LINE2)                                                AUT03160
      CALL CENTER(LINE3)                                                AUT03170
      CALL READ(LINE5)                                                  AUT03180
      CALL READ(LINE6)                                                  AUT03190
C-----------------------------------------------------------------------AUT03200
C   PUT IN THE NEXT CARD AND ALL THAT FOLLOW ONCE FOR EACH DISTRICT.    |AUT03210
C-----------------------------------------------------------------------AUT03220
C                          ****DEFINITIONS****                          |AUT03230
C                                                                       |AUT03240
C       N = NUMBER OF BASE YEARS OF DATA INPUT (MAX=40)                 |AUT03250
C       K = NUMBER OF YEARS TO MAKE PROJECTIONS FOR (MAX=60)            |AUT03260
C       L = NUMBER OF COUNTIES FOR THIS DISTRICT (MAX=25)               |AUT03270
C    IGAP = THIS CONTROLS THE MINIMUM DIFFERENCE IN RSQ A PROJECTION    |AUT03280
C           MUST HAVE TO REPLACE ANOTHER IN THE SET OF BEST PROJEC-     |AUT03290
C           TIONS.  IF LEFT BLANK, THIS DEFAULTS TO A DIFFERENCE OF     |AUT03300
C           0.0.  IF YOU SET THE VALUE LARGER, IT REDUCES WORK FOR THE  |AUT03310
C           PROGRAM VIA REDUCING THE NUMBER OF PROJECTIONS THAT MUST BE |AUT03320
C           SHUFFLED IN AND OUT OF THE BEST PROJECTIONS MATRIX BY SUB-   |AUT03330
C           ROUTINE PROSTO.  ALSO, THE PROJECTIONS WILL BE SPACED OUT    |AUT03340
C           IN THAT PROJECTIONS WITH AN RSQ LESS THAN IGAP APART WILL    |AUT03350
C           NOT EXIST IN THE BEST PROJECTIONS MATRIX (UNLESS ONE OF THE  |AUT03360
C           FIRST THREE PROJECTIONS RUN AND ITS RSQ EXCEEDS THAT OF ALL  |AUT03370
C           SUBSEQUENT PROJECTIONS.)                                    |AUT03380
C   IZERO = THIS TURNS ON CHECKING FOR ZEROS OR MISSING DATA IN THE     |AUT03390
C           INPUT DATA.  THIS WILL ONLY WORK IF NO 2 CONSEQUTIVE YEARS  |AUT03400
C           ARE MISSING DATA.  (SEE SUBROUTINE MINVAL FOR FURTHER       |AUT03410
C           DETAILS.)  TURN THIS FEATURE ON BY INPUT OF YES IN THE      |AUT03420
C           CORRECT COLUMNS--DEFAULT IS NO CHECKING.                    |AUT03430
C  IDISTR = THE THREE CHARACTERS READ FOR THE VALUE OF IDISTR ARE USED  |AUT03440
C           ON EACH PAGE OF OUTPUT TO IDENTIFY WHICH DISTRICT THE PRO-  |AUT03450
C           JECTIONS ARE FOR.                                           |AUT03460
C-----------------------------------------------------------------------AUT03470
   78 READ(5,90,END=98) N,K,L,IGAP,IZERO,IDISTR                         AUT03480
   90 FORMAT(4(2X,I2),1X,2A3)                                           AUT03490
      IF(N.GT.40) GO TO 21                                              AUT03500
      IF(K.GT.60) GO TO 23                                              AUT03510
      IF(L.GT.25) GO TO 25                                              AUT03520
C-----------------------------------------------------------------------AUT03530
C   THE NAME OF EACH COUNTY IN THE DISTRICT IS READ AT THIS POINT.  SEE |AUT03540
C   SUBROUTINE COUNTY FOR FURTHER DETAILS.) THE CARD(S) WITH THESE      |AUT03550
```

```
C   NAMES SHOULD BE INCLUDED FOR EACH DISTRICT.                          |AUT03560
C-------------------------------------------------------------------------AUT03570
      CALL CCUNTY(LINE4)                                                  AUT03580
      IF(L.LE.12) GO TO 6                                                 AUT03590
      IF(L.LE.24) GO TO 5                                                 AUT03600
C-------------------------------------------------------------------------AUT03610
C   THESE READ FORMATS FCR CCUNTY DATA MATCH THCSE OF THE MASTER          |AUT03620
C   PROJECTICN PROGRAM FCR CCNDENSED DATA.                                |AUT03630
C-------------------------------------------------------------------------AUT03640
      DO 91 I=1,N                                                         AUT03650
   91 READ(5,4) XYEAR(I),(YBASE(I,ICOL2),ICOL2=1,L)                       AUT03660
    4 FORMAT(I4,4X,12F6.C/8X,12F6.0/8X,F6.0)                              AUT03670
      GO TO 9                                                             AUT03680
    5 DO 89 I=1,N                                                         AUT03690
   89 READ(5,7) XYEAR(I),(YBASE(I,ICOL2),ICOL2=1,L)                       AUT03700
    7 FORMAT(I4,4X,12F6.0/8X,12F6.0)                                      AUT03710
      GO TO 9                                                             AUT03720
    6 DO 88 I=1,N                                                         AUT03730
   88 READ(5,8) XYEAR(I),(YBASE(I,ICOL2),ICOL2=1,L)                       AUT03740
    8 FORMAT(I4,4X,12F6.C)                                                AUT03750
C-------------------------------------                                    AUT03760
C   WRITE TITLE PAGE FCR EACH DISTRICT. |                                 AUT03770
C-------------------------------------                                    AUT03780
    9 DO 710 I=1,8                                                        AUT03790
  710 WRITE(6,702)                                                        AUT03800
  702 FORMAT('-')                                                         AUT03810
      WRITE(6,706) (LINE3(I),I=1,33)                                      AUT03820
  706 FORMAT(1X,33A4)                                                     AUT03830
      WRITE(6,707)                                                        AUT03840
  707 FORMAT('0')                                                         AUT03850
      WRITE(6,703)                                                        AUT03860
  703 FORMAT(33X,69('*'))                                                 AUT03870
      WRITE(6,704)                                                        AUT03880
  704 FORMAT(33X,'*',67X,'*')                                             AUT03890
      WRITE(6,704)                                                        AUT03900
      WRITE(6,705) IDISTR,L,N,K                                           AUT03910
  705 FORMAT(33X,'*',2X,'DISTRICT',1X,A3,3X,I2,1X,'COUNTIES',3X,I2,1X,'BAUT03920
     *ASE YEARS',3X,'PROJECTED',1X,I2,1X,'YEARS',2X,'*')                  AUT03930
      WRITE(6,704)                                                        AUT03940
      WRITE(6,704)                                                        AUT03950
      WRITE(6,703)                                                        AUT03960
      CALL MINVAL(IZERO)                                                  AUT03970
C-----------------------------------------                                AUT03980
C   INITIALIZE MATRICES DOPMAT AND BESPRO FOR |                           AUT03990
C   FUTURE USE IN SUBRCUTINE PROSTO.          |                           AUT04000
C-----------------------------------------                                AUT04010
      DO 14 ICOL2=1,L                                                     AUT04020
      DO 15 J=1,K                                                         AUT04030
      BESPRO(J,ICOL2,3)=0.0                                               AUT04040
      BESPRO(J,ICOL2,2)=0.0                                               AUT04050
   15 BESPRO(J,ICOL2,1)=0.0                                               AUT04060
      DO 14 I=1,5                                                         AUT04070
      DOPMAT(I,ICOL2,3)=0.0                                               AUT04080
      DOPMAT(I,ICOL2,2)=0.0                                               AUT04090
   14 DOPMAT(I,ICOL2,1)=0.0                                               AUT04100
C-------------------------------------------------------------------      AUT04110
C   START REGRESSION SEQUENCE FOR EACH CCUNTY.  (SEE SUBROUTINES |        AUT04120
C   REGRES,PROJCT,AND EXPFLW FOR FURTHER DETAILS.)               |        AUT04130
C-------------------------------------------------------------------      AUT04140
      DO 93 ICOL2=1,L                                                     AUT04150
```

```
      IREGTP=1                                                          AUT04160
      PRONAM=99                                                         AUT04170
      CALL REGRES                                                       AUT04180
      IREGTP=2                                                          AUT04190
      PRONAM=98                                                         AUT04200
      CALL REGRES                                                       AUT04210
      IREGTP=3                                                          AUT04220
      CALL EXPFLW                                                       AUT04230
   93 CONTINUE                                                          AUT04240
      DO 500 I=1,K                                                      AUT04250
  500 XSYEAR(I)=XYEAR(N)+I                                              AUT04260
      CALL SUMDIS                                                       AUT04270
      CALL SUMMRY(L)                                                    AUT04280
      CALL IDIST                                                        AUT04290
      WRITE(6,70)                                                       AUT04300
   70 FORMAT('1')                                                       AUT04310
      GO TO 78                                                          AUT04320
C-------------------------                                            AUT04330
C   ERROR MESSAGE SECTION. |                                          AUT04340
C-------------------------                                            AUT04350
   21 WRITE(6,27) IDISTR,N                                              AUT04360
   27 FORMAT('0','***ERROR***THE VALUE OF N FOR DISTRICT ',A3,' WAS ',I2AUT04370
     *,' WHICH EXCEEDS'/'THE MAXIMUM FOR THIS PROGRAM OF 40.  TRY AGAIN AUT04380
     *MATE.')                                                           AUT04390
      GO TO 98                                                          AUT04400
   23 WRITE(6,22) IDISTR,K                                              AUT04410
   22 FORMAT('0','***ERROR***THE VALUE OF K FOR DISTRICT ',A3,' WAS ',I2AUT04420
     *,' WHICH EXCEEDS'/'THE MAXIMUM FOR THIS PROGRAM OF 60.  SORRY ABOUAUT04430
     *T THAT.')                                                         AUT04440
      GO TO 98                                                          AUT04450
   25 WRITE(6,24) IDISTR,L                                              AUT04460
   24 FORMAT('0','***ERROR***THE VALUE OF L FOR DISTRICT ',A3,' WAS ',I2AUT04470
     *,' WHICH EXCEEDS'/'THE MAXIMUM FOR THIS PROGRAM OF 25.  TRY ONE MOAUT04480
     *RE TIME.')                                                        AUT04490
   98 STOP                                                              AUT04500
      END                                                               AUT04510
      SUBROUTINE REGRES                                                 AUT04520
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT04530
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR             AUT04540
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT04550
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT04560
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                AUT04570
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                AUT04580
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)          AUT04590
      COMMON/$EXP/LINRSQ                                                AUT04600
      REAL*4 LINRSQ                                                     AUT04610
      INTEGER*4 XYEAR,XSYEAR                                            AUT04620
C-----------------------------------------------------------------------AUT04630
C   THIS SECTION TRANSFORMS XBASE ACCORDING TO THE PROJECTION TYPE AS  |AUT04640
C   SPECIFIED BY IREGTP AND STORES THE RESULTS IN XSBASE.  (SEE SUB-   |AUT04650
C   ROUTINE PROJCT FOR VARIABLE DEFINITIONS.)                         |AUT04660
C-----------------------------------------------------------------------AUT04670
      GO TO (620,621,622),IREGTP                                        AUT04680
  622 DO 708 I=1,N                                                      AUT04690
  708 XSBASE(I)=XBASE(I)**EXPON                                         AUT04700
      PRONAM=(EXPON*10)+.0001                                           AUT04710
      GO TO 623                                                         AUT04720
  621 DO 700 I=1,N                                                      AUT04730
  700 XSBASE(I)=DLOG(XBASE(I))                                          AUT04740
      GO TO 623                                                         AUT04750
```

```
  620 DO 701 I=1,N                                                        AUT04760
  701 XSBASE(I)=XBASE(I)                                                  AUT04770
C-------------------------------------------                              AUT04780
C   SUBROUTINE PROJCT CALLED HERE PERFORMS THE |                         AUT04790
C   ACTUAL REGRESSIONS AND MAKES PROJECTIONS.  |                         AUT04800
C-------------------------------------------                              AUT04810
  623 CALL PROJCT                                                         AUT04820
  996 IF(IREGTP.EQ.1) LINRSQ=RSQ                                          AUT04830
C     IF(IREGTP.EQ.2) LOGRSQ=RSQ    (STATEMENT FOR FUTURE EXPANSION.)   AUT04840
C-------------------------------------                                    AUT04850
C   SUBROUTINE PROSTO CALLED HERE     |                                  AUT04860
C   STORES THE COMPLETED PROJECTIONS. |                                  AUT04870
C-------------------------------------                                    AUT04880
      CALL PROSTO                                                         AUT04890
      RETURN                                                              AUT04900
      END                                                                 AUT04910
      SUBROUTINE PROJCT                                                   AUT04920
C-----------------------------------------------------------------------AUT04930
C   THIS SUBROUTINE PROJECTS INPUT DATA ACCORDING TO THE VALUE OF      |AUT04940
C   IREGTP.  IF:  IREGTP =1  ; THE PROJECTION IS LINEAR                |AUT04950
C                        2  ; THE PROJECTION IS LOGARITHMIC            |AUT04960
C                        3  ; THE PROJECTION IS EXPONENTIAL            |AUT04970
C                                                                      |AUT04980
C   THE LOG PROJECTION IS OF THE FORM Y=A+B LOGX.  THE EXPONENTIAL     |AUT04990
C   PROJECTION IS OF THE FORM Y=A+BX**EXPON; WHERE EXPON IS INPUT TO   |AUT05000
C   THE SUBROUTINE AND LIES BETWEEN .4 AND 1.4, EXCLUDING 1.0.         |AUT05010
C-----------------------------------------------------------------------AUT05020
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                      AUT05030
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR               AUT05040
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                  AUT05050
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT05060
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                  AUT05070
C-----------------------------------------------------------------------AUT05080
C   PASS IN:  XSBASE(40),YBASE(40,26),L,N,K,IREGTP,ICOL2               |AUT05090
C                                                                      |AUT05100
C   PASS OUT: YSPROJ(60),ALPHA,BETA,YBAR,RSQ                           |AUT05110
C                                                                      |AUT05120
C   WHERE:  XBASE = LOW RANGE EQUIVALENTS OF XYEAR AS CONVERTED BY     |AUT05130
C                   SUBROUTINE MINVAL, THE X'S                         |AUT05140
C           YBASE = THE Y'S, ONE COLUMN PER COUNTY                     |AUT05150
C          YSPROJ = THE PROJECTED Y'S, THE SAME VECTOR IS REUSED FOR   |AUT05160
C                   ALL COUNTIES AFTER THE RESULTS ARE TRANSFERRED TO  |AUT05170
C                   TO MATRIX BESPRO VIA SUBROUTINE PROSTO             |AUT05180
C               L = THE NUMBER OF COUNTIES                             |AUT05190
C               N = THE NUMBER OF YEARS OF BASE DATA                   |AUT05200
C               K = THE NUMBER OF YEARS TO APPLY THE REGRESSION EQUA-  |AUT05210
C                   TION TO AND THUS MAKE PROJECTIONS FOR              |AUT05220
C          IREGTP = KEY TO THE TYPE OF REGRESSION TO BE PERFORMED      |AUT05230
C           EXPON = EXPONENT TO BE USED IN AN EXPONENTIAL REGRESSION   |AUT05240
C           ICOL2 = POINTER SHOWING WHICH OF THE COLUMNS IN YSBASE     |AUT05250
C                   (WHICH COUNTY) IS BEING REGRESSED OR PRINTED       |AUT05260
C          XSBASE = VECTOR CONTAINING TRANSFORMED X'S, SUCH AS LOG X   |AUT05270
C          XSPROJ = VECTOR CONTAINING PROJECTED X'S, USED TO CALCULATE |AUT05280
C                   THE PROJECTED Y'S BY APPLYING THE REGRESSION       |AUT05290
C                   EQUATION                                           |AUT05300
C-----------------------------------------------------------------------AUT05310
      SUMXY=0.000000000                                                   AUT05320
      SUMSQX=0.000000000                                                  AUT05330
      SUMX=0.000000000                                                    AUT05340
      SUMY=0.000000000                                                    AUT05350
```

```
      SUMSQY=0.00000000                                                  AUT05360
C-------------------------------                                         AUT05370
C   FIT TREND LINE TO BASE DATA.|                                       AUT05380
C-------------------------------                                         AUT05390
  623 DO 600 I=1,N                                                      AUT05400
      SUMY=SUMY+YBASE(I,ICOL2)                                          AUT05410
      SUMX=SUMX+XSBASE(I)                                               AUT05420
      SUMSQX=SUMSQX+XSBASE(I)**2                                        AUT05430
      SUMSQY=SUMSQY+YBASE(I,ICOL2)**2                                   AUT05440
  600 SUMXY=SUMXY+YBASE(I,ICOL2)*XSBASE(I)                              AUT05450
      SUMXSQ=SUMX**2                                                    AUT05460
      XBAR=SUMX/N                                                       AUT05470
      YBAR=SUMY/N                                                       AUT05480
      TOP=SUMXY-((SUMX*SUMY)/N)                                         AUT05490
      BETA=TOP/(SUMSQX-SUMXSQ/N)                                        AUT05500
      ALPHA=YBAR-(BETA*XBAR)                                            AUT05510
C-----------------------------                                          AUT05520
C   PROJECT BASE DATA K YEARS.|                                         AUT05530
C-----------------------------                                          AUT05540
      GO TO (624,625,626),IREGTP                                        AUT05550
  624 DO 709 I=1,K                                                      AUT05560
      XSPROJ(I)=XBASE(N)+I                                              AUT05570
  709 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)                                    AUT05580
      GO TO 640                                                         AUT05590
  626 DO 710 I=1,K                                                      AUT05600
      XSPROJ(I)=(XBASE(N)+I)**EXPON                                     AUT05610
  710 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)                                    AUT05620
      GO TO 640                                                         AUT05630
  625 DO 601 I=1,K                                                      AUT05640
      XSPROJ(I)=DLOG(XBASE(N)+I)                                        AUT05650
  601 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)                                    AUT05660
C-----------------------------------------                              AUT05670
C   THIS SECTION COMPUTES THE VALUE OF RSQ.|                            AUT05680
C-----------------------------------------                              AUT05690
  640 TOP=(SUMXY/N-XBAR*YBAR)                                           AUT05700
      BOT=DSQRT((SUMSQX/N-XBAR**2)*(SUMSQY/N-YBAR**2))                  AUT05710
      RSQ=(TOP/BOT)**2                                                  AUT05720
  998 RETURN                                                            AUT05730
      END                                                               AUT05740
      SUBROUTINE MINVAL(IZERO)                                          AUT05750
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT05760
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR             AUT05770
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT05780
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT05790
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                AUT05800
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                AUT05810
      INTEGER*4 XYEAR,XSYEAR                                            AUT05820
      INTEGER*4 YES/'YES'/                                              AUT05830
C-------------------------------------------------------------------------AUT05840
C   THIS SECTION REPLACES ZEROS IN THE BASE DATA.. IF THE MISSING      |AUT05850
C   DATA IS THE FIRST YEAR'S DATA, IT IS REPLACED WITH THE SECOND      |AUT05860
C   YEAR'S DATA MINUS THE ABSOLUTE VALUE OF THE DIFFERENCE BETWEEN THE |AUT05870
C   SECOND AND THIRD YEAR'S DATA.  IF THIS IS LESS THAN ZERO, THE FIRST|AUT05880
C   YEAR IS LEFT AS ZERO.                                              |AUT05890
C   IF THE MISSING DATA IS THE LAST YEAR, IT IS REPLACED WITH THE NEXT |AUT05900
C   TO LAST YEAR'S DATA PLUS THE DIFFERENCE BETWEEN THE SECOND TO LAST |AUT05910
C   AND THE THIRD TO LAST YEAR'S DATA.                                 |AUT05920
C   IF ANY OTHER YEAR IS MISSING, THE AVERAGE OF THE YEARS BEFORE AND  |AUT05930
C   AFTER IT IS USED IN PLACE OF THE MISSING DATA.                     |AUT05940
C   IF TWO YEAR'S DATA IN A ROW ARE MISSING, THE USER MUST CALCULATE   |AUT05950
```

```
C   ONE OF THEM AS THIS ROUTINE WILL ONLY REPLACE THE SECOND ONE.       |AUT05960
C-------------------------------------------------------------------------AUT05970
C***READ IN IZERO IN MAIN/ IF YES,TEST FOR ZEROS, IF NO SKIP THIS SECT.|AUT05980
C-------------------------------------------------------------------------AUT05990
      IF(IZERO.NE.YES) GO TO 12                                          AUT06000
      DO 5 J=1,L                                                         AUT06010
      DO 5 I=1,N                                                         AUT06020
      IF(YBASE(I,J).EQ.0) GO TO 8                                        AUT06030
      GO TO 5                                                            AUT06040
    8 IF(I.EQ.1) GO TO 9                                                 AUT06050
      IF(I.EQ.N) GO TO 10                                                AUT06060
      YBASE(I,J)=(YBASE(I-1,J)+YBASE(I+1,J))/2                           AUT06070
      GO TO 5                                                            AUT06080
    9 YBASE(1,J)=YBASE(2,J)-(YBASE(3,J)-YBASE(2,J))                      AUT06090
      IF(YBASE(1,J).LT.0) YBASE(1,J)=0                                   AUT06100
      GO TO 5                                                            AUT06110
   10 YBASE(I,J)=YBASE(I-1,J)+(YBASE(I-1,J)-YBASE(I-2,J))                AUT06120
      IF(YBASE(N,J).LT.0) YBASE(N,J)=0                                   AUT06122
    5 CONTINUE                                                           AUT06130
C-------------------------------------------------------------------------AUT06140
C   THIS SECTION TRANSFORMS THE INPUT YEARS (XYEAR) INTO INTEGERS       |AUT06150
C   STARTING WITH 1 SO THE REGRESSION INPUT VALUES WILL BE SCALED COR-  |AUT06160
C   RECTLY.  THE SCALED RESULTS ARE PUT INTO XBASE.                     |AUT06170
C-------------------------------------------------------------------------AUT06180
   12 MINXYR=XYEAR(1)                                                    AUT06190
      DO 6 I=1,N                                                         AUT06200
      IF(XYEAR(I).GT.MINXYR) GO TO 6                                     AUT06210
      MINXYR=XYEAR(I)                                                    AUT06220
    6 CONTINUE                                                           AUT06230
      IREDUC=MINXYR-1                                                    AUT06240
      DO 7 I=1,N                                                         AUT06250
    7 XBASE(I)=XYEAR(I)-IREDUC                                           AUT06260
      RETURN                                                             AUT06270
      END                                                                AUT06280
      SUBROUTINE IDIST                                                   AUT06290
C-------------------------------------------------------------------------AUT06300
C   THIS SUBROUTINE CONTROLS THE PRINTING OF OUTPUT VIA THE CALLING     |AUT06310
C   OF SUBROUTINES ITITLE,IDATA, AND ISTAT.                             |AUT06320
C-------------------------------------------------------------------------AUT06330
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                     AUT06340
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR              AUT06350
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                 AUT06360
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT06370
     *ON,RSQ,YSPROJ(60),ICCL2,IREGTP,K,N                                 AUT06380
      COMMON/$PROST/IGAP,PROMAM,DOPMAT(5,25,3),BESPRC(60,26,3)           AUT06390
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                 AUT06400
      COMMON/C/IDISTR,IPRTDT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT06410
     1E5(31),LINE6(31),IQUIT                                             AUT06420
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPROST,ISKIP                    AUT06430
      COMMON/$PAGE/NOPAGE                                                AUT06440
      INTEGER XYEAR,XSYEAR                                               AUT06450
C-------------------------------------------------------------------------AUT06460
C       K = NUMBER OF YEARS OF DATA TO PROJECT. (MAX=60)                |AUT06470
C       L = NUMBER OF COUNTIES PER DISTRICT (MAX=25)                    |AUT06480
C       N = NUMBER OF BASE YEARS OF DATA (MAX=40)                       |AUT06490
C   IPAGE = NUMBER OF PAGES IN LENGTH 1,2,OR 3                          |AUT06500
C   IWIDTH= NUMBER OF PAGES IN WIDTH  1 OR 2                            |AUT06510
C   NOYEAR= 0 = PRINT STATISTICS AT THE BOTTOM OF THE LAST PAGE.        |AUT06520
C           1 = PRINT STATISTICS AT THE TOP OF THE LAST PAGE.           |AUT06530
C   IBSEC = NUMBER OF SPACES BETWEEN 5 YEAR GROUPS IN THE BASE DATA     |AUT06540
```

```
C    IPSEC = NUMBER OF SPACES BETWEEN 5 YEAR GROUPS IN THE PROJECTED     |AUT06550
C            DATA                                                        |AUT06560
C    ITOTLN= TOTAL NUMBER OF OUTPUT LINES EXCLUSIVE OF TITLE BLOCKS AND  |AUT06570
C            STATISTICS                                                  |AUT06580
C    NFLINE= FIRST LINE OF BASE DATA TO BE PRINTED WHEN IDATA IS CALLED  |AUT06590
C    KFLINE= FIRST LINE OF THE PROJECTED DATA TO BE PRINTED WHEN         |AUT06600
C            IDATA IS CALLED                                             |AUT06610
C    IPRTDT= SIGNALS ITITLE TO OMIT PRINTING THE YEAR IN THE TITLE BLOCK|AUT06620
C    ISKIP = SIGNAL TO IDATA THAT BASE DATA WAS ALREADY PRINTED ON A     |AUT06630
C            PREVIOUS CALL AND TO SKIP PRINTING IT THIS TIME             |AUT06640
C    IPROST= PROJECTION SET NUMBER FROM 1 TO 3 WHERE 1 IS THE BEST       |AUT06650
C            (BASED ON RSQ'S)                                            |AUT06660
C    ISTART= NUMBER OF THE COUNTY THAT THE OUTPUT PAGE STARTS WITH.      |AUT06670
C            (COLUMN NO. OF MATRICES BESPRO, OR DOPMAT WHEN USED IN      |AUT06680
C            SUBROUTINE ISTAT.)                                          |AUT06690
C      IEND= NUMBER OF COLUMN IN BESPRO THAT OUTPUT PAGE ENDS WITH.      |AUT06700
C            (= L+1 ON THE LAST PAGE IN WIDTH, TO ALLOW FOR THE DIS-     |AUT06710
C            TOTAL COLUMN.                                               |AUT06720
C     IQUIT= NUMBER OF THE COLUMN OF MATRIX DOPMAT THE OUTPUT PAGE'S     |AUT06730
C            STATISTIC BLOCK ENDS WITH.  (= L ON THE LAST PAGE IN WIDTH  |AUT06740
C            AS THE DISTRICT COLUMN HAS NO STATISTICS.                   |AUT06750
C    NOPAGE= FLAG FOR ITITLE NOT TO DO PAGING AS THE LAST PAGE OF OUT-   |AUT06760
C            PUT WAS FILLED AND CAUSED PAGING BY THE SYSTEM.             |AUT06770
C-------------------------------------------------------------------------AUT06780
      NOPAGE=0                                                           AUT06790
      IBSEC=N/5+.4                                                       AUT06800
      IPSEC=K/5+.4                                                       AUT06810
      ITOTLN=IBSEC+N+1+IPSEC+K                                           AUT06820
      IF(ITOTLN.LE.38) GO TO 1                                           AUT06830
      IF(ITOTLN.LE.48) GO TO 2                                           AUT06840
      IF(ITOTLN.LE.86) GO TO 3                                           AUT06850
      IF(ITOTLN.LE.96) GO TO 4                                           AUT06860
      GO TO 5                                                            AUT06870
C-------------------------------------------------------------------------AUT06880
C    IPAGE = 60 LINES   TITLE BLOCK = 12 LINES   STATISTICS = 10 LINES  |AUT06890
C    48 LINES/PAGE+TITLE BLOCK  38 LINES/PAGE+TITLE BLOCK+STATISTICS    |AUT06900
C                                                                       |AUT06910
C    ITOTLN.LE.38 = PRINT 1 PAGE WITH STATISTICS AT BOTTOM              |AUT06920
C    ITOTLN.LE.48 = PRINT 2ND PAGE WITH ONLY TITLE BLOCK (WITHOUT YEAR  |AUT06930
C                   IN IT) AND STATISTICS.                              |AUT06940
C    ITOTLN.LE.86 = PRINT 2 PAGES WITH STATISTICS AT BOTTOM OF 2ND PAGE |AUT06950
C    ITOTLN.LE.96 = PRINT 3RD PAGE WITH ONLY TITLE BLOCK (WITHOUT YEAR  |AUT06960
C                   IN IT) AND STATISTICS.                              |AUT06970
C    TO MAX OF 121= PRINT 3 PAGES WITH STATISTICS AT BOTTOM OF 3RD PAGE |AUT06980
C-------------------------------------------------------------------------AUT06990
    1 IPAGE=1                                                            AUT07000
      NOYEAR=0                                                           AUT07010
      GO TO 50                                                           AUT07020
    2 IPAGE=2                                                            AUT07030
      NOYEAR=1                                                           AUT07040
      GO TO 50                                                           AUT07050
    3 IPAGE=2                                                            AUT07060
      NOYEAR=0                                                           AUT07070
      GO TO 50                                                           AUT07080
    4 IPAGE=3                                                            AUT07090
      NOYEAR=1                                                           AUT07100
      GO TO 50                                                           AUT07110
    5 IPAGE=3                                                            AUT07120
      NOYEAR=0                                                           AUT07130
   50 IF(L.LE.12) GO TO 51                                               AUT07140
```

```
      IWIDTH=2                                                          AUT07150
      GO TO 32                                                          AUT07160
   51 IWIDTH=1                                                          AUT07170
C-------------------------------------------------                      AUT07180
C   DO THE FOLLOWING ONCE FOR EACH PROJECTION SET |                     AUT07190
C   AND ONCE FOR EACH PAGE IN WIDTH.              |                     AUT07200
C-------------------------------------------------                      AUT07210
   32 DO 30 IPROST=1,3                                                  AUT07220
      DO 30 I=1,IWIDTH                                                  AUT07230
      NFLINE=1                                                          AUT07240
      KFLINE=1                                                          AUT07250
      IPRTDT=0                                                          AUT07260
      IF(IWIDTH.EQ.2) GO TO 10                                          AUT07270
      ISTART=1                                                          AUT07280
      IQUIT=L                                                           AUT07290
      IEND=L+1                                                          AUT07300
      GO TO 11                                                          AUT07310
   10 IF(I.EQ.2) GO TO 12                                               AUT07320
      ISTART=1                                                          AUT07330
      IEND=13                                                           AUT07340
      IQUIT=13                                                          AUT07350
      GO TO 11                                                          AUT07360
   12 ISTART=14                                                         AUT07370
      IEND=L+1                                                          AUT07380
      IQUIT=L                                                           AUT07390
   11 ISKIP=0                                                           AUT07400
      IF(IPAGE.EQ.3) GO TO 6                                            AUT07410
      IF(IPAGE.EQ.2) GO TO 7                                            AUT07420
C--------------------                                                   AUT07430
C   IPAGE=1,NCYEAR=0|                                                   AUT07440
C--------------------                                                   AUT07450
      CALL ITITLE                                                       AUT07460
      CALL IDATA                                                        AUT07470
      CALL ISTAT                                                        AUT07480
      GO TO 30                                                          AUT07490
    7 IF(NCYEAR.EQ.0) GO TO 8                                           AUT07500
C--------------------                                                   AUT07510
C   IPAGE=2,NCYEAR=1|                                                   AUT07520
C--------------------                                                   AUT07530
      CALL ITITLE                                                       AUT07540
      CALL IDATA                                                        AUT07550
C-----------------------------------------------------------------------AUT07552
C   DO NOT PRINT STATISTIC BLOCK IF DISTRICT IS THE ONLY COLUMN ON THE |AUT07554
C   PAGE AND ONLY THE DISTRICT'S STAT BLOCK WOULD BE PRINTED.          |AUT07555
C-----------------------------------------------------------------------AUT07556
      IF(IEND.EQ.14) GO TO 30                                           AUT07557
      IPRTDT=1                                                          AUT07560
      CALL ITITLE                                                       AUT07570
      CALL ISTAT                                                        AUT07580
      GO TO 30                                                          AUT07590
C--------------------                                                   AUT07600
C   IPAGE=2,NCYEAR=0|                                                   AUT07610
C--------------------                                                   AUT07620
    8 CALL ITITLE                                                       AUT07630
      CALL IDATA                                                        AUT07640
      CALL ITITLE                                                       AUT07650
      CALL IDATA                                                        AUT07660
      CALL ISTAT                                                        AUT07670
      GO TO 30                                                          AUT07680
C--------------------                                                   AUT07690
```

```
C   IPAGE=3,NOYEAR=1|                                                    AUT07700
C--------------------                                                    AUT07710
    6 IF(NOYEAR.EQ.0) GO TO 9                                            AUT07720
      CALL ITITLE                                                        AUT07730
      CALL IDATA                                                         AUT07740
      CALL ITITLE                                                        AUT07750
      CALL IDATA                                                         AUT07760
      IF(IEND.EQ.14) GO TO 30                                            AUT07762
      IPRTCT=1                                                           AUT07770
      CALL ITITLE                                                        AUT07780
      CALL ISTAT                                                         AUT07790
      GO TO 30                                                           AUT07800
C--------------------                                                    AUT07810
C   IPAGE=3,NOYEAR=0|                                                    AUT07820
C--------------------                                                    AUT07830
    9 CALL ITITLE                                                        AUT07840
      CALL IDATA                                                         AUT07850
      CALL ITITLE                                                        AUT07860
      CALL IDATA                                                         AUT07870
      CALL ITITLE                                                        AUT07880
      CALL IDATA                                                         AUT07890
      CALL ISTAT                                                         AUT07900
   30 CONTINUE                                                           AUT07910
      RETURN                                                             AUT07920
      END                                                                AUT07930
      SUBROUTINE ITITLE                                                  AUT07940
C-------------------------------------------------------------------------AUT07950
C   THIS SUBROUTINE PRINTS THE TITLE BLOCK ON EACH PAGE OF OUTPUT.      |AUT07960
C-------------------------------------------------------------------------AUT07970
      COMMON/C/IDISTR,IPRTCT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT07980
     1E5(31),LINE6(31),IQUIT                                             AUT07990
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPROST,ISKIP                    AUT08000
      COMMON/$PAGE/NOPAGE                                                AUT08010
      COMMON/$SUM/ISUMRY                                                 AUT08020
      INTEGER=4 ONE/'1ST'/,TWO/'2ND'/,THREE/'3RD'/                       AUT08030
      IF(NOPAGE.EQ.1) GO TO 25                                           AUT08040
      WRITE(6,1) LINE1                                                   AUT08050
    1 FORMAT('1',33A4)                                                   AUT08060
      GO TO 20                                                           AUT08070
   25 WRITE(6,26) LINE1                                                  AUT08080
   26 FORMAT(1X,33A4)                                                    AUT08090
      NOPAGE=0                                                           AUT08100
   20 WRITE(6,2) LINE2                                                   AUT08110
    2 FORMAT('0',33A4)                                                   AUT08120
C---------------------------------                                       AUT08130
C   SELECT PROJECTION SET NUMBER. |                                      AUT08140
C---------------------------------                                       AUT08150
      IF(ISUMRY.EQ.1) GO TO 81                                           AUT08160
      GO TO (4,5,6),IPROST                                               AUT08170
    4 ISETNM=ONE                                                         AUT08180
      GO TO 7                                                            AUT08190
    5 ISETNM=TWO                                                         AUT08200
      GO TO 7                                                            AUT08210
    6 ISETNM=THREE                                                       AUT08220
    7 WRITE(6,3) ISETNM,IDISTR                                           AUT08230
    3 FORMAT('0',A3,' PROJECTION SET',102X,'DISTRICT ',A3)               AUT08240
      GO TO 80                                                           AUT08250
   81 ISUMRY=0                                                           AUT08260
      WRITE(6,82)                                                        AUT08270
   82 FORMAT(1X)                                                         AUT08280
```

```
   80 WRITE(6,8)                                                        AUT08290
    8 FORMAT(1X,132('-'))                                               AUT08300
      WRITE(6,9) LINE3                                                  AUT08310
    9 FORMAT(1X,33A4)                                                   AUT08320
      WRITE(6,10)                                                       AUT08330
   10 FORMAT('0',63X,'COUNTY')                                          AUT08340
      WRITE(6,8)                                                        AUT08350
C-------------------------------------------------------               AUT08360
C   IF ISTART > 14 = SECOND PAGE WIDTH OF OUTPUT SO USE  |              AUT08370
C   COUNTY NAMES IN SECOND HALF OF LINE4.                |              AUT08380
C-------------------------------------------------------               AUT08390
      IF(ISTART.GE.14) GO TO 15                                         AUT08400
C-------------------------------------------------------               AUT08410
C   WHEN IPRTDT = 1 DO NOT PRINT YEAR IN TITLE BLOCK.    |              AUT08420
C-------------------------------------------------------               AUT08430
      IF(IPRTDT.EQ.1) GO TO 11                                          AUT08440
      WRITE(6,12) (LINE4(I),I=1,30)                                     AUT08450
   12 FORMAT(4X,'YEAR',5X,30A4)                                         AUT08460
      GO TO 13                                                          AUT08470
   11 WRITE(6,14) (LINE4(I),I=1,30)                                     AUT08480
   14 FORMAT(13X,30A4)                                                  AUT08490
      GO TO 13                                                          AUT08500
   15 IF(IPRTDT.EQ.1) GO TO 16                                          AUT08510
      WRITE(6,12) (LINE4(I),I=31,60)                                    AUT08520
      GO TO 13                                                          AUT08530
   16 WRITE(6,14) (LINE4(I),I=31,60)                                    AUT08540
   13 WRITE(6,8)                                                        AUT08550
      RETURN                                                            AUT08560
      END                                                               AUT08570
      SUBROUTINE ISTAT                                                  AUT08580
C-------------------------------------------------------------------   AUT08590
C   THIS SUBROUTINE PRINTS THE STATISTICS BLOCK AT THE END OF EACH      AUT08600
C   SET OF PROJECTIONS.                                                |AUT08610
C-------------------------------------------------------------------   AUT08620
      COMMON/$PPOST/IGAP,PRONAM,DOPMAT(5,25,3),RESPRO(60,26,3)          AUT08630
      COMMON/C/IDISTR,IPRTDT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT08640
     1E5(31),LINE6(31),IQUIT                                            AUT08650
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPPOST,ISKIP                   AUT08660
      COMMON/$SUM/ISUMRY                                                AUT08680
      REAL*8 LINEAR/' LINEAR'/,LOG/'    LOG'/,OUTLIN(25)                AUT08690
      REAL*8 OUTPUT(16)/'0.1 EXP','0.2 EXP','0.3 EXP','0.4 EXP','0.5 EXPAUT08700
     1','0.6 EXP','0.7 EXP','0.8 EXP','0.9 EXP','1.0 EXP','1.1 EXP','1.2AUT08710
     2 EXP','1.3 EXP','1.4 EXP','1.5 EXP','1.6 EXP'/                    AUT08720
      WRITE(6,1)                                                        AUT08730
    1 FORMAT(1X,132('-'))                                               AUT08740
      IF(IEND.NE.14) GO TO 20                                           AUT08750
      IF(ISTART.NE.14) GO TO 20                                         AUT08760
      DO 26 J=1,5                                                       AUT08770
   26 WRITE(6,25)                                                       AUT08780
   25 FORMAT(1X)                                                        AUT08790
      GO TO 27                                                          AUT08800
   20 WRITE(6,2) (DOPMAT(1,J,IPROST),J=ISTART,IQUIT)                    AUT08810
    2 FORMAT(4X,'RSQ',6X,13(1X,F8.4))                                   AUT08820
      WRITE(6,3) (DOPMAT(2,J,IPROST),J=ISTART,IQUIT)                    AUT08830
    3 FORMAT(4X,'ALPHA',4X,13(1X,F8.2))                                 AUT08840
      WRITE(6,4) (DOPMAT(3,J,IPROST),J=ISTART,IQUIT)                    AUT08850
    4 FORMAT(4X,'BETA',5X,13(1X,F8.2))                                  AUT08860
      WRITE(6,5) (DOPMAT(4,J,IPROST),J=ISTART,IQUIT)                    AUT08870
    5 FORMAT(4X,'YBAR',5X,13(1X,F8.2))                                  AUT08880
C-------------------------------------------------------------------   AUT08890
```

```
C   THIS SECTION TRANSLATES PRONAM INTO A CHARACTER NAME.  THIS IS      |AUT08900
C   DONE BY USING THE VALUE OF PRONAM AS A POINTER TO AN ARRAY CON-     |AUT08910
C   TAINING THE CHARACTER NAME EQUIVALENTS OF THE NUMBERS.              |AUT08920
C-------------------------------------------------------------------------AUT08930
      DO 12 J=ISTART,IQUIT                                               AUT08940
      IF(DOPMAT(5,J,IPROST).EQ.99.0) GO TO 10                            AUT08950
      IF(DOPMAT(5,J,IPROST).EQ.98.0) GO TO 11                            AUT08960
      KEY=DOPMAT(5,J,IPROST)                                             AUT08970
      OUTLIN(J)=OUTPUT(KEY)                                              AUT08980
      GO TO 12                                                           AUT08990
   10 OUTLIN(J)=LINEAR                                                   AUT09000
      GO TO 12                                                           AUT09010
   11 OUTLIN(J)=LOG                                                      AUT09020
   12 CONTINUE                                                           AUT09030
      WRITE(6,6) (OUTLIN(J),J=ISTART,IQUIT)                              AUT09040
    6 FORMAT(4X,'PRO TYPE',2X,13(1X,A8))                                 AUT09050
   27 WRITE(6,1)                                                         AUT09060
      IF(ISUMRY.EQ.1) GO TO 43                                           AUT09070
      WRITE(6,7)                                                         AUT09080
    7 FORMAT(4X,'SOURCE:')                                               AUT09090
      WRITE(6,8) (LINE5(I),I=1,31)                                       AUT09100
    8 FORMAT(8X,32A4)                                                    AUT09110
      WRITE(6,8) (LINE6(I),I=1,31)                                       AUT09120
      RETURN                                                             AUT09130
   43 ISUMRY=0                                                           AUT09140
      RETURN                                                             AUT09150
      END                                                                AUT09160
      SUBROUTINE IDATA                                                   AUT09170
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                     AUT09180
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR              AUT09190
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                 AUT09200
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT09210
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                 AUT09220
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)           AUT09230
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                 AUT09240
      COMMON/C/IDISTR,IPRTDT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT09250
     1E5(31),LINE6(31),IQUIT                                             AUT09260
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPROST,ISKIP                    AUT09270
      COMMON/$PAGE/NOPAGE                                                AUT09280
      INTEGER XYEAR,XSYEAR                                               AUT09290
C-----------------------------------                                     AUT09300
C   WRITE BASE DATA WITH DISTRICT TOTALS|                               AUT09310
C-----------------------------------                                     AUT09320
      ICOUNT=0                                                           AUT09330
      IFLAG=0                                                            AUT09340
      IF(ISKIP.EQ.1) GO TO 29                                            AUT09350
      DO 21 I=NFLINE,N                                                   AUT09360
      ICOUNT=ICOUNT+1                                                    AUT09370
      IF(ICOUNT.GT.48) GO TO 26                                          AUT09380
      WRITE(6,20) XYEAR(I),(YBASE(I,J),J=ISTART,IEND)                    AUT09390
   20 FORMAT(4X,I4,5X,13(1X,F8.0))                                       AUT09400
      IFLAG=IFLAG+1                                                      AUT09410
      IF(IFLAG.LT.5) GO TO 21                                            AUT09420
      IFLAG=0                                                            AUT09430
      ICOUNT=ICOUNT+1                                                    AUT09440
      IF(ICOUNT.GT.48) GO TO 26                                          AUT09450
      WRITE(6,23)                                                        AUT09460
   23 FORMAT(1X)                                                         AUT09470
   21 CONTINUE                                                           AUT09480
C-------------------------------------------------------------------------AUT09490
```

```
C    IF IFLAG=0 IT MEANS A SPACE WAS JUST PUT IN SO ONLY PUT ONE MORE   |AUT09500
C    SPACE BETWEEN THE BASE AND PROJECTED DATA /OTHERWISE PUT TWO SPACES|AUT09510
C-------------------------------------------------------------------------AUT09520
      IF(IFLAG.EQ.0) GO TO 24                                             AUT09530
      ICOUNT=ICOUNT+1                                                     AUT09540
      IF(ICOUNT.GT.48) GO TO 27                                           AUT09550
      WRITE(6,23)                                                         AUT09560
      ICOUNT=ICOUNT+1                                                     AUT09570
      IF(ICOUNT.GT.48) GO TO 27                                           AUT09580
   24 WRITE(6,23)                                                         AUT09590
C-----------------------------------------                              AUT09600
C    WRITE PROJECTED DATA WITH DISTRICT TOTALS|                          AUT09610
C-----------------------------------------                              AUT09620
   29 IFLAG=0                                                             AUT09630
      DO 25 I=KFLINE,K                                                    AUT09640
      ICOUNT=ICOUNT+1                                                     AUT09650
      IF(ICOUNT.GT.48) GO TO 28                                           AUT09660
      WRITE(6,20) XSYEAR(I),(BESPRO(I,J,IPROST),J=ISTART,IEND)            AUT09670
      IFLAG=IFLAG+1                                                       AUT09680
      IF(IFLAG.LT.5) GO TO 25                                             AUT09690
      IFLAG=0                                                             AUT09700
      ICOUNT=ICOUNT+1                                                     AUT09710
      IF(ICOUNT.GT.48) GO TO 28                                           AUT09720
      WRITE(6,23)                                                         AUT09730
   25 CONTINUE                                                            AUT09740
      RETURN                                                              AUT09750
   26 KFLINE=I                                                            AUT09760
      NOPAGE=1                                                            AUT09770
      RETURN                                                              AUT09780
   27 ISKIP=1                                                             AUT09790
      NOPAGE=1                                                            AUT09800
      RETURN                                                              AUT09810
   28 ISKIP=1                                                             AUT09820
      NOPAGE=1                                                            AUT09830
      KFLINE=I                                                            AUT09840
      RETURN                                                              AUT09850
      END                                                                 AUT09860
      SUBROUTINE PROSTC                                                   AUT09870
C-------------------------------------------------------------------------AUT09880
C    THIS SUBROUTINE KEEPS THE BEST THREE PROJECTIONS AND THEIR          |AUT09890
C    ASSOCIATED DOPE VECTORS (CONTAINING RSQ, ALPHA, BETA, YBAR,         |AUT09900
C    AND PRONAM.)  THEY ARE STORED IN MATRIX DOPMAT (FOR THE DOPE VEC-   |AUT09910
C    TORS) AND IN MATRIX BESPRO FOR THE BEST THREE PROJECTIONS.  THE     |AUT09920
C    PROJECTIONS ARE FILED IN THE MATRIX IN ORDER OF THEIR RSQ, WITH     |AUT09930
C    THE PROJECTION WITH THE HIGHEST RSQ GOING INTO THE FIRST PLANE      |AUT09940
C    OF THE MATRIX AND SO ON.  NEW PROJECTIONS ARE COMPARED TO THOSE     |AUT09950
C    ON FILE IN THIS SUBROUTINE AND IF THEIR RSQ EXCEEDS THAT OF THE     |AUT09960
C    LOWEST RSQ-PROJECTION ON FILE, THE NEW PROJECTION IS STORED IN      |AUT09970
C    THE PLANE  BEFITTING ITS RSQ RANK AND ONE OF THE OTHER PROJEC-      |AUT09980
C    TIONS IS DROPPED AND THE OTHER PROJECTIONS AND THEIR ASSOCIATED     |AUT09990
C    DOPE VECTORS ARE MOVED OVER TO MAKE ROOM FOR THE NEW PROJECTION     |AUT10000
C    AND ITS DOPE VECTOR, AS DICTATED BY ITS RSQ RANK, SHOULD GO.        |AUT10010
C    **THUS THE PRINTING OF BESPRO WOULD GIVE THE BEST THREE PROJEC-     |AUT10020
C    TIONS FOR EACH COUNTY (COLUMN OF BESPRO).                           |AUT10030
C                                                                        |AUT10040
C    PRONAM = NAME OF THE TYPE OF PROJECTION JUST MADE BY REGRES.        |AUT10050
C             PRONAM IS REPRESENTED BY A NUMBER, THE KEY TO WHICH IS:    |AUT10060
C                                                                        |AUT10070
C    -------KEY-----------------PRONAM-----------------KEY--------      |AUT10080
C                                                                        |AUT10090
```

```
C            99 = LINEAR                                                    |AUT10100
C            98 = LOGARITHMIC                                               |AUT10110
C            OTHERWISE:  10 * EXPONENT    (EX: 1.4->PPONAM=14               |AUT10120
C-----------------------------------------------------------------------AUT10130
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                        AUT10140
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQX,XBAR                        AUT10150
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                    AUT10160
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXP AUT10170
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                    AUT10180
      COMMON/$PROST/IGAP,PPONAM,DOPMAT(5,25,3),BESPRO(60,26,3)              AUT10190
C-------------------------------------------------------------          AUT10200
C   IGAP IS THE MINIMUM REQUIRED RSQ DIFFERENCE BETWEEN RSQ'S.|           AUT10210
C       THE VALUE IS DETERMINED IN THE MAIN BY READING.       |           AUT10220
C-------------------------------------------------------------          AUT10230
      IF(RSQ.LE.(DOPMAT(1,ICOL2,3)+(IGAP/100))) GO TO 999                   AUT10240
      IF(RSQ.LE.(DOPMAT(1,ICOL2,2)+(IGAP/100))) GO TO 1                     AUT10250
      IF(RSQ.LE.(DOPMAT(1,ICOL2,1)+(IGAP/100))) GO TO 2                     AUT10260
C-------------------------------------------                             AUT10270
C   PUT NEW PROJECTION IN 1ST PROJECTION SET.|                            AUT10280
C-------------------------------------------                             AUT10290
      DO 3 I=1,5                                                            AUT10300
      DOPMAT(I,ICOL2,3)=DOPMAT(I,ICOL2,2)                                   AUT10310
    3 DOPMAT(I,ICOL2,2)=DOPMAT(I,ICOL2,1)                                   AUT10320
      DOPMAT(1,ICOL2,1)=RSQ                                                 AUT10330
      DOPMAT(2,ICOL2,1)=ALPHA                                               AUT10340
      DOPMAT(3,ICOL2,1)=BETA                                                AUT10350
      DOPMAT(4,ICOL2,1)=YBAR                                                AUT10360
      DOPMAT(5,ICOL2,1)=PRONAM                                              AUT10370
      DO 4 I=1,K                                                            AUT10380
      BESPRO(I,ICOL2,3)=BESPRO(I,ICOL2,2)                                   AUT10390
      BESPRO(I,ICOL2,2)=BESPRO(I,ICOL2,1)                                   AUT10400
    4 BESPRO(I,ICOL2,1)=YSPROJ(I)                                           AUT10410
      GO TO 999                                                             AUT10420
C-------------------------------------------                             AUT10430
C   PUT NEW PROJECTION IN 2ND PROJECTION SET.|                            AUT10440
C-------------------------------------------                             AUT10450
    2 DO 5 I=1,5                                                            AUT10460
    5 DOPMAT(I,ICOL2,3)=DOPMAT(I,ICOL2,2)                                   AUT10470
      DOPMAT(1,ICOL2,2)=RSQ                                                 AUT10480
      DOPMAT(2,ICOL2,2)=ALPHA                                               AUT10490
      DOPMAT(3,ICOL2,2)=BETA                                                AUT10500
      DOPMAT(4,ICOL2,2)=YBAR                                                AUT10510
      DOPMAT(5,ICOL2,2)=PRONAM                                              AUT10520
      DO 6 I=1,K                                                            AUT10530
      BESPRO(I,ICOL2,3)=BESPRO(I,ICOL2,2)                                   AUT10540
    6 BESPRO(I,ICOL2,2)=YSPROJ(I)                                           AUT10550
      GO TO 999                                                             AUT10560
C-------------------------------------------                             AUT10570
C   PUT NEW PROJECTION IN 3RD PROJECTION SET.|                            AUT10580
C-------------------------------------------                             AUT10590
    1 DOPMAT(1,ICOL2,3)=RSQ                                                 AUT10600
      DOPMAT(2,ICOL2,3)=ALPHA                                               AUT10610
      DOPMAT(3,ICOL2,3)=BETA                                                AUT10620
      DOPMAT(4,ICOL2,3)=YBAR                                                AUT10630
      DOPMAT(5,ICOL2,3)=PRONAM                                              AUT10640
      DO 7 I=1,K                                                            AUT10650
    7 BESPRO(I,ICOL2,3)=YSPROJ(I)                                           AUT10660
  999 RETURN                                                                AUT10670
      END                                                                   AUT10680
C-----------------------------------------------------------------------AUT10690
```

```
C   THIS SUBROUTINE CONTAINS THE LOGIC FOR PICKING THE BEST EXPONENTIAL|AUT10700
C   PROJECTION.  THIS IS DONE BY PICKING THE NEXT PROJECTION TO TRY    |AUT10710
C   BASED ON THE RSQ OF THE BEST PROJECTION MADE TO DATE.  THIS SUB-   |AUT10720
C   ROUTINE ASSUMES THAT IF THE LINEAR RSQ (LINRSQ) WAS NOT ABOVE 0.7  |AUT10730
C   THAT THAT THE TREND WOULD COME NEARER TO AN EXPONENTIAL PROJECTION |AUT10740
C   THAT WAS FURTHER AWAY FROM 1.0 AND THUS THE SUBROUTINE LOOKS FIRST |AUT10750
C   AT EXPONENTIAL PROJECTIONS WHERE THE EXPONENT IS FURTHER AWAY FROM |AUT10760
C   1.0.  OTHERWISE (WHEN THE RSQ OF THE LINEAR PROJECTION IS ABOVE    |AUT10770
C   .70) THE EXPONENTIAL PROJECTIONS WITH AN EXPONENT NEAR 1.0 ARE     |AUT10780
C   TRIED FIRST IN AN EFFORT TO SAVE THE NUMBER OF PROJECTIONS NEC-    |AUT10790
C   CESSARY.                                                           |AUT10800
C-------------------------------------------------------------------  |AUT10810
C                          SYMBOL DEFINITION                         |AUT10820
C                                                                     |AUT10830
C   LINRSQ = R SQUARE OF THE LINEAR REGRESSION.                       |AUT10840
C                                                                     |AUT10850
C   LOGRSQ = R SQUARE OF THE LOGARITHMIC REGRESSION.                  |AUT10860
C                                                                     |AUT10870
C   REGRES = SUBROUTINE WHICH PERFORMS THE FOLLOWING:                 |AUT10880
C            LINEAR      WHEN IREGTP = 1.0                          |AUT10890
C            LOGARITHMIC             = 2.0                          |AUT10900
C            EXPONENTIAL             = 3.0,                         |AUT10910
C            TYPES OF REGRESSION.  IREGTP IS THE KEY TO THE TYPE OF  |AUT10920
C            REGRESSION REGRES IS TO PERFORM.                       |AUT10930
C                                                                     |AUT10940
C   EXPON  = VALUE PASSED IN COMMON TO SUBROUTINE REGRES TO DETER-    |AUT10950
C            MINE THE EXPONENT USED IN EXPONENTIAL REGRESSION       |AUT10960
C                                                                     |AUT10970
C   ERSQ7  = RSQ FOR AN EXPONENTIAL PROJECTION USING THE EXPONENT     |AUT10980
C     ETC.    .7.  ERSQ12 MEANS THE EXPONENT 1.2.                   |AUT10990
C-------------------------------------------------------------------AUT11000
      SUBROUTINE EXPFLW                                             AUT11010
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR         AUT11020
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                            AUT11030
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                AUT11040
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT11050
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                            AUT11060
      COMMON/$EXP/LINRSQ                                            AUT11070
      REAL*4 LINRSQ                                                 AUT11080
      IF(LINRSQ.LT.0.70) GO TO 500                                  AUT11090
      EXPON=.90000000000                                            AUT11100
      CALL REGRES                                                   AUT11110
      ERSQ9=RSQ                                                     AUT11120
      IF(ERSQ9.LT.LINRSQ) GO TO 501                                 AUT11130
      EXPON=.70000000                                               AUT11140
      CALL REGRES                                                   AUT11150
      ERSQ7=RSQ                                                     AUT11160
      IF(ERSQ7.LT.ERSQ9) GO TO 502                                  AUT11170
      EXPON=.50000000                                               AUT11180
      CALL REGRES                                                   AUT11190
      ERSQ5=RSQ                                                     AUT11200
      IF(ERSQ5.LT.ERSQ7) GO TO 503                                  AUT11210
  505 EXPON=.40000000                                               AUT11220
      CALL REGRES                                                   AUT11230
      GO TO 888                                                     AUT11240
  503 EXPON=.60000000                                               AUT11250
      CALL REGRES                                                   AUT11260
      GO TO 888                                                     AUT11270
  510 EXPON=1.1000000                                               AUT11280
      CALL REGRES                                                   AUT11290
```

```
      ERSQ11=RSQ                                                        AUT11300
      IF(ERSQ11.LT.ERSQ9) GO TO 888                                     AUT11310
      GO TO 530                                                         AUT11320
  502 EXPON=.800000000                                                  AUT11330
      CALL REGRES                                                       AUT11340
      ERSQ8=RSQ                                                         AUT11350
      IF(ERSQ8.LT.ERSQ9) GO TO 510                                      AUT11360
  501 EXPON=1.100000000                                                 AUT11370
      CALL REGRES                                                       AUT11380
      ERSQ11=RSQ                                                        AUT11390
      IF(ERSQ11.LT.LINRSQ) GO TO 888                                    AUT11400
  530 EXPON=1.200000000                                                 AUT11410
      CALL REGRES                                                       AUT11420
      EPSQ12=RSQ                                                        AUT11430
      IF(ERSQ12.LT.ERSQ11) GO TO 888                                    AUT11440
      EXPON=1.300000000                                                 AUT11450
      CALL REGRES                                                       AUT11460
      ERSQ13=RSQ                                                        AUT11470
      IF(ERSQ13.LT.EPSQ12) GO TO 888                                    AUT11480
  531 EXPON=1.400000000                                                 AUT11490
      CALL REGRES                                                       AUT11500
      GO TO 888                                                         AUT11510
  500 EXPON=.700000000                                                  AUT11520
      CALL REGRES                                                       AUT11530
      ERSQ7=RSQ                                                         AUT11540
      IF(ERSQ7.LT.LINRSQ) GO TO 516                                     AUT11550
      EXPON=.600000000                                                  AUT11560
      CALL REGRES                                                       AUT11570
      ERSQ6=RSQ                                                         AUT11580
      IF(ERSQ6.LT.ERSQ7) GO TO 517                                      AUT11590
      EXPON=.500000000                                                  AUT11600
      CALL REGRES                                                       AUT11610
      ERSQ5=RSQ                                                         AUT11620
      IF(ERSQ5.LT.ERSQ6) GO TO 888                                      AUT11630
      GO TO 505                                                         AUT11640
  517 EXPON=.800000000                                                  AUT11650
      CALL REGRES                                                       AUT11660
      ERSQ8=RSQ                                                         AUT11670
      IF(ERSQ8.LT.ERSQ7) GO TO 888                                      AUT11680
      EXPON=.900000000                                                  AUT11690
      CALL REGRES                                                       AUT11700
      GO TO 888                                                         AUT11710
  516 EXPON=.800000000                                                  AUT11720
      CALL REGRES                                                       AUT11730
      ERSQ8=RSQ                                                         AUT11740
      IF(ERSQ8.LT.LINRSQ) GO TO 522                                     AUT11750
      EXPON=.900000000                                                  AUT11760
      CALL REGRES                                                       AUT11770
      GO TO 888                                                         AUT11780
  522 EXPON=1.200000000                                                 AUT11790
      CALL REGRES                                                       AUT11800
      ERSQ12=RSQ                                                        AUT11810
      IF(ERSQ12.LT.LINRSQ) GO TO 888                                    AUT11820
      EXPON=1.300000000                                                 AUT11830
      CALL REGRES                                                       AUT11840
      ERSQ13=RSQ                                                        AUT11850
      IF(ERSQ13.GE.ERSQ12) GO TO 531                                    AUT11860
      EXPON=1.100000000                                                 AUT11870
      CALL REGRES                                                       AUT11880
  888 RETURN                                                            AUT11890
```

```
      END                                                                AUT11900
      SUBROUTINE SUMDIS                                                  AUT11910
      REAL*8 YBASE,TOP,BOT,RSQ,EXPCN                                     AUT11920
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR              AUT11930
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                 AUT11940
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT11950
     *CN,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                 AUT11960
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                 AUT11970
      COMMON/$PROST/IGAP,PROMAM,DORMAT(5,25,3),BESPRO(60,26,3)           AUT11980
C-----------------------------------------------------------------------AUT11990
C   TEST FOR NEGATIVES AND CONVERT TO ZERO AS NEGATIVE PRODUCTION IS    |AUT12000
C   NOT FEASIBLE.                                                       |AUT12010
C-----------------------------------------------------------------------AUT12020
      DO 6 J=1,3                                                         AUT12030
      DO 6 ICOL2=1,L                                                     AUT12040
      I=K                                                                AUT12050
      DO 5 ICK=1,K                                                       AUT12060
C-----------------------------------------------------------------------AUT12070
C   CHANGE THE FOLLOWING STEP IF CYCLICAL REGRESSION IS ADDED, AS THIS  |AUT12080
C   SECTION SEARCHES FROM THE BOTTOM UP FOR A POSITIVE NUMBER AND WHEN  |AUT12090
C   ONE IS FOUND STOPS REPLACING NEGATIVES WITH ZERO.  IN CYCLICAL RE-  |AUT12100
C   GRESSION NEGATIVES COULD BE FOUND THROUGHOUT THE PROJECTIONS.       |AUT12110
C-----------------------------------------------------------------------AUT12120
      IF(BESPRO(I,ICOL2,J).GE.0) GO TO 6                                 AUT12130
      BESPRO(I,ICOL2,J)=0.0                                              AUT12140
      I=I-1                                                              AUT12150
    5 CONTINUE                                                           AUT12160
    6 CONTINUE                                                           AUT12170
C-----------------------------------                                     AUT12180
C   SET DISTRICT COLUMNS TO ZERO. |                                     AUT12190
C-----------------------------------                                     AUT12200
      ICOL2=L+1                                                          AUT12210
      DO 1 J=1,3                                                         AUT12220
      DO 1 I=1,K                                                         AUT12230
    1 BESPRO(I,ICOL2,J)=0.0                                              AUT12240
      DO 3 I=1,N                                                         AUT12250
    3 YBASE(I,ICOL2)=0.0                                                 AUT12260
C-----------------------------------------                              AUT12270
C   SUM ROWS OF BASE AND PROJECTED DATA. |                              AUT12280
C   MATRICES TO GET DISTRICT TOTALS.     |                              AUT12290
C-----------------------------------------                              AUT12300
      DO 2 J=1,3                                                         AUT12310
      DO 2 I=1,K                                                         AUT12320
      DO 2 ICOL=1,L                                                      AUT12330
    2 BESPRO(I,ICOL2,J)=BESPRO(I,ICOL2,J)+BESPRO(I,ICOL,J)               AUT12340
      DO 4 I=1,N                                                         AUT12350
      DO 4 ICOL=1,L                                                      AUT12360
    4 YBASE(I,ICOL2)=YBASE(I,ICOL2)+YBASE(I,ICOL)                        AUT12370
      RETURN                                                             AUT12380
      END                                                                AUT12390
      SUBROUTINE SUMMRY(L)                                               AUT12400
C-----------------------------------------------------------------------AUT12402
C   THIS SUBROUTINE PRINTS SUMMARY STATISTICS FOR EACH DISTRICT.        |AUT12404
C-----------------------------------------------------------------------AUT12406
      COMMON/C/IDISTR,IPRTCT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT12410
     1E5(31),LINE6(31),ICUIT                                             AUT12420
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPROST,ISKIP                    AUT12430
      COMMON/$PAGE/NOPAGE                                                AUT12440
      COMMON/$SLM/ISUMRY                                                 AUT12450
      INTEGER*4 ONE/'1ST'/,TWO/'2ND'/,THREE/'3RD'/                       AUT12460
```

```
      IF(L.LE.13) GO TO 51                                                AUT12470
      IWIDTH=2                                                            AUT12480
      GO TO 32                                                            AUT12490
   51 IWIDTH=1                                                            AUT12500
   32 DO 30 I=1,IWIDTH                                                    AUT12510
      WRITE(6,14) IDISTR                                                  AUT12520
   14 FORMAT('1',48X,'SUMMARY STATISTICS FOR DISTRICT ',A3/'+',48X,35('_AUT12530
     *'))                                                                 AUT12540
      WRITE(6,15)                                                         AUT12550
   15 FORMAT('-')                                                         AUT12560
      NOPAGE=1                                                            AUT12570
      ISUMRY=1                                                            AUT12580
      IPRTDT=1                                                            AUT12590
      IF(IWIDTH.EQ.2) GO TO 10                                            AUT12600
C-----------------------------------------------------------------------AUT12602
C   SET PARAMETERS FOR WRITING 1ST PAGE OF OUTPUT.                     |AUT12604
C-----------------------------------------------------------------------AUT12606
      ISTART=1                                                            AUT12610
      IQUIT=L                                                             AUT12620
      IEND=L+1                                                            AUT12630
      GO TO 11                                                            AUT12640
   10 IF(I.EQ.2) GO TO 12                                                 AUT12650
C-----------------------------------------------------------------------AUT12652
C   SET PARAMETERS FOR WRITING 2ND PAGE OF OUTPUT.                     |AUT12654
C-----------------------------------------------------------------------AUT12656
      ISTART=1                                                            AUT12660
      IEND=13                                                             AUT12670
      IQUIT=13                                                            AUT12680
      GO TO 11                                                            AUT12690
   12 ISTART=14                                                           AUT12700
      IEND=L+1                                                            AUT12710
      IQUIT=L                                                             AUT12720
   11 CALL ITITLE                                                         AUT12730
C-----------------------------------------------------------------------AUT12732
C   WRITE ONE PAGE WIDTH OF EACH PROJECTION SET'S STATISTICS, ALL ON   |AUT12734
C   PAGE.                                                              |AUT12736
C-----------------------------------------------------------------------AUT12738
      DO 30 IPROST=1,3                                                    AUT12740
      IF(IPROST.EQ.1) ISET=ONE                                            AUT12750
      IF(IPROST.EQ.2) ISET=TWO                                            AUT12760
      IF(IPROST.EQ.3) ISET=THREE                                          AUT12770
      ISUMRY=1                                                            AUT12780
      WRITE(6,27) ISET                                                    AUT12790
   27 FORMAT('-',A3,' PROJECTION SET')                                    AUT12800
      CALL ISTAT                                                          AUT12810
   30 CONTINUE                                                            AUT12820
      RETURN                                                              AUT12830
      END                                                                 AUT12840
//LKED.SYSIN DD *                                                         AUT12850
//GO.GETUM DD *                                                           AUT12860
//GO.INCARD DD *                                                          AUT12870
//GO.CARDS DD *                                                           AUT12880
//GO.SYSIN DD *                                                           AUT12890
```

V. AUTOREG MULTI PROGRAMMER'S GUIDE

I. Introduction

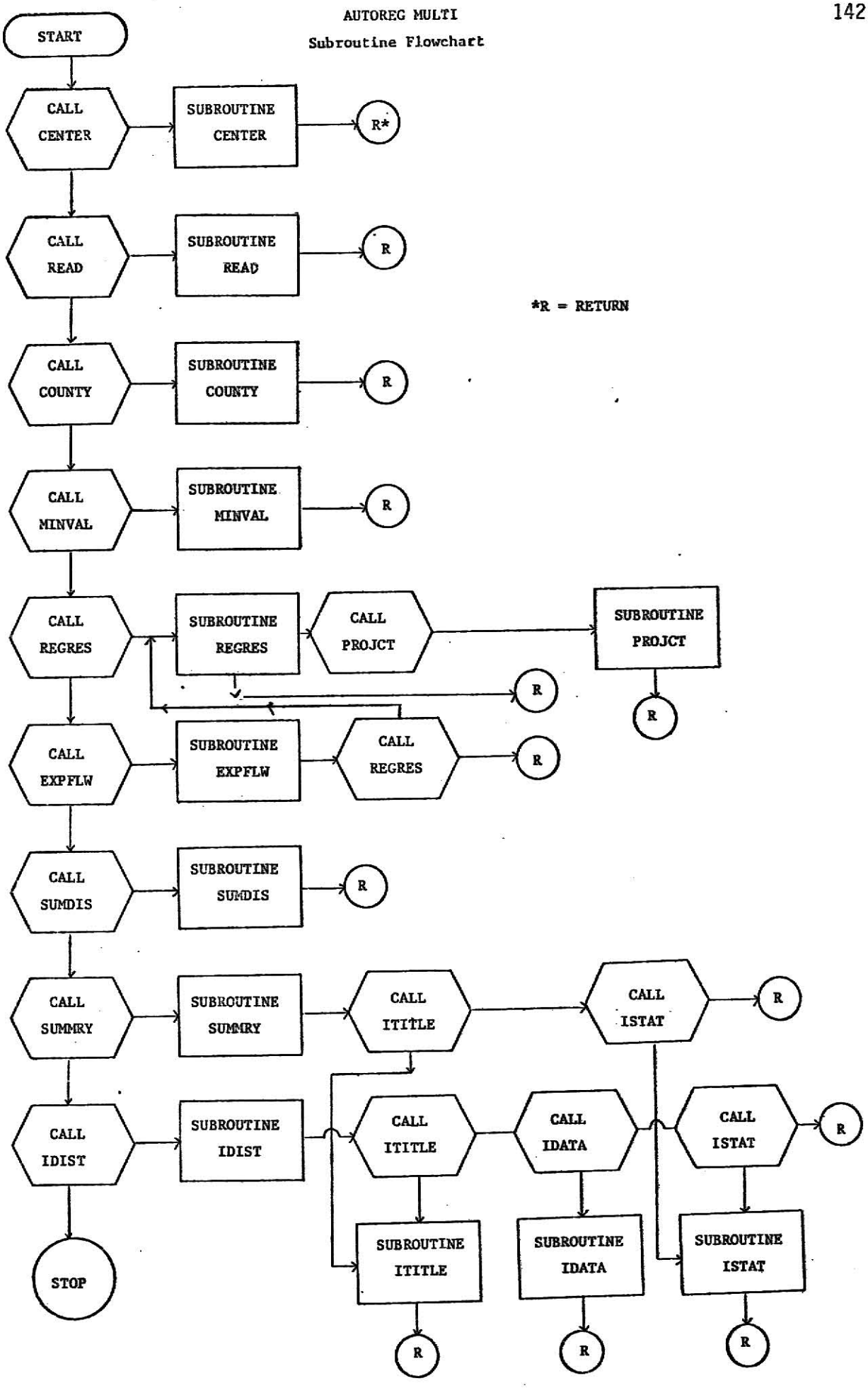
Autoreg Multi is constructed as a series of subroutines for ease in revision. The order of the subroutine calls from the main and from each other is shown on the following page. The use of the three Assembler routines requires that they be compiled as a separate job step and then link-edited with the rest of the subroutines. The JCL and the correct order of steps is shown in the program listing in the user's section.

The program is presently about 1300 cards in length, including comments. Its size is about 80K. The present version now on disk was compiled using the Assembler level G and the Fortran H, with optimization level two, compilers.

The following sections explain the main routine and each subroutine in detail, along with a listing and flowchart of each for ease in making changes.

AUTOREG MULTI

Subroutine Flowchart



II. Main Program

The main program serves to read in all data, do some error checking, and controls the calling of subroutines. (See the subroutine flowchart on the next page which shows the calling order of subroutines.) First, the main program calls two of the assembler programs which read a title and center it or a footnote and then pass them back to the main. Next, the main reads in a control card for a district. (The variables on that card are defined on the main program listing.) The variables on the control card are then tested for being too large for the program size limits. If this occurs an error message is printed telling what the error was and the program terminates.

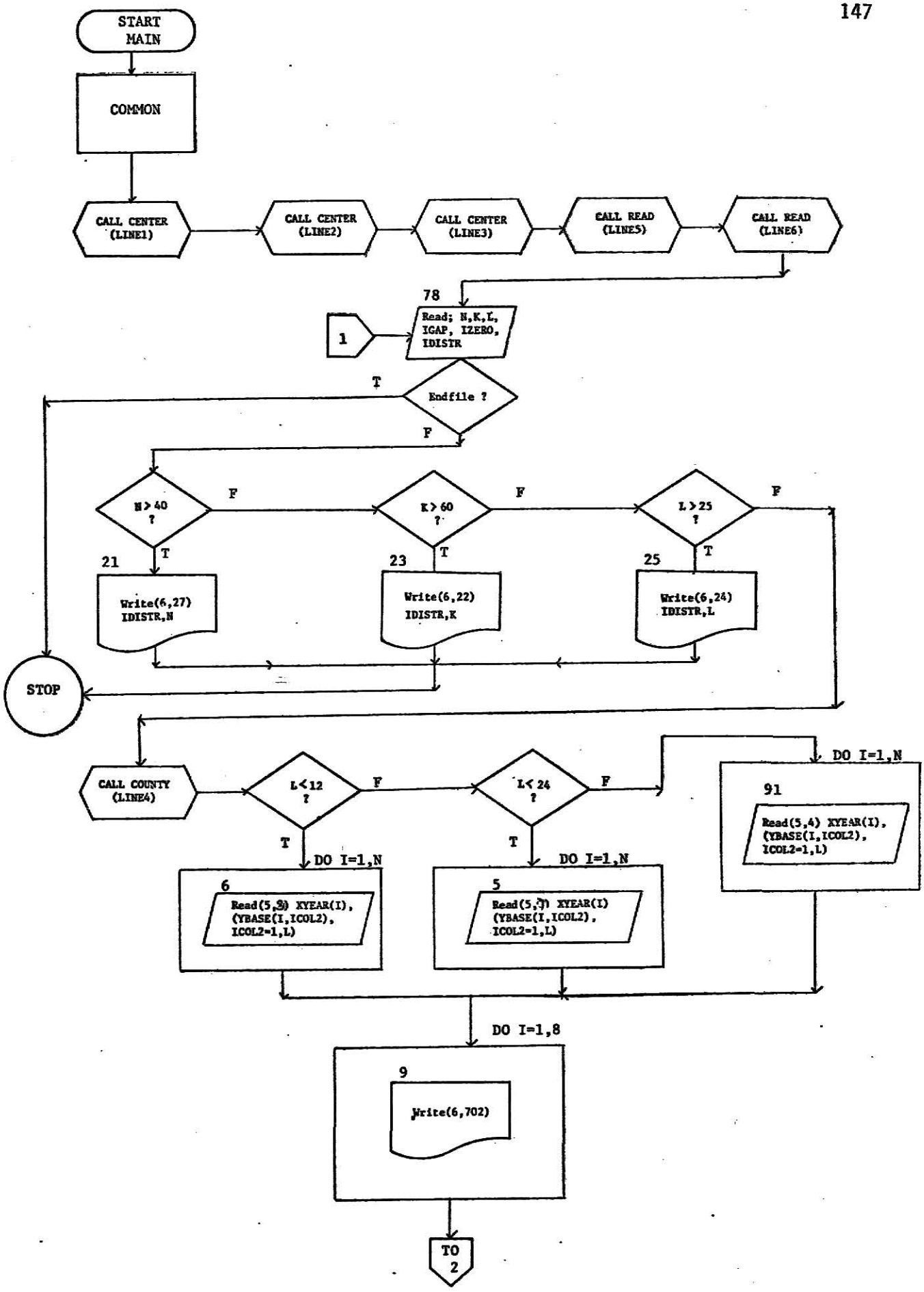
If the program doesn't terminate, the base data is then read in using a suitable format (determined by the value of L on the control card.) The main program then prints a title page for the district. Following this, it initializes the output matrices to zero and then calls all the subroutines necessary to perform the projections and output for a district. The main then returns to the section that reads the district control card. If one is found, all the above steps are performed for a new district. If not, the program terminates. Since the main only returns back to the point reading the control card for all districts after first, it is not necessary to input new titles and footnotes for each district, as the same are used for all.

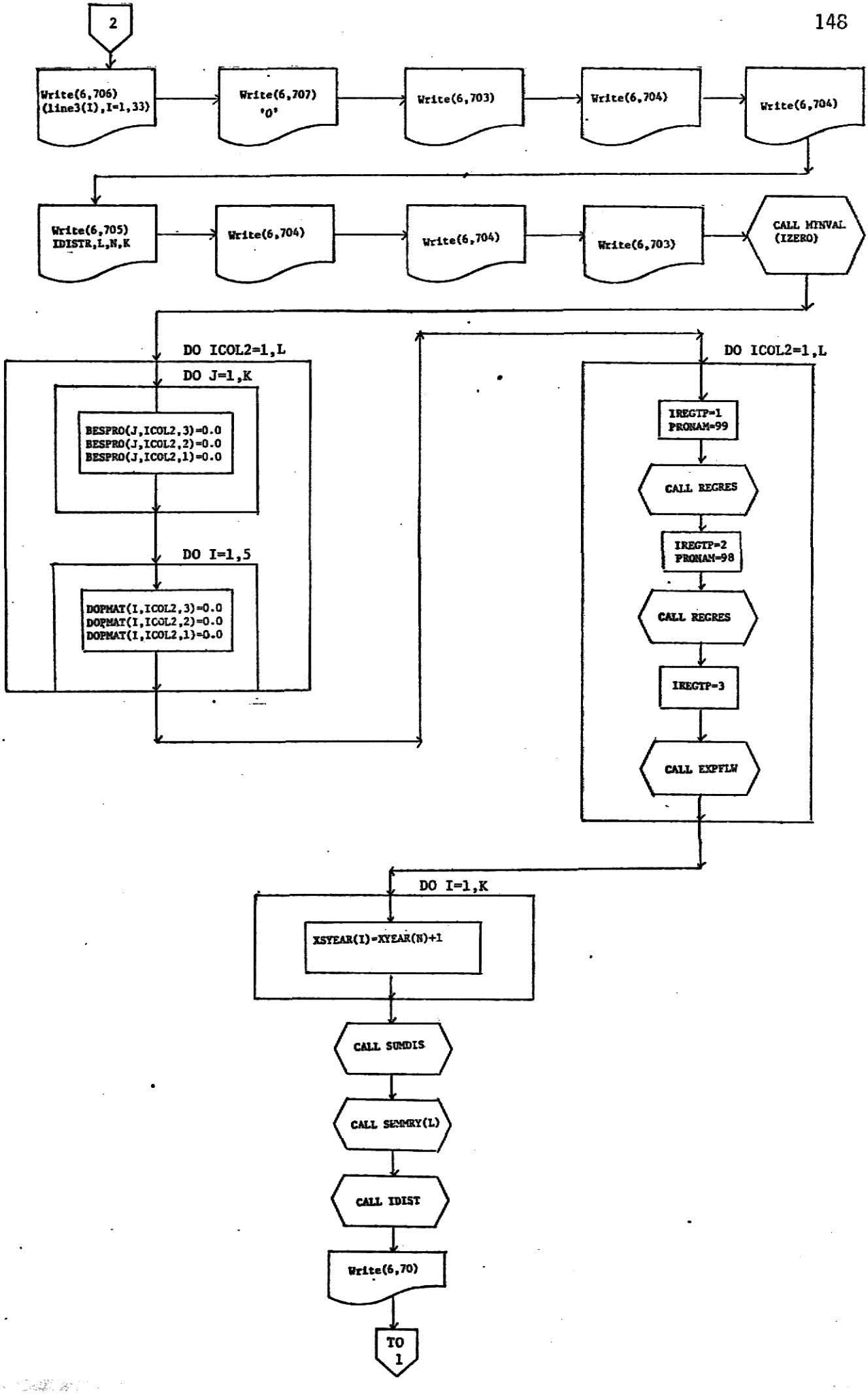
A flowchart of the main program follows a listing of it.

```
      REAL*8 YBASE,TOP,BOT,RSQ,EXPCN                                    AUT02980
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR             AUT02990
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT03000
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT03010
     *ON,RSQ,YSPROJ(60),ICCL2,IREGTP,K,N                                AUT03020
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)          AUT03030
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                AUT03040
      COMMON/C/IDISTR,IPRTDT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT03050
     1E5(31),LINE6(31),ICUIT                                            AUT03060
      INTEGER*4 XYEAR,XSYEAR                                            AUT03070
C-----------------------------------------------------------------------AUT03080
C   THIS SECTION CALLS ASSEMBLER SUBROUTINES WHICH READ AND CENTER THE  |AUT03090
C   3 TITLES FOR THE OUTPUT PAGES AND READ THE 2 LINES OF FOOTNOTES     |AUT03100
C   USED ON EACH OUTPUT PAGE.  (SEE SUBROUTINES READ AND CENTER FOR     |AUT03110
C   FURTHER DETAILS.)  THE INPUT CARDS FOR THIS SECTION ARE INSERTED    |AUT03120
C   ONLY ONCE PER RUN.                                                  |AUT03130
C-----------------------------------------------------------------------AUT03140
   77 CALL CENTER(LINE1)                                                AUT03150
      CALL CENTER(LINE2)                                                AUT03160
      CALL CENTER(LINE3)                                                AUT03170
      CALL READ(LINE5)                                                  AUT03180
      CALL READ(LINE6)                                                  AUT03190
C-----------------------------------------------------------------------AUT03200
C   PUT IN THE NEXT CARD AND ALL THAT FOLLOW ONCE FOR EACH DISTRICT.    |AUT03210
C-----------------------------------------------------------------------AUT03220
C                          ****DEFINITIONS****                          |AUT03230
C                                                                       |AUT03240
C       N = NUMBER OF BASE YEARS OF DATA INPUT (MAX=40)                 |AUT03250
C       K = NUMBER OF YEARS TO MAKE PROJECTIONS FOR (MAX=60)            |AUT03260
C       L = NUMBER OF COUNTIES FOR THIS DISTRICT (MAX=25)               |AUT03270
C    IGAP = THIS CONTROLS THE MINIMUM DIFFERENCE IN RSQ A PROJECTION    |AUT03280
C           MUST HAVE TO REPLACE ANOTHER IN THE SET OF BEST PROJEC-     |AUT03290
C           TIONS.  IF LEFT BLANK, THIS DEFAULTS TO A DIFFERENCE OF     |AUT03300
C           0.0.  IF YOU SET THE VALUE LARGER, IT REDUCES WORK FOR THE  |AUT03310
C           PROGRAM VIA REDUCING THE NUMBER OF PROJECTIONS THAT MUST BE |AUT03320
C           SHUFFLED IN AND OUT OF THE BEST PROJECTIONS MATRIX BY SUB-   |AUT03330
C           ROUTINE PROSTO.  ALSO, THE PROJECTIONS WILL BE SPACED OUT    |AUT03340
C           IN THAT PROJECTIONS WITH AN RSQ LESS THAN IGAP APART WILL    |AUT03350
C           NOT EXIST IN THE BEST PROJECTIONS MATRIX (UNLESS ONE OF THE  |AUT03360
C           FIRST THREE PROJECTIONS RUN AND ITS RSQ EXCEEDS THAT OF ALL  |AUT03370
C           SUBSEQUENT PROJECTIONS.)                                    |AUT03380
C   IZERO = THIS TURNS ON CHECKING FOR ZEROS OR MISSING DATA IN THE     |AUT03390
C           INPUT DATA.  THIS WILL ONLY WORK IF NO 2 CONSEQUTIVE YEARS  |AUT03400
C           ARE MISSING DATA.  (SEE SUBROUTINE MINVAL FOR FURTHER       |AUT03410
C           DETAILS.)  TURN THIS FEATURE ON BY INPUT OF YES IN THE      |AUT03420
C           CORRECT COLUMNS--DEFAULT IS NO CHECKING.                    |AUT03430
C  IDISTR = THE THREE CHARACTERS READ FOR THE VALUE OF IDISTR ARE USED  |AUT03440
C           ON EACH PAGE OF OUTPUT TO IDENTIFY WHICH DISTRICT THE PRO-  |AUT03450
C           JECTIONS ARE FOR.                                           |AUT03460
C-----------------------------------------------------------------------AUT03470
   78 READ(5,90,END=98) N,K,L,IGAP,IZERO,IDISTR                         AUT03480
   90 FORMAT(4(2X,I2),1X,2A3)                                           AUT03490
      IF(N.GT.40) GO TO 21                                              AUT03500
      IF(K.GT.60) GO TO 23                                              AUT03510
      IF(L.GT.25) GO TO 25                                              AUT03520
C-----------------------------------------------------------------------AUT03530
C   THE NAME OF EACH COUNTY IN THE DISTRICT IS READ AT THIS POINT.  SEE |AUT03540
C   SUBROUTINE COUNTY FOR FURTHER DETAILS.) THE CARD(S) WITH THESE      |AUT03550
C   NAMES SHOULD BE INCLUDED FOR EACH DISTRICT.                         |AUT03560
C-----------------------------------------------------------------------AUT03570
```

```
      CALL CCUNTY(LINE4)                                                AUT03580
      IF(L.LE.12) GO TO 6                                               AUT03590
      IF(L.LE.24) GO TO 5                                               AUT03600
C-----------------------------------------------------------------------AUT03610
C   THESE READ FORMATS FCR COUNTY CATA MATCH THOSE OF THE MASTER        |AUT03620
C   PROJECTICN PROGRAM FCR CCNDENSED CATA.                              |AUT03630
C-----------------------------------------------------------------------AUT03640
      DO 91 I=1,N                                                       AUT03650
   91 READ(5,4) XYEAR(I),(YBASE(I,ICOL2),ICOL2=1,L)                     AUT03660
    4 FORMAT(I4,4X,12F6.0/8X,12F6.0/8X,F6.0)                            AUT03670
      GO TC 9                                                           AUT03680
    5 DO 89 I=1,N                                                       AUT03690
   89 READ(5,7) XYEAR(I),(YBASE(I,ICOL2),ICOL2=1,L)                     AUT03700
    7 FORMAT(I4,4X,12F6.0/8X,12F6.0)                                    AUT03710
      GO TO 5                                                           AUT03720
    6 DC 88 I=1,N                                                       AUT03730
   88 READ(5,8) XYEAR(I),(YBASE(I,ICOL2),ICOL2=1,L)                     AUT03740
    8 FORMAT(I4,4X,12F6.0)                                              AUT03750
C--------------------------------------                                 AUT03760
C   WRITE TITLE PAGE FOR EACH DISTRICT. |                               AUT03770
C--------------------------------------                                 AUT03780
    9 DC 710 I=1,8                                                      AUT03790
  710 WRITE(6,702)                                                      AUT03800
  702 FORMAT('-')                                                       AUT03810
      WRITE(6,706) (LINE3(I),I=1,33)                                    AUT03820
  706 FORMAT(1X,33A4)                                                   AUT03830
      WRITE(6,707)                                                      AUT03840
  707 FORMAT('0')                                                       AUT03850
      WRITE(6,703)                                                      AUT03860
  703 FORMAT(33X,69('*'))                                               AUT03870
      WRITE(6,704)                                                      AUT03880
  704 FORMAT(33X,'*',67X,'*')                                           AUT03890
      WRITE(6,704)                                                      AUT03900
      WRITE(6,705) IDISTR,L,N,K                                         AUT03910
  705 FORMAT(33X,'*',2X,'DISTRICT',1X,A3,3X,I2,1X,'COUNTIES',3X,I2,1X,'BAUT03920
     *ASE YEARS',3X,'PRCJECTED',1X,I2,1X,'YEARS',2X,'*')                AUT03930
      WRITE(6,704)                                                      AUT03940
      WRITE(6,704)                                                      AUT03950
      WRITE(6,703)                                                      AUT03960
      CALL MINVAL(IZERO)                                                AUT03970
C-------------------------------------------                            AUT03980
C   INITIALIZE MATRICES DOPMAT AND BESPRC FOR |                         AUT03990
C   FUTURE USE IN SUBRCUTINE PROSTO.          |                         AUT04000
C-------------------------------------------                            AUT04010
      DO 14 ICOL2=1,L                                                   AUT04020
      DO 15 J=1,K                                                       AUT04030
      BESPRC(J,ICOL2,3)=0.0                                             AUT04040
      BESPRC(J,ICOL2,2)=0.0                                             AUT04050
   15 BESPRC(J,ICOL2,1)=0.0                                             AUT04060
      DO 14 I=1,5                                                       AUT04070
      DOPMAT(I,ICOL2,3)=0.0                                             AUT04080
      DOPMAT(I,ICOL2,2)=0.0                                             AUT04090
   14 DOPMAT(I,ICOL2,1)=C.0                                             AUT04100
C-------------------------------------------------------------------    AUT04110
C   START REGRESSION SEQUENCE FOR EACH COUNTY.  (SEE SUBRCUTINES |      AUT04120
C   REGRES,PROJCT,AND EXPFLW FOR FURTHER DETAILS.)               |      AUT04130
C-------------------------------------------------------------------    AUT04140
      DO 93 ICOL2=1,L                                                   AUT04150
      IREGTP=1                                                          AUT04160
      PRONAM=99                                                         AUT04170
```

```
      CALL REGRES                                                       AUT04180
      IREGTP=2                                                          AUT04190
      PRONAM=98                                                         AUT04200
      CALL REGRES                                                       AUT04210
      IREGTP=3                                                          AUT04220
      CALL EXPFLW                                                       AUT04230
   93 CONTINUE                                                          AUT04240
      DO 500 I=1,K                                                      AUT04250
  500 XSYEAR(I)=XYEAR(N)+I                                              AUT04260
      CALL SUMDIS                                                       AUT04270
      CALL SUMMRY(L)                                                    AUT04280
      CALL IDIST                                                        AUT04290
      WRITE(6,70)                                                       AUT04300
   70 FORMAT('1')                                                       AUT04310
      GO TO 78                                                          AUT04320
C-------------------------                                             AUT04330
C   ERROR MESSAGE SECTION. |                                           AUT04340
C-------------------------                                             AUT04350
   21 WRITE(6,27) IDISTR,N                                              AUT04360
   27 FORMAT('0','***ERROR***THE VALUE OF N FOR DISTRICT ',A3,' WAS ',I2AUT04370
     *,' WHICH EXCEEDS'/'THE MAXIMUM FOR THIS PROGRAM OF 40.  TRY AGAIN AUT04380
     *MATE.')                                                           AUT04390
      GO TO 98                                                          AUT04400
   23 WRITE(6,22) IDISTR,K                                              AUT04410
   22 FORMAT('0','***ERROR***THE VALUE OF K FOR DISTRICT ',A3,' WAS ',I2AUT04420
     *,' WHICH EXCEEDS'/'THE MAXIMUM FOR THIS PROGRAM OF 60.  SORRY ABOUAUT04430
     *T THAT.')                                                         AUT04440
      GO TO 98                                                          AUT04450
   25 WRITE(6,24) IDISTR,L                                              AUT04460
   24 FORMAT('0','***ERROR***THE VALUE OF L FOR DISTRICT ',A3,' WAS ',I2AUT04470
     *,' WHICH EXCEEDS'/'THE MAXIMUM FOR THIS PROGRAM OF 25.  TRY ONE MOAUT04480
     *RE TIME.')                                                        AUT04490
   98 STOP                                                              AUT04500
      END                                                               AUT04510
```


START MAIN
COMMON
CALL CENTER (LINE1)
CALL CENTER (LINE2)
CALL CENTER (LINE3)
CALL READ (LINE5)
CALL READ (LINE6)
78
1
Read; N,K,L, IGAP, IZERO, IDISTR
T
Endfile ?
F
N>40 ?
F
K>60 ?
F
L>25 ?
F
T
21
Write(6,27) IDISTR,N
23
Write(6,22) IDISTR,K
25
Write(6,24) IDISTR,L
STOP
CALL COUNTY (LINE4)
L<12 ?
F
L<24 ?
F
T
DO I=1,N
6
Read(5,3) XYEAR(I), (YBASE(I,ICOL2), ICOL2=1,L)
5
Read(5,7) XYEAR(I) (YBASE(I,ICOL2), ICOL2=1,L)
DO I=1,N
91
Read(5,4) XYEAR(I), (YBASE(I,ICOL2), ICOL2=1,L)
DO I=1,8
9
Write(6,702)
TO 2


2
Write(6,706)
(line3(I),I=1,33)
Write(6,707)
'0'
Write(6,703)
Write(6,704)
Write(6,704)
Write(6,705)
IDISTR,L,N,K
Write(6,704)
Write(6,704)
Write(6,703)
CALL MINVAL
(IZERO)
DO ICOL2=1,L
DO J=1,K
BESPRO(J,ICOL2,3)=0.0
BESPRO(J,ICOL2,2)=0.0
BESPRO(J,ICOL2,1)=0.0
DO I=1,5
DOPMAT(I,ICOL2,3)=0.0
DOPMAT(I,ICOL2,2)=0.0
DOPMAT(I,ICOL2,1)=0.0
DO ICOL2=1,L
IREGTP=1
PRONAM=99
CALL REGRES
IREGTP=2
PRONAM=98
CALL REGRES
IREGTP=3
CALL EXPFLW
DO I=1,K
XSYEAR(I)=XYEAR(N)+1
CALL SUMDIS
CALL SUMMRY(L)
CALL IDIST
Write(6,70)
TO
1

III. Subroutine Center

This subroutine reads one or two card images, concatenates them, if two, and centers the result in an output vector that is the length in characters (132) of a page width. If the first column of the first card has a '+' then a second card is read. If not, no attempt is made to read a second card as part of that title. Thus, the user does not have to worry about inserting blank cards in the input deck.

The subroutine works by first blanking out the output vector. Then, the first card is read into a temporary storage position. It is searched for a '+' and, if found, the subroutine reads a second card, putting it in temporary storage at the end of the first card image--thus concatenating the two card images.

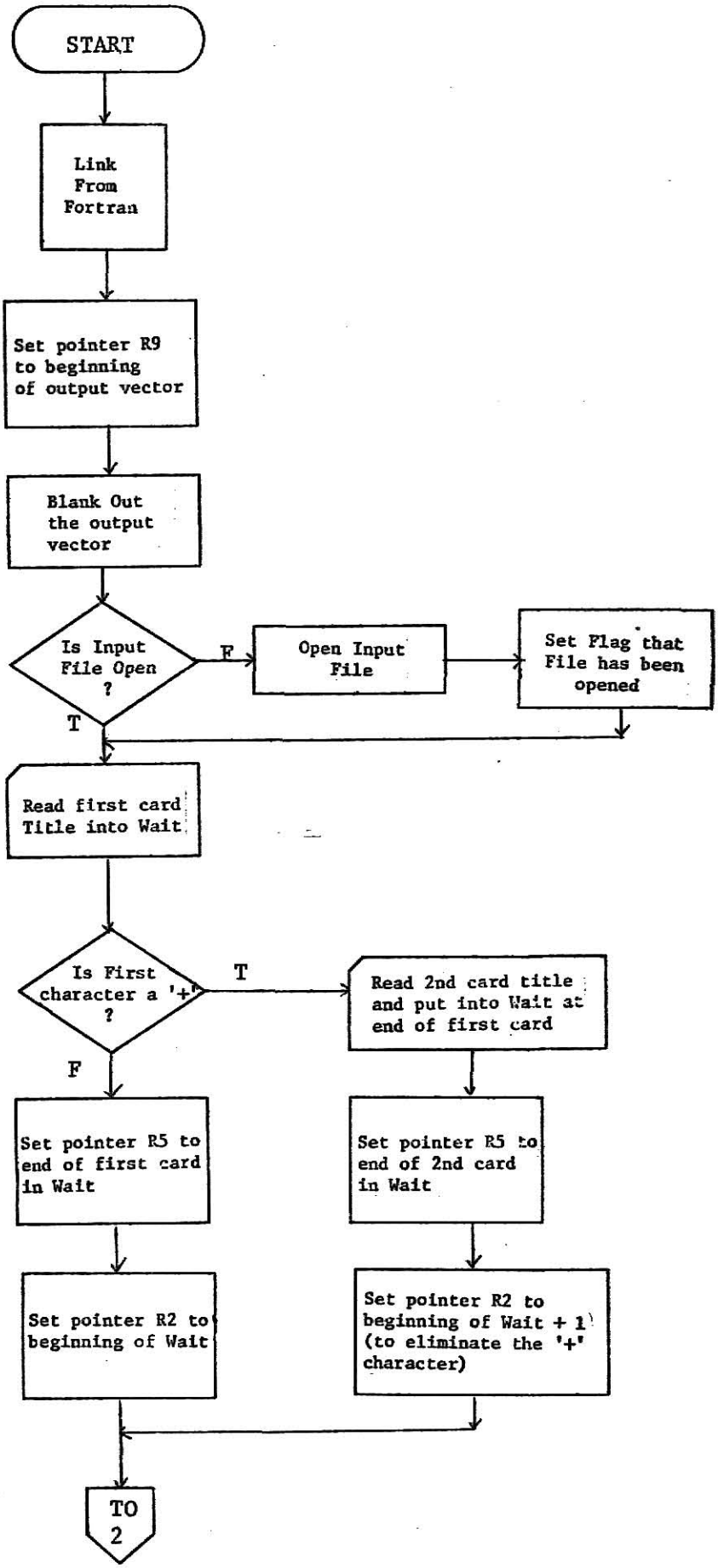
Next, the concatenated card images (or single card image) are searched, starting at the left, for the occurrence of three consecutive blanks that signal the end of the title. Thus, the title may contain up to two consecutive imbedded blanks. As the end of the title is searched, a counter is incremented, so that when the end of the title is found the number of characters in it are known.

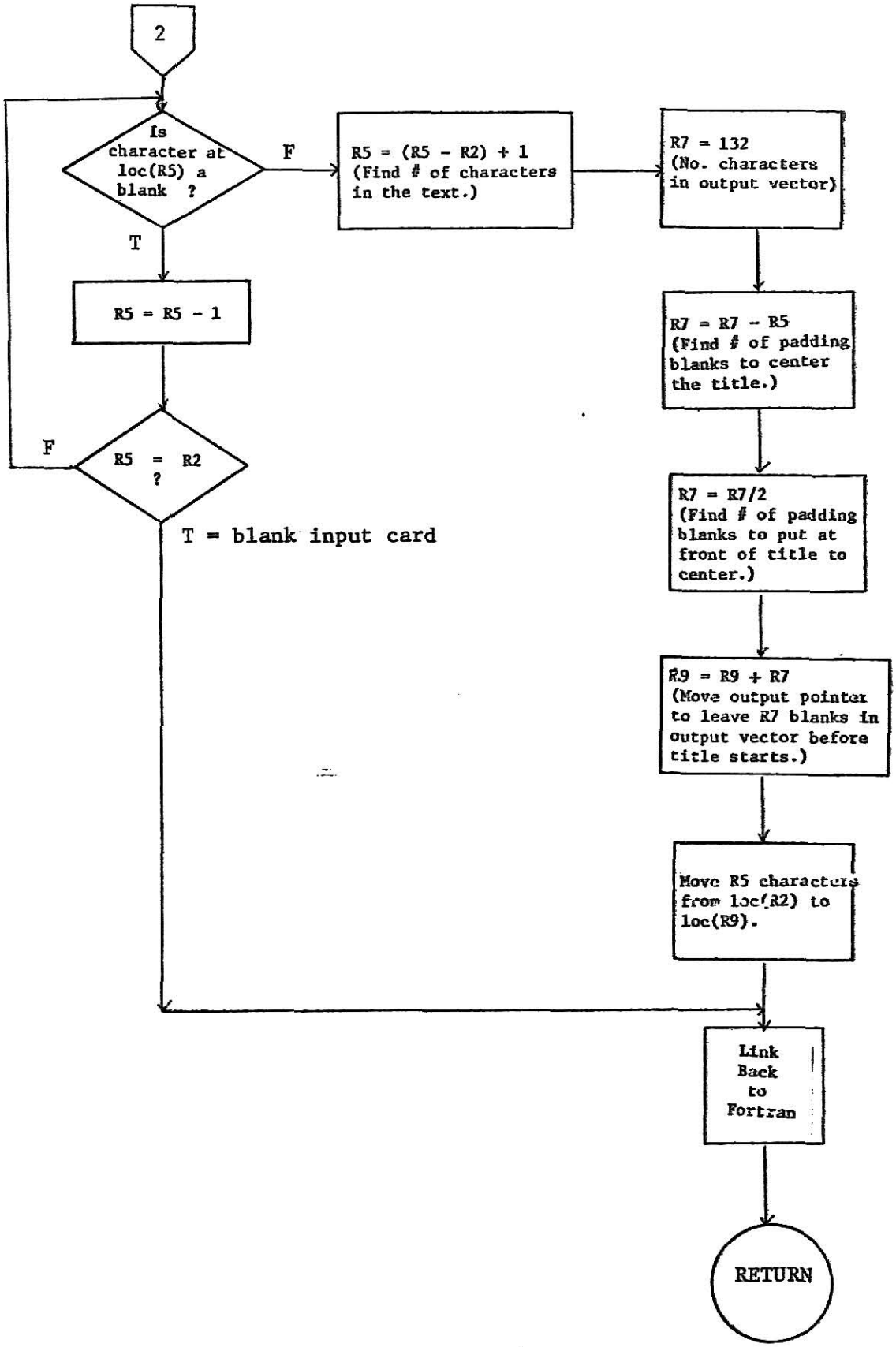
The number of characters are subtracted from 132 (the length of the output vector) to give the number of padding blanks necessary to center the title within the output vector. (As the output vector is the length of one page width--132 characters, when the output vector, containing the title centered within it, is printed on a page of output, the title will appear centered on the output page.) One half the padding blanks are put in front of the title, sans the '+', to center the title in the output vector. This is done by moving the title into the output vector (which is already filled with blanks), starting at a position in the output vector which leaves the correct number of blanks in front of the title to center it. The subroutine then returns to the caller.

A flowchart of subroutine Center follows a listing of it.

```
*********************************************************************** AUT00020
*********************************************************************** AUT00030
*   THIS PROGRAM CENTERS A TITLE PASSED IN WITHIN A 132 SPACE LINE AND  AUT00040
*   PASSES THE RESULTING CENTERED LINE BACK TO THE CALLING PROGRAM.     AUT00050
*   THE TITLE CAN BE ONE OR TWO CARDS IN LENGTH.  IF THE TITLE TO BE    AUT00060
*   PASSED TO THIS ROUTINE IS MORE THAN ONE CARD IN LENGTH, THE FIRST   AUT00070
*   CHARACTER OF THE FIRST CARD SHOULD BE A '+' TO TELL THIS ROUTINE    AUT00080
*   THAT IT WILL BE CENTERING A TWO CARD TITLE.                         AUT00090
*********************************************************************** AUT00100
*********************************************************************** AUT00100
*   R2 = BEGINNING ADDRESS OF TEXT.                                     AUT00110
*   R5 = ENDING ADDRESS OF TEXT.                                        AUT00120
*   R6 = TOP HALF OF R7 FOR DIVISION.                                   AUT00130
*   R7 = # PADDING CHARACTERS FOR FRONT.                                AUT00140
*   R9 = ADDRESS OF FORTRAN'S CHAR. STRING OUTPUT VECTOR.               AUT00150
*********************************************************************** AUT00160
CENTER   CSECT                                                          AUT00170
         PRINT NOGEN                                                    AUT00180
         STM   14,12,12(13)        ST REGS. IN AREA GIVEN BY FORTRAN.   AUT00190
         BALR  12,0                                                     AUT00200
         USING *,12                                                     AUT00210
         LA    14,SAV              LOAD R14 WITH ADDR. OF SUB SAVE AREA AUT00220
         ST    13,4(14)            PUT R13 INTO SUB SAVE AREA.          AUT00230
         ST    14,8(13)            PUT R14 (KEY TO SUB SAVE AREA) INTO  AUT00240
*                                    FORTRAN SAVE AREA.                 AUT00250
         LR    13,14               PUT KEY TO SUB SAVE AREA IN R13.     AUT00260
         L     9,0(1)              LOAD 9 WITH ADDRESS OF LINE IN MAIN. AUT00270
         MVC   0(132,9),=CL132' '  BLANK OUT OUTPUT VECTOR.             AUT00280
         CLI   FLAG,C'/'           TEST FOR FLAG /. IF IT OCCURS THE    AUT00290
         BE    AGIN                  INPUT HAS ALREADY BEEN OPENED.     AUT00300
         OPEN  (CARD,INPUT)        PREPARE FOR INPUT.                   AUT00310
         MVI   FLAG,C'/'           SET FLAG THAT INPUT HAS BEEN OPENED. AUT00320
AGIN     GET   CARD,WAIT           READ A CARD INTO WAIT.               AUT00330
         CLI   WAIT,C'+'           TEST 1ST CHAR OF INPUT FOR A '+'.    AUT00340
         BE    PLUS                IF PLUS IS FOUND, BRANCH TO PLUS.    AUT00350
***** FOR A 1 CARD TITLE.                                               AUT00360
         LA    2,WAIT              LOAD R2 WITH ADDR. OF 1ST CHAR INPUT AUT00370
         LA    5,WAIT+79           LOAD R5 WITH LAST CHAR (INPUT) ADDR. AUT00380
         B     CONTIN              BRANCH TO CONTIN.                    AUT00390
***** FOR A 2 CARD TITLE.                                               AUT00400
PLUS     GET   CARD,WAIT+80        READ CARD INTO WAIT AND PUT IT AT THE AUT00410
*                                    END OF THE FIRST CARD IMAGE .      AUT00420
         LA    5,WAIT+132          LOAD R5 WITH LAST CHAR (INPUT) ADDR. AUT00430
         LA    2,WAIT+1            LOAD R2 WITH ADDR. OF 2ND CHAR IN    AUT00440
*                                    WAIT, SO '+' WILL BE DISCARDED.    AUT00450
***** FIND LAST NON-BLANK CHARACTER.                                     AUT00460
CONTIN   CLI   0(5),C' '           TEST FOR BLANK CHARACTER.            AUT00470
         BNE   FOUND               IF BLANK CHAR NOT FOUND (=LAST NON-  AUT00480
*                                    BLANK CHAR),GO TO FOUND.           AUT00490
         BCTR  5,0                 OTHERWISE,SET PTR.R5 BACK 1,CONTINUE AUT00500
*****IF CARD(S) BLANK, RETURN BLANK.                                     AUT00510
         CR    5,2                 COMPARE PTR.R5 TO PTR.R2.            AUT00520
         BE    RETURN              IF THEY ARE =,INPUT CARD WAS BLANK,  AUT00530
*                                    SO RETURN.                         AUT00540
         B     CONTIN              OTHERWISE, CONTINUE TEST FOR BLANK.  AUT00550
FOUND    SR    5,2                 SUBTRACT PTR.R2 FROM PTR.R5, THE     AUT00560
         LA    5,1(5)                DIFFERENCE+1 = # CHARS IN TEXT.    AUT00570
***** R5 = # CHARACTERS IN THE TEXT.                                    AUT00580
         LA    7,132               LOAD R7 WITH # CHARS IN OUT VECTOR.  AUT00590
         SR    7,5                 CALC. # PADDING BLANKS (R7-R5).      AUT00600
```

```
         SR    6,6                 ZERO R6.                                  AUT00610
         D     6,=F'2'             DIVIDE # BLANKS BY 2 TO FIND # TO         AUT00620
*                                    USE IN CENTERING THE INPUT TITLE.       AUT00630
         AR    9,7                 ADD R7 (# PADDING BLANKS) TO R9.          AUT00640
         LA    6,1                 SUBTRACT 1 FROM # CHARS TO MOVE AS        AUT00650
         SR    5,6                   EX MOVES 1 EXTRA CHAR.                  AUT00660
         EX    5,MOVE              MOVE THE NUMBER OF CHARS SPECIFIED        AUT00670
*                                    BY R5 ACCORDING TO STMT. MOVE.          AUT00680
RETURN   L     13,4(13)            RETR. KEY TO FORTRAN'S REG. STORAGE.      AUT00690
         LM    14,12,12(13)        RESTORE FORTRAN'S REGISTERS.              AUT00700
         MVI   12(13),X'FF'        INDICATE TO FORTRAN WE ARE DONE.          AUT00710
         BR    14                  RETURN.                                   AUT00720
FLAG     DC    CL1' '                                                        AUT00730
SAV      DS    18F                                                           AUT00740
CARD     DCB   MACRF=GM,DDNAME=CARDS,EODAD=RETURN,DSORG=PS                   AUT00750
WAIT     DS    CL160                                                         AUT00760
MOVE     MVC   0(*-*,9),0(2)       MOVE CHARS FROM LOC(R2) TO LOC(R9).       AUT00770
         END                                                                 AUT00780
```


START
Link From Fortran
Set pointer R9 to beginning of output vector
Blank Out the output vector
Is Input File Open ?
F
Open Input File
Set Flag that File has been opened
T
Read first card Title into Wait
Is First character a '+' ?
T
Read 2nd card title and put into Wait at end of first card
F
Set pointer R5 to end of first card in Wait
Set pointer R5 to end of 2nd card in Wait
Set pointer R2 to beginning of Wait
Set pointer R2 to beginning of Wait + 1 (to eliminate the '+' character)
TO 2


2
Is character at loc(R5) a blank ?
F
T
R5 = (R5 - R2) + 1 (Find # of characters in the text.)
R7 = 132 (No. characters in output vector)
R5 = R5 - 1
R7 = R7 - R5 (Find # of padding blanks to center the title.)
F
R5 = R2 ?
T = blank input card
R7 = R7/2 (Find # of padding blanks to put at front of title to center.)
R9 = R9 + R7 (Move output pointer to leave R7 blanks in output vector before title starts.)
Move R5 characters from loc(R2) to loc(R9).
Link Back to Fortran
RETURN

IV. Subroutine Read

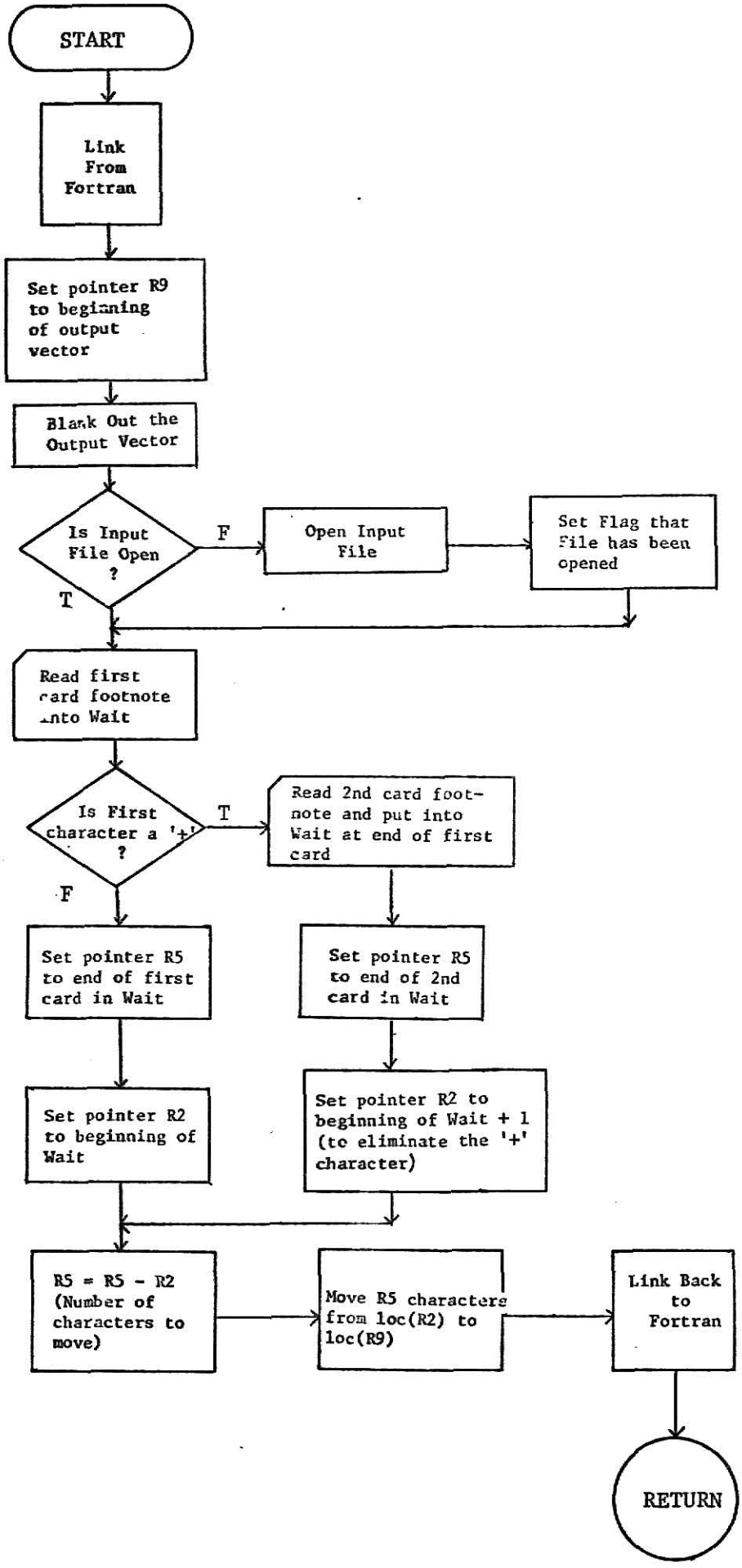
This subroutine reads one or two cards containing footnotes. If two, it concatenates them into one footnote which it passes back in a single vector to the calling program. If the first column of the first card has a '+' then a second card is read. If not, no attempt is made to read a second card. Thus, the user doesn't have to worry about inserting blank cards in the input deck.

The subroutine works by first blanking out the output vector. Then, the first card is read into a temporary storage position. It is searched for a '+' and, if found, the subroutine reads a second card, putting it in temporary storage at the end of the first card image, thus concatenating the two card images. Finally, either the characters from one card, or both if two are read, are transferred to the output vector, eliminating the '+' in the process. The subroutine then returns to the caller.

A flowchart of subroutine Read follows a listing of it.

```
*********************************************************************** AUT00820
*********************************************************************** AUT00820
*   THIS PROGRAM READS A TITLE AND PASSES IT BACK TO THE CALLING        AUT00830
*   ROUTINE.  THE TITLE CAN BE ONE OR TWO CARDS IN LENGTH.  IF THE      AUT00840
*   TITLE TO BE READ IS ONE CARD IN LENGTH, JUST PUT THE CARD IN NOR-   AUT00850
*   MALLY.  IF THE TITLE TO BE READ IN TAKES TWO CARDS, THEN PUT A '+'  AUT00860
*   IN THE FIRST SPOT ON THE FIRST CARD.                                AUT00870
*********************************************************************** AUT00880
*   R2 = BEGINNING ADDRESS OF TEXT.                                     AUT00890
*   R5 = ENDING ADDRESS OF TEXT.                                        AUT00900
*   R9 = ADDRESS OF FORTRAN'S CHAR. STRING OUTPUT VECTOR.               AUT00910
*********************************************************************** AUT00920
READ     CSECT                                                         AUT00930
         PRINT NOGEN                                                   AUT00940
         STM   14,12,12(13)        ST REGS. IN AREA GIVEN BY FORTRAN.  AUT00950
         BALR  12,0                                                    AUT00960
         USING *,12                                                    AUT00970
         LA    14,SAV              LOAD R14 WITH ADDR. OF SUB SAVE AREA AUT00980
         ST    13,4(14)            PUT R13 INTO SUB SAVE AREA.         AUT00990
         ST    14,8(13)            PUT R14 (KEY TO SUB SAVE AREA) INTO AUT01000
*                                    FORTRAN SAVE AREA.                AUT01010
         LR    13,14               PUT KEY TO SUB SAVE AREA IN R13.    AUT01020
         L     9,0(1)              LOAD 9 WITH ADDRESS OF LINE IN MAIN. AUT01030
         MVC   0(124,9),=CL124' '  BLANK OUT OUTPUT VECTOR.            AUT01040
         CLI   FLAG,C'/'           TEST FOR FLAG /. IF IT OCCURS THE   AUT01050
         BE    AGIN                  INPUT HAS ALREADY BEEN OPENED.    AUT01060
         OPEN  (CARD,INPUT)        PREPARE FOR INPUT.                  AUT01070
         MVI   FLAG,C'/'           SET FLAG THAT INPUT HAS BEEN OPENED. AUT01080
AGIN     GET   CARD,WAIT           READ A CARD INTO WAIT.              AUT01090
         CLI   WAIT,C'+'           TEST 1ST CHAR OF INPUT FOR A '+'.   AUT01100
         BE    PLUS                IF PLUS IS FOUND, BRANCH TO PLUS.   AUT01110
***** FOR A 1 CARD TITLE.                                              AUT01120
         LA    2,WAIT              LOAD R2 WITH ADDR. OF 1ST CHAR INPUT AUT01130
         LA    5,WAIT+79           LOAD R5 WITH LAST CHAR (INPUT) ADDR. AUT01140
         B     FOUND               BRANCH TO FOUND.                    AUT01150
***** FOR A 2 CARD TITLE.                                              AUT01160
PLUS     GET   CARD,WAIT+80        READ CARD INTO WAIT AND PUT IT AT THE AUT01170
         LA    5,WAIT+124          LOAD R5 WITH LAST CHAR (INPUT) ADDR. AUT01180
         LA    2,WAIT+1            LOAD R2 WITH ADDR. OF 2ND CHAR IN   AUT01190
*                                    WAIT, SO '+' WILL BE DISCARDED.   AUT01200
FOUND    SR    5,2                 SUBTRACT PTR.R2 FROM PTR.R5, = THE  AUT01210
*                                    ACTUAL NUMBER-1.                  AUT01220
***** R5 = # CHARACTERS IN THE TEXT TO MOVE (=ACTUAL-1).               AUT01230
         EX    5,MOVE              MOVE THE NUMBER OF CHARS SPECIFIED  AUT01240
*                                    BY R5 ACCORDING TO STMT. MOVE.    AUT01250
RETURN   L     13,4(13)            RETR. KEY TO FORTRAN'S REG. STORAGE. AUT01260
         LM    14,12,12(13)        RESTORE FORTRAN'S REGISTERS.        AUT01270
         MVI   12(13),X'FF'        INDICATE TO FORTRAN WE ARE DONE.    AUT01280
         BR    14                  RETURN.                             AUT01290
FLAG     DC    CL1' '                                                  AUT01300
SAV      DS    18F                                                     AUT01310
CARD     DCB   MACRF=GM,DDNAME=INCARD,EODAD=RETURN,DSORG=PS            AUT01320
WAIT     DS    CL160                                                   AUT01330
MOVE     MVC   0(*-*,9),0(2)       MOVE CHARS FROM LOC(R2) TO LOC(R9). AUT01340
         END                                                           AUT01350
```

Flowchart For Read



V. Subroutine County

This subroutine reads up to six input cards containing county names of any length separated by commas, truncates the names to eight characters, centers them over each column of the output pages, and adds the name DISTRICT above the district totals column.

County reads the first card into a temporary storage space. It then checks the first column of the card for a '1', indicating only one card of input. If a '1' is found the reading of cards stops. If a '1' is not found, County reads another card, putting it into the same storage area as the first card, concatenating the images together. The first column of the first card is then checked for a '2', and so on, up to reading in six cards and forming one long character string out of them in the temporary storage area.

The subroutine then takes each set of characters between commas and either truncates them to eight characters long or, if the name between commas is less than 8 characters long, centers it within eight characters. (The county names between commas can contain up to two consecutive imbedded blanks.) Once this is done for a county, the resulting eight character string (or eight characters with centering blanks) is transferred to the output vector such that the eight character names will appear over their respective columns of output later in another subroutine.

The above procedure is repeated for each county until the name for each county has been transferred to the output vector, leaving suitable blanks between county names. Then, the name DISTRICT is added at the end of the output vector so that,when the output vector is later printed by another subroutine, the name DISTRICT will appear over the district totals column. The output vector is then returned to the calling routine.

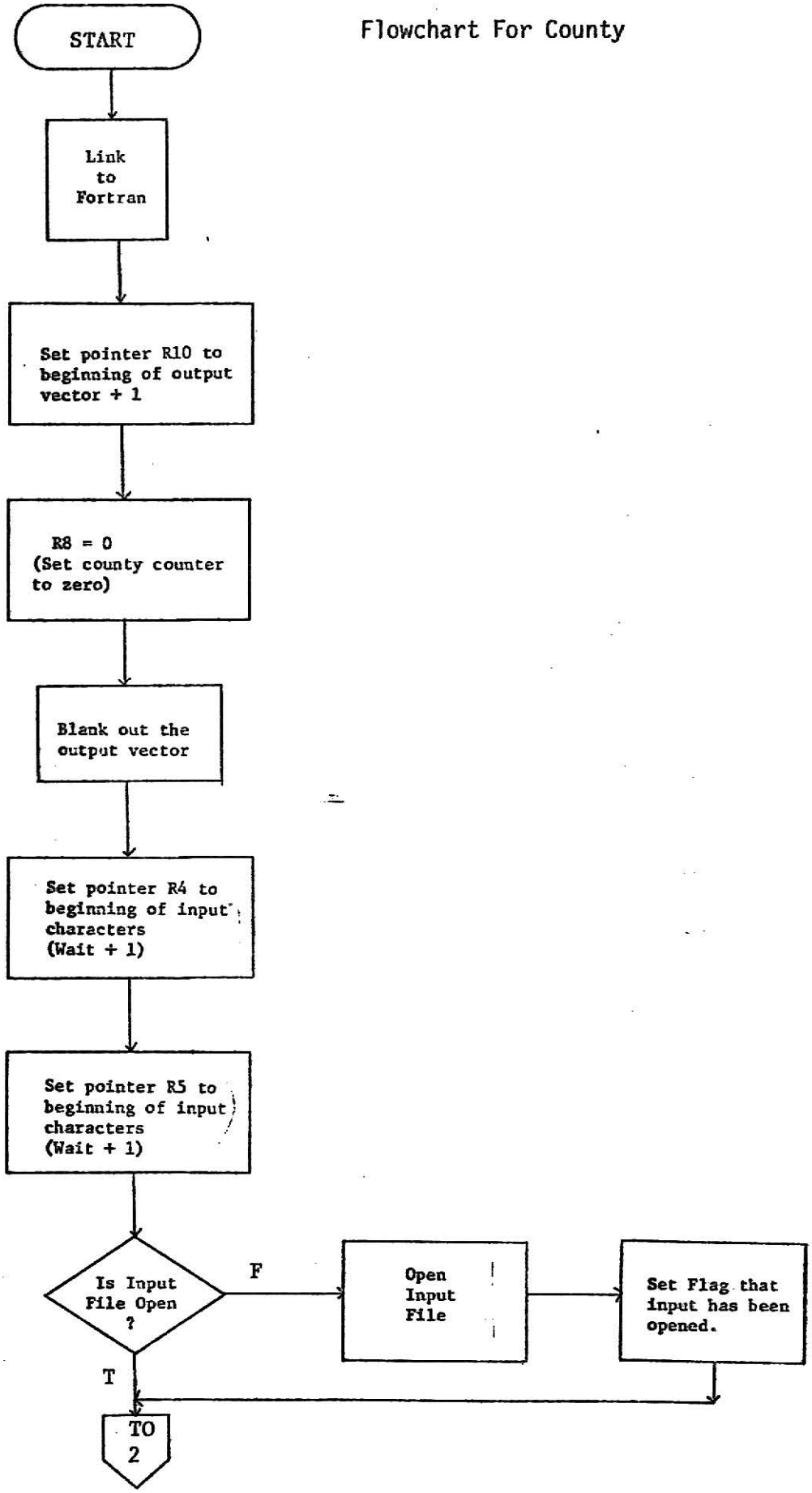
A flowchart of subroutine County follows a listing of it.

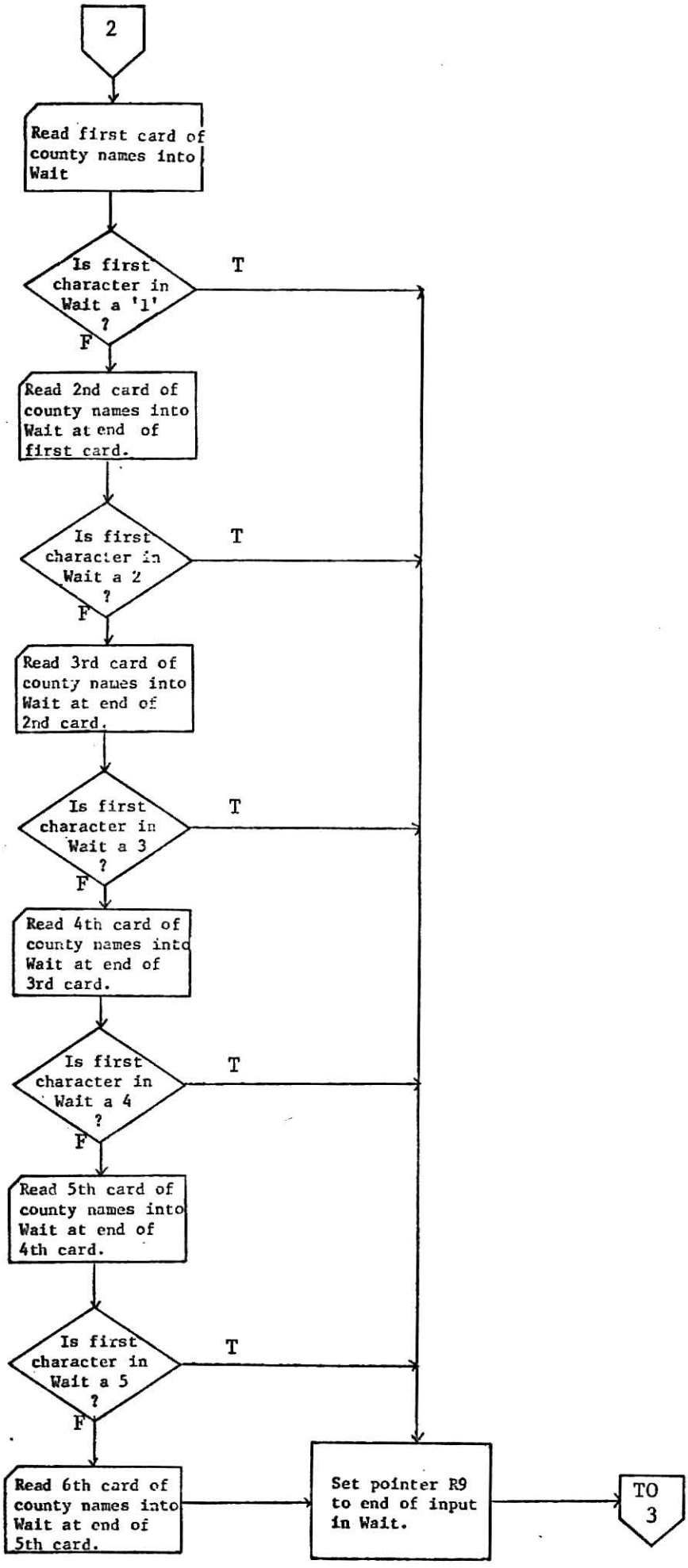
```
COUNTY   CSECT                                                          AUT01390
*********************************************************************** AUT01400
*   THIS SUBROUTINE AUTOMATICALLY CENTERS THE COUNTY NAMES ABOVE THEIR  AUT01410
*   COLUMN IN THE OUTPUT TABLES.  THE INPUT TO THIS PROGRAM CAN BE UP   AUT01420
*   TO 6 CARDS IN LENGTH.  THE NUMBER OF CARDS OF INPUT MUST BE PUT     AUT01430
*   IN THE FIRST COLUMN OF CARD ONE.  THE COUNTY NAMES CAN BE ANY       AUT01440
*   LENGTH, BUT THIS ROUTINE WILL TRUNCATE THEM TO 8 CHARACTERS.  THE   AUT01450
*   COUNTY NAMES MUST BE SEPARATED BY A COMMA AND CONNOT CONTAIN MORE   AUT01460
*   THAN TWO CONSECUTIVE IMBEDDED BLANKS.  THE LAST COUNTY NAME MUST    AUT01470
*   BE FOLLOWED BY A COMMA.  THE TITLE 'DISTRICT' SHOULD NOT BE INPUT,  AUT01480
*   AS THIS ROUTINE SUPPLIES IT.                                        AUT01490
*********************************************************************** AUT01500
*********************************************************************** AUT01500
         PRINT NOGEN                                                    AUT01510
         STM   14,12,12(13)        ST REGS. IN AREA GIVEN BY FORTRAN.   AUT01520
         BALR  12,0                                                     AUT01530
         USING *,12                                                     AUT01540
         LA    14,SAV              LOAD R14 WITH ADDR. OF SUB SAVE AREA AUT01550
         ST    13,4(14)            PUT R13 INTO SUB SAVE AREA.          AUT01560
         ST    14,8(13)            PUT R14 (KEY TO SUB SAVE AREA) INTO  AUT01570
*                                    FORTRAN SAVE AREA.                 AUT01580
         LR    13,14               PUT KEY TO SUB SAVE AREA IN R13.     AUT01590
*********************************************************************** AUT01600
*********************************************************************** AUT01600
*   R2  = COUNTER OF THE NUMBER OF CHARS IN EACH COUNTY NAME.           AUT01610
*   R4  = PTR TO BEGIN OF COUNTY NAME.  USED AS REFERENCE TO MOVE IT.   AUT01620
*   R5  = POINTER TO CURRENT POSITION IN INPUT LINE.                    AUT01630
*   R8  = COUNTER OF THE NUMBER OF COUNTIES IN THE OUTPUT VECTOR.       AUT01640
*   R9  = POINTER TO END OF INPUT LINE.                                 AUT01650
*   R10 = POINTER TO LAST CHAR WRITTEN INTO THE OUTPUT VECTOR.          AUT01660
*********************************************************************** AUT01670
         L     10,0(1)             SET PTR R10 TO BEGIN OF OUTPUT LINE. AUT01680
         LA    8,0                 SET COUNTY COUNTER TO ZERO.          AUT01690
         MVC   0(240,10),=CL240' ' BLANK OUT OUTPUT VECTOR.             AUT01700
         LA    10,1(10)            MOVE PTR 1 TO RIGHT TO LEAVE 1ST     AUT01710
*                                    SPOT IN OUTPUT VECTOR BLANK.       AUT01720
         LA    4,WAIT+1            SET PTR R4 TO BEGIN OF INPUT CHARS.  AUT01730
         LA    5,WAIT+1            SET PTR R5 TO BEGIN OF INPUT CHARS.  AUT01740
         CLI   OPENER,C'/'         TEST FOR FLAG /.  IF IT OCCURS, THE  AUT01750
         BE    AGIN                  INPUT HAS ALREADY BEEN OPENED.     AUT01760
         OPEN  (CARD,INPUT)        PREPARE FOR INPUT.                   AUT01770
         MVI   OPENER,C'/'         SET FLAG THAT INPUT HAS BEEN OPENED. AUT01780
*** FOR 1 CARD OF INPUT.                                                 AUT01790
AGIN     GET   CARD,WAIT           READ 1 CARD INTO WAIT.               AUT01800
         CLI   WAIT,C'1'           TEST FOR NO. CARDS = 1.              AUT01810
         BNE   AGIN2               IF NOT = 1, READ ANOTHER CARD.       AUT01820
         LA    9,WAIT+79           SET PTR R9 TO END OF INPUT LINE.     AUT01830
         B     CONTIN              GO TO THE NEXT SECTION.              AUT01840
*** FOR 2 CARDS OF INPUT.                                               AUT01850
AGIN2    GET   CARD,WAIT+80        PUT 2ND CARD INTO WAIT AT END OF 1ST AUT01860
         CLI   WAIT,C'2'           TEST FOR NO. CARDS = 2.              AUT01870
         BNE   AGIN3               IF NOT = 2,READ ANOTHER CARD.        AUT01880
         LA    9,WAIT+159          SET PTR R9 TO END OF INPUT LINE.     AUT01890
         B     CONTIN              GO TO THE NEXT SECTION.              AUT01900
*** FOR 3 CARDS OF INPUT.                                               AUT01910
AGIN3    GET   CARD,WAIT+160       PUT 3RD CARD INTO WAIT AT END OF 2ND AUT01920
         CLI   WAIT,C'3'           TEST FOR NO. CARDS = 3.              AUT01930
         BNE   AGIN4               IF NOT = 3,READ ANOTHER CARD.        AUT01940
         LA    9,WAIT+239          SET PTR R9 TO END OF INPUT LINE.     AUT01950
         B     CONTIN              GO TO THE NEXT SECTION.              AUT01960
```

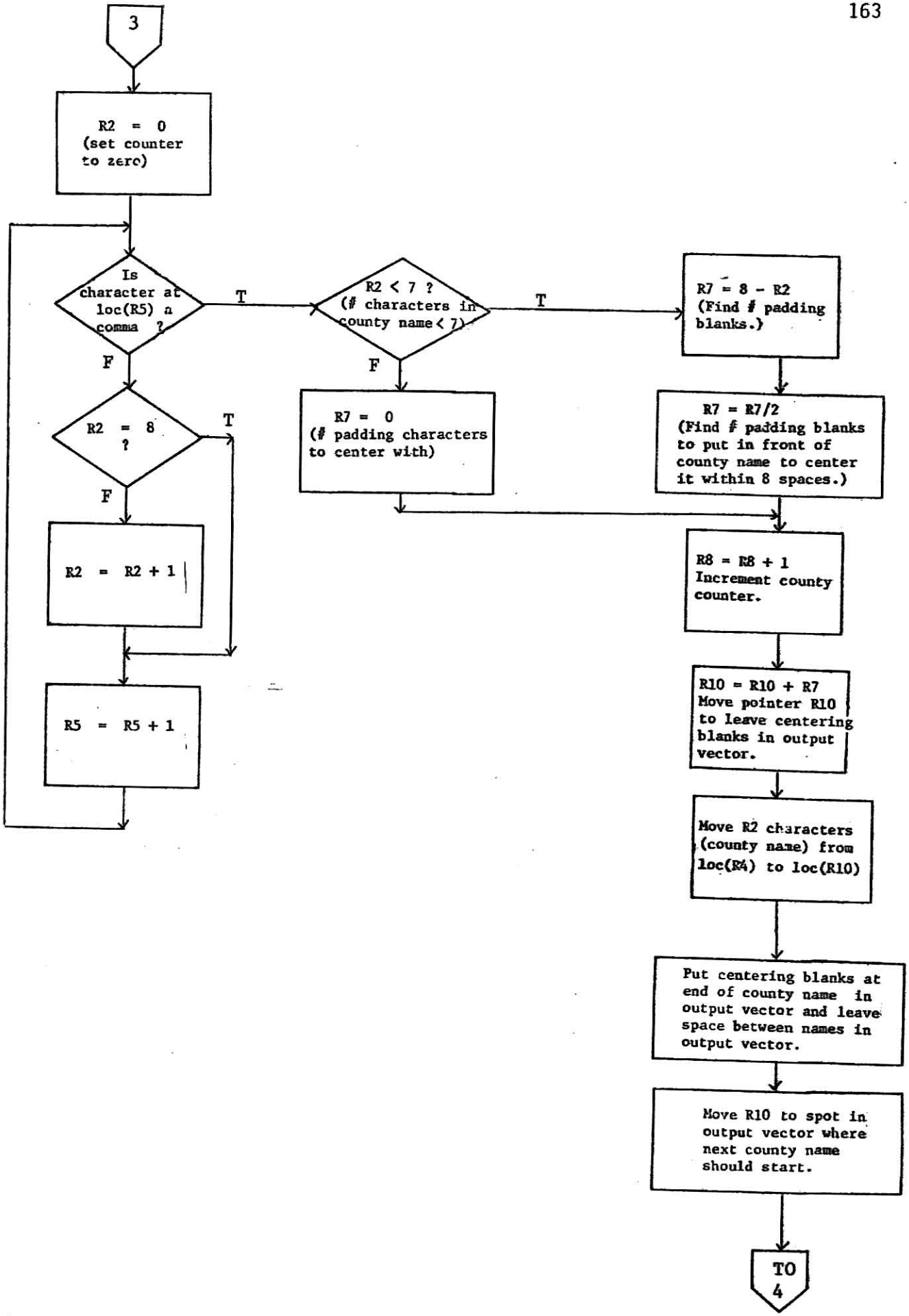
```
*** FOR 4 CARDS OF INPUT.                                              AUT01970
AGIN4    GET   CARD,WAIT+240      PUT 4TH CARD INTO WAIT AT END OF 3RD  AUT01980
         CLI   WAIT,C'4'          TEST FOR NO. CARDS = 4.               AUT01990
         BNE   AGIN5              IF NOT = 4,READ ANOTHER CARD.         AUT02000
         LA    9,WAIT+319         SET PTR R9 TO END OF INPUT LINE.      AUT02010
         B     CONTIN             GO TO THE NEXT SECTION.               AUT02020
*** FOR 5 CARDS OF INPUT.                                               AUT02030
AGIN5    GET   CARD,WAIT+320      PUT 5TH CARD INTO WAIT AT END OF 4TH  AUT02040
         CLI   WAIT,C'5'          TEST FOR NO. CARDS = 5.               AUT02050
         BNE   AGIN6              IF NOT = 5,READ ANOTHER CARD.         AUT02060
         LA    9,WAIT+399         SET PTR R9 TO END OF INPUT LINE.      AUT02070
         B     CONTIN             GO TO THE NEXT SECTION.               AUT02080
*** FOR 6 CARDS OF INPUT.                                               AUT02090
AGIN6    GET   CARD,WAIT+400      PUT 6TH CARD INTO WAIT AT END OF 5TH  AUT02100
         LA    9,WAIT+479         SET PTR R9 TO END OF INPUT LINE.      AUT02110
*********************************************************************** AUT02120
CONTIN   LA    2,0                SET COUNTER TO 0.                     AUT02130
         LA    11,1               LOAD R11 WITH 1.                      AUT02140
         LA    3,8                SET LIMIT TO 8.                       AUT02150
NEXT     CLI   0(5),C','          SEARCH FOR A COMMA.                   AUT02160
         BE    COMMA              IF A COMMA IS FOUND, GO TO COMMA.     AUT02170
         CR    3,2                TEST FOR # CHARS = 8.                 AUT02180
         BE    STILL              IF # CHARS = 8 DON'T INCREMENT CTR.    AUT02190
         LA    2,1(2)             INCREMENT COUNTER.                    AUT02200
STILL    LA    5,1(5)             INCREMENT PTR R5.                     AUT02210
         B     NEXT               CONTINUE LOOP NEXT.                   AUT02220
COMMA    LA    5,1(5)             INCREMENT PTR R5 SO IT WILL POINT     AUT02230
*                                   TO BEGIN OF NEXT COUNTY NAME.       AUT02240
         LA    8,1(8)             INCREMENT COUNTER.                    AUT02250
         SR    3,11               SET LIMIT TO 7.                       AUT02260
         CR    3,2                TEST # CHARS IN COUNTY NAME AGAINST   AUT02270
*                                   LIMIT (R3).                         AUT02280
         BH    CENTER             IF R2 < 7 GO TO CENTER.               AUT02290
***COUNTY NAME TO PUT IN OUTPUT IS 7 OR 8 CHARS.                        AUT02300
         SR    7,7                ZERO R7 (# BLANKS TO PAD FRONT WITH)  AUT02310
         B     MOVEIT             GO TO MOVEIT.                         AUT02320
***COUNTY NAME IS LESS THAN 7 CHARS LONG, MUST BE CENTERED.            AUT02330
CENTER   LA    7,8                LOAD R7 WITH 8.                       AUT02340
         SR    7,2                FIND NUMBER OF PADDING BLANKS.        AUT02350
         SR    6,6                ZERO R6 FOR DIVISION.                 AUT02360
         D     6,=F'2'            FIND # OF BLANKS TO PAD FRONT WITH.   AUT02370
*********************************************************** AUT02380
*** R6  = # PADDING BLANKS IN FRONT OF COUNTY NAME. ***     AUT02390
*** R4  = START OF COUNTY NAME.                     ***     AUT02400
*** R2  = LENGTH OF COUNTY NAME.                    ***     AUT02410
*** R10 = DESTINATION OF COUNTY NAME.               ***     AUT02420
*********************************************************** AUT02430
MOVEIT   AR    10,7               MOVE PTR TO OUTPUT VECTOR FORWARD.    AUT02440
*                                   TO SKIP PADDING BLANKS.             AUT02450
         SR    2,11               SUBTRACT 1 FROM # CHARS TO MOVE AS    AUT02460
*                                   EXEC MOVES 1 TOO MANY CHARS.        AUT02470
         EX    2,MOVE             MOVE # CHARS SPECIFIED BY R2 ACCORD-  AUT02480
*                                   ING TO STMT. MOVE.                  AUT02490
         LA    2,8                LOAD R2 WITH 8.                       AUT02500
         SR    2,7                FIND DIFFERENCE OF R2-NUMBER OF       AUT02510
*                                   SPACES PTR R10 HAS ALREADY BEEN     AUT02520
*                                   MOVED TO ALLOW FOR PADDING BLANKS.  AUT02530
         AR    10,2               ADD THIS NUMBER TO PTR 10 SO IT       AUT02540
*                                   NOW POINTS TO SPOT IN OUTPUT        AUT02550
*                                   VECTOR WHERE NEXT NAME GOES.        AUT02560
```

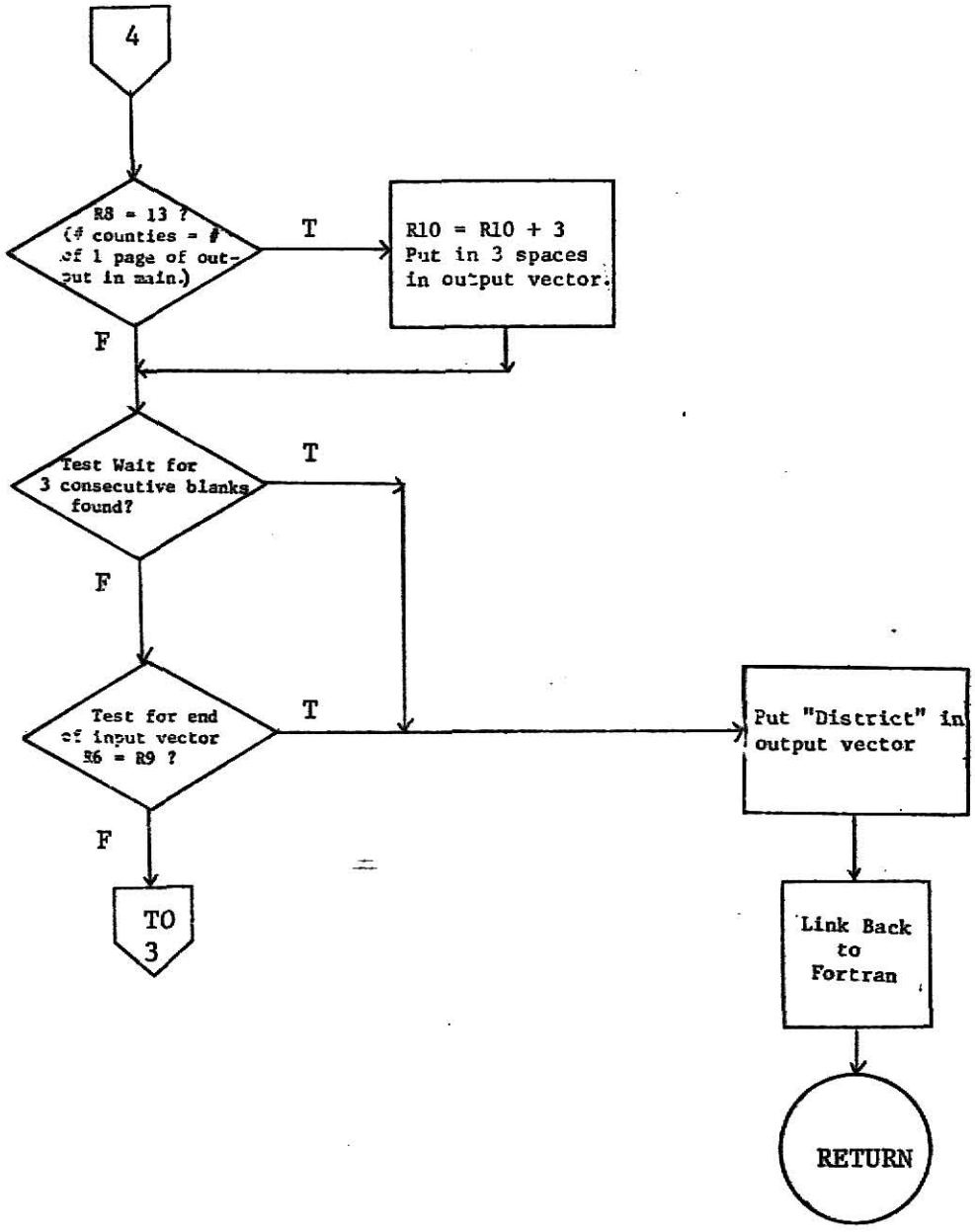
```
         LA    10,1(10)            MOVE OUTPUT PTR TO LEAVE SPACE       AUT02570
*                                    BETWEEN COUNTY NAMES IN OUTPUT.     AUT02580
***TEST FOR SPOT IN OUTPUT VECTOR THAT = THE END OF TITLE FOR ONE         AUT02590
***  PAGE HEADING IN MAIN.  IF FOUND PUT IN TWO SPACES.                  AUT02600
         C     8,=F'13'            TEST FOR 13 COUNTIES,TIME TO SPACE.  AUT02610
         BNE   NOSPACE             IF NOT = 118 DON'T SPACE.            AUT02620
         LA    10,3(10)            PUT IN THREE BLANK SPACES.           AUT02630
***SEARCH FOR 3 CONSECUTIVE BLANKS IN INPUT OR INPUT END.                AUT02640
NOSPACE  LR    6,5                 LOAD R6 WITH ADDRESS OF INPUT PTR.   AUT02650
         LA    3,3                 LOAD R3 WITH 3.                      AUT02660
         SR    7,7                 ZERO R7 (=COUNTER).                  AUT02670
LOOP     CLI   0(6),C' '           SEARCH FOR A BLANK CHAR.             AUT02680
         BNE   GOMORE              IF BLANK NOT FOUND, GO TO GOMORE.    AUT02690
         LA    7,1(7)              INCREMENT COUNTER R7 BY 1.           AUT02700
         CR    7,3                 TEST FOR COUNTER=3 (3 BLANKS).       AUT02710
         BE    RETURN              IF 3 BLANKS, GO TO RETURN.           AUT02720
         LA    6,1(6)              INCREMENT POINTER BY 1.              AUT02730
         B     LOOP                GO TO LOOP.                          AUT02740
GOMORE   CR    6,9                 TEST FOR END OF INPUT VECTOR, USING  AUT02750
*                                  R6 IN CASE THE LAST OF THE INPUT     AUT02760
*                                    CONSISTS OF LESS THAN 3 BLANKS.    AUT02770
         BE    RETURN              GO TO RETURN IF END OF INPUT.        AUT02780
         BH    RETURN                                                   AUT02790
         LR    4,5                 SET PTR R4 TO BEGINNING OF NEXT      AUT02800
*                                    COUNTY NAME.                       AUT02810
         B     CONTIN              GO TO CONTIN.                        AUT02820
RETURN   MVC   0(8,10),DIST        PUT 'DISTRICT' AT END OF OUTPUT.     AUT02830
         L     13,4(13)            RETR. KEY TO FORTRAN'S REG. STORAGE. AUT02840
         LM    14,12,12(13)        RESTORE FORTRAN'S REGISTERS.         AUT02850
         MVI   12(13),X'FF'        INDICATE TO FORTRAN WE ARE DONE.     AUT02860
         BR    14                  RETURN.                              AUT02870
SAV      DS    18F                                                      AUT02880
CARD     DCB   MACRF=GM,DDNAME=GETUM,EODAD=RETURN,DSORG=PS              AUT02890
WAIT     DS    CL480               RESERVE STORAGE FOR 6 CARDS OF INPUT AUT02900
OPENER   DC    CL1' '                                                   AUT02910
MOVE     MVC   0(*-*,10),0(4)      MOVE CHARS FROM LOC(R4) TO LOC(R10). AUT02920
DIST     DC    CL8'DISTRICT'                                            AUT02930
         END                                                            AUT02940
```

Flowchart For County




2
Read first card of county names into Wait
Is first character in Wait a '1' ?
T
F
Read 2nd card of county names into Wait at end of first card.
Is first character in Wait a 2 ?
T
F
Read 3rd card of county names into Wait at end of 2nd card.
Is first character in Wait a 3 ?
T
F
Read 4th card of county names into Wait at end of 3rd card.
Is first character in Wait a 4 ?
T
F
Read 5th card of county names into Wait at end of 4th card.
Is first character in Wait a 5 ?
T
F
Read 6th card of county names into Wait at end of 5th card.
Set pointer R9 to end of input in Wait.
TO 3


3
R2 = 0 (set counter to zero)
Is character at loc(R5) a comma ?
T
F
R2 < 7 ? (# characters in county name < 7)
T
F
R7 = 8 - R2 (Find # padding blanks.)
R7 = 0 (# padding characters to center with)
R7 = R7/2 (Find # padding blanks to put in front of county name to center it within 8 spaces.)
R2 = 8 ?
T
F
R2 = R2 + 1
R5 = R5 + 1
R8 = R8 + 1 Increment county counter.
R10 = R10 + R7 Move pointer R10 to leave centering blanks in output vector.
Move R2 characters (county name) from loc(R4) to loc(R10)
Put centering blanks at end of county name in output vector and leave space between names in output vector.
Move R10 to spot in output vector where next county name should start.
TO 4


4
R8 = 13 ?
(# counties = # of 1 page of output in main.)
T
R10 = R10 + 3
Put in 3 spaces in output vector.
F
Test Wait for 3 consecutive blanks found?
T
F
Test for end of input vector R6 = R9 ?
T
F
TO 3
Put "District" in output vector
Link Back to Fortran
RETURN

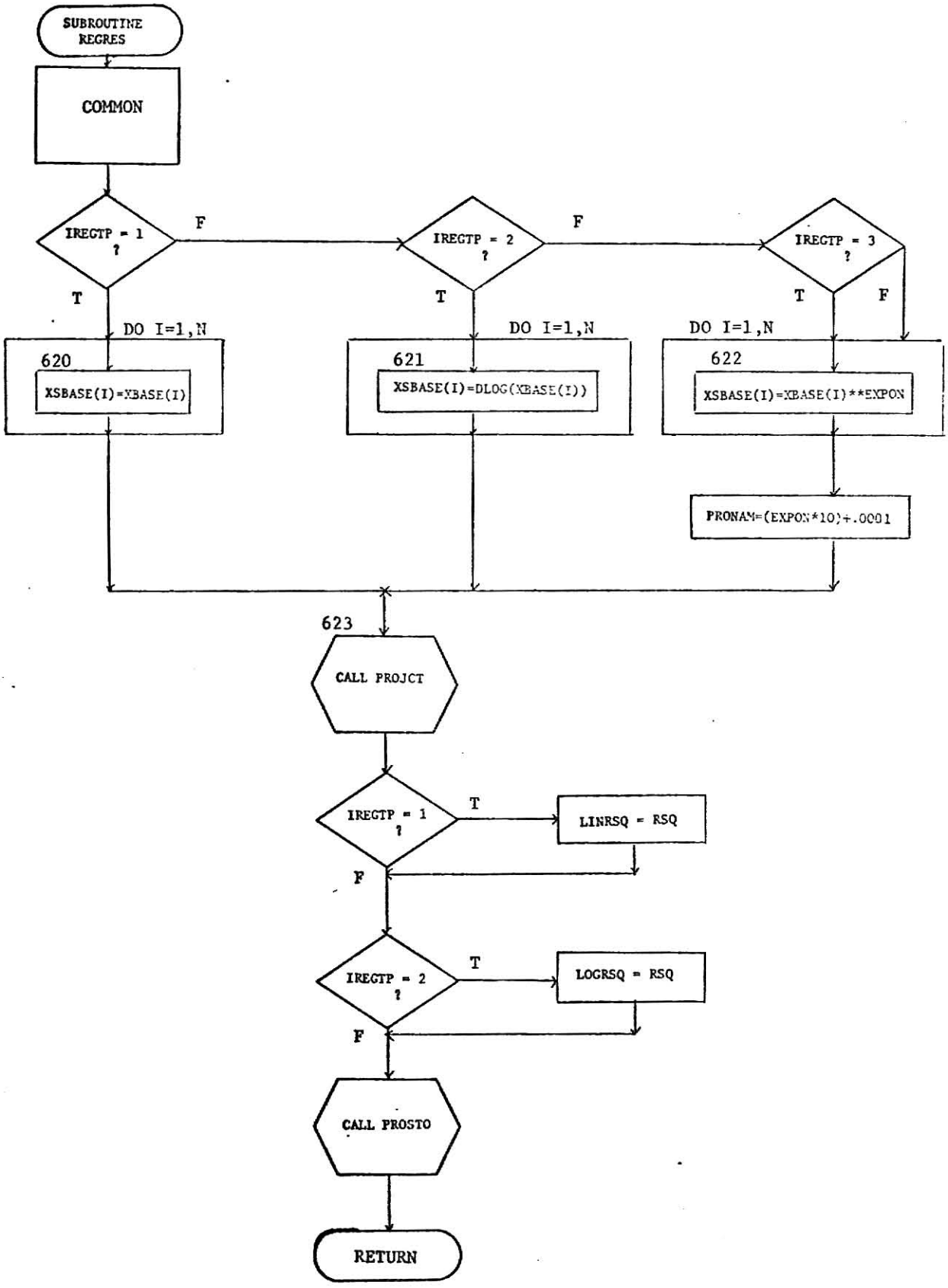
VI. Subroutine Regres

This subroutine transforms the X values in the input data (years), calls Projct to project the data, and then calls Prosto to store the projection. Regres is a separate subroutine from Projct so that Projct can be replaced at a future date by an Assembler language version of it.

The input data is transformed according to the input values of EXPON and IREGTP (set when Expflw or Main calls Regres). If IREGTP equals 1 the X's aren't transformed and the resulting linear projections equation takes the form Y = a + bX. When IREGTP equals 2 the X's are transformed by taking their logarithms with the resulting projection equation being of the form Y = a + b(LogX). When IREGTP is equal to 3 the X's are taken to the EXPON power before calculating the trend line equation. This equation takes the form $Y = a + b(X^{EXPON})$ where EXPON is passed to REGRES by Expflw. When EXPON is passed in from Expflw it always lies between 0.4 and 1.4 inclusive.

A flowchart of REGRES follows a listing of it.

```
      SUBROUTINE REGRES                                                    AUT04520
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                       AUT04530
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR                AUT04540
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                   AUT04550
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT04560
     *CN,RSQ,YSPROJ(60),ICCL2,IREGTP,K,N                                   AUT04570
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                   AUT04580
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)             AUT04590
      COMMON/$EXP/LINRSQ                                                   AUT04600
      REAL*4 LINRSQ                                                        AUT04610
      INTEGER*4 XYEAR,XSYEAR                                               AUT04620
C-----------------------------------------------------------------------AUT04630
C   THIS SECTION TRANSFORMS XBASE ACCORDING TO THE PROJECTION TYPE AS  |AUT04640
C   SPECIFIED BY IREGTP AND STORES THE RESULTS IN XSBASE.  (SEE SUB-   |AUT04650
C   ROUTINE PROJCT FOR VARIABLE DEFINITIONS.)                          |AUT04660
C-----------------------------------------------------------------------AUT04670
      GO TO (620,621,622),IREGTP                                           AUT04680
  622 DO 708 I=1,N                                                         AUT04690
  708 XSBASE(I)=XBASE(I)**EXPON                                            AUT04700
      PRONAM=(EXPON*10)+.0001                                              AUT04710
      GO TO 623                                                            AUT04720
  621 DO 700 I=1,N                                                         AUT04730
  700 XSBASE(I)=DLOG(XBASE(I))                                             AUT04740
      GO TO 623                                                            AUT04750
  620 DO 701 I=1,N                                                         AUT04760
  701 XSBASE(I)=XBASE(I)                                                   AUT04770
C-------------------------------------------                             AUT04780
C   SUBROUTINE PROJCT CALLED HERE PERFORMS THE |                         AUT04790
C   ACTUAL REGRESSIONS AND MAKES PROJECTIONS.  |                         AUT04800
C-------------------------------------------                             AUT04810
  623 CALL PROJCT                                                          AUT04820
  996 IF(IREGTP.EQ.1) LINRSQ=RSQ                                           AUT04830
C     IF(IREGTP.EQ.2) LOGRSQ=RSQ    (STATEMENT FOR FUTURE EXPANSION.)    AUT04840
C-------------------------------------                                   AUT04850
C   SUBROUTINE PROSTC CALLED HERE     |                                  AUT04860
C   STORES THE COMPLETED PROJECTIONS. |                                  AUT04870
C-------------------------------------                                   AUT04880
      CALL PROSTO                                                          AUT04890
      RETURN                                                               AUT04900
      END                                                                  AUT04910
```


SUBROUTINE REGRES
COMMON
IREGTP = 1 ?
F
T
IREGTP = 2 ?
F
T
IREGTP = 3 ?
T
F
DO I=1,N
620
XSBASE(I)=XBASE(I)
DO I=1,N
621
XSBASE(I)=DLOG(XBASE(I))
DO I=1,N
622
XSBASE(I)=XBASE(I)**EXPON
PRONAM=(EXPON*10)+.0001
623
CALL PROJCT
IREGTP = 1 ?
T
LINRSQ = RSQ
F
IREGTP = 2 ?
T
LOGRSQ = RSQ
F
CALL PROSTO
RETURN

VII. Subroutine Projct

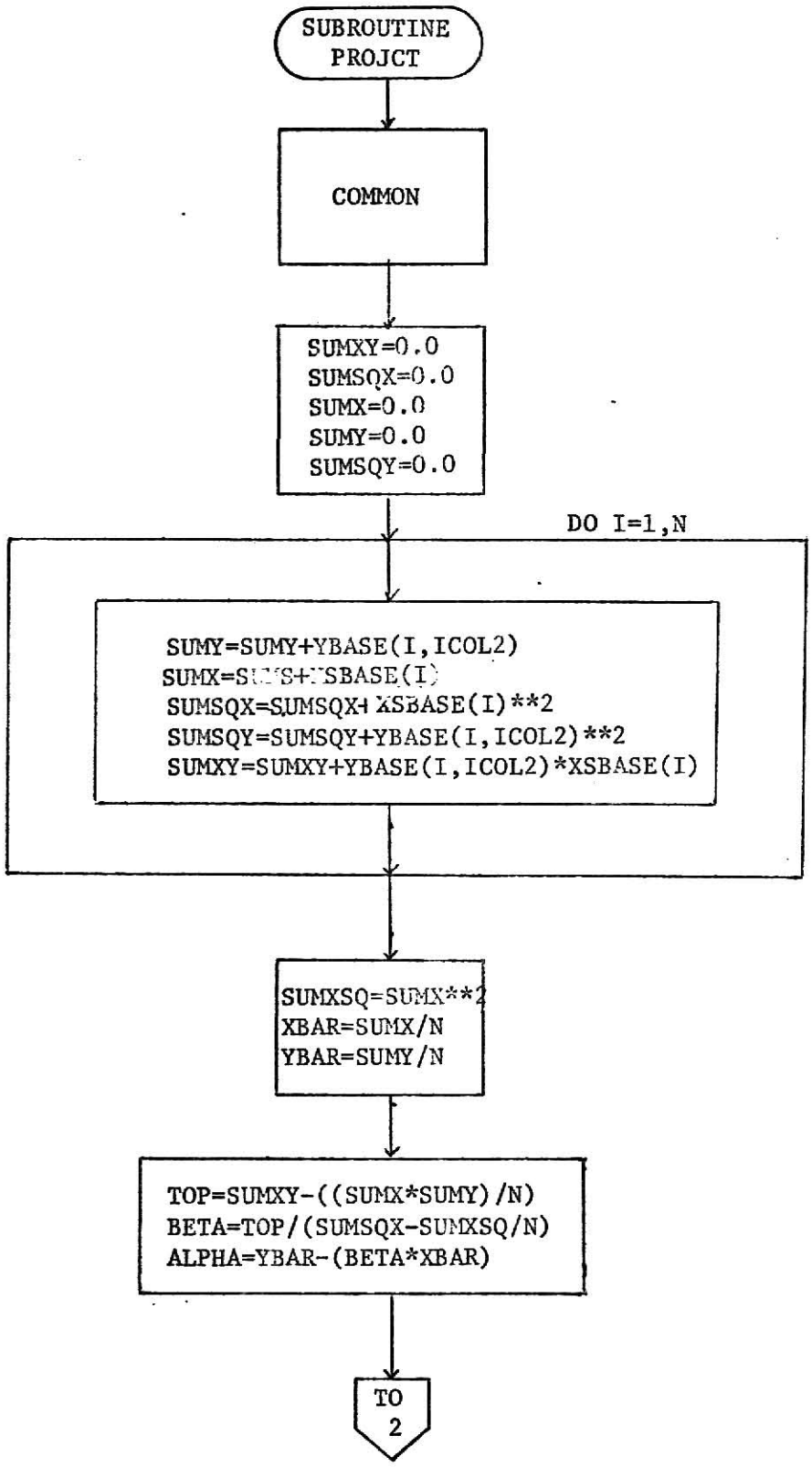
This subroutine fits a trend line to the base data input and uses this to extrapolate projected data for the number of years specified by the user in the main program. First, the trend line is fitted using normal values of Y and transformed values of X passed in by Regres.

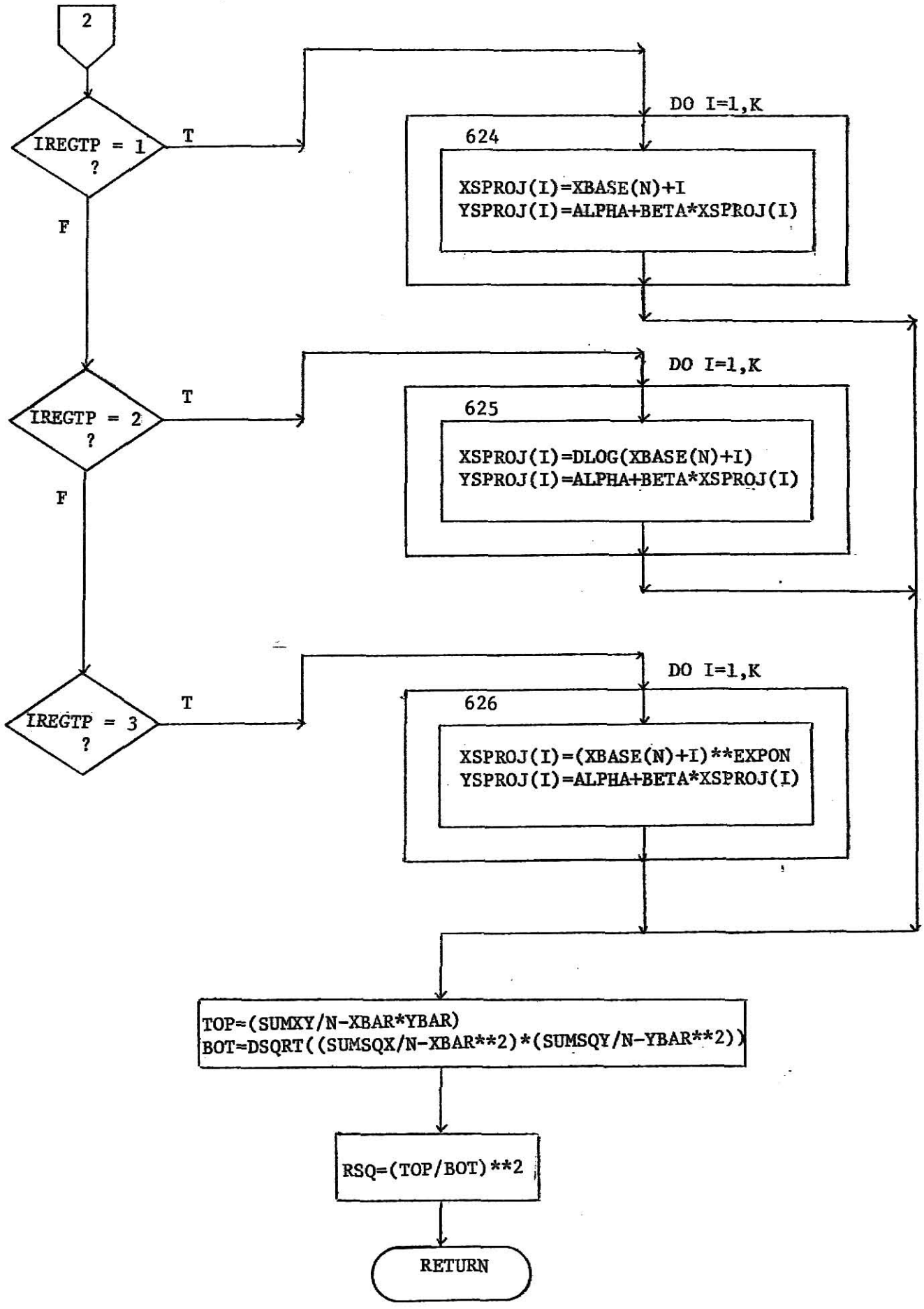
All computations for finding the trendline equation are done in double precision. Next, the projected values of Y are found by transforming future values of X in the same way that the base data X's were transformed and then plugging these future transformed X values into the calculated regression equation. Finally the r^2 of the trend line fit is calculated. Projct then returns to Regress which calls Prosto to store the projection made by Projct.

A flowchart of Projct follows a listing of it.

```
      SUBRCUTINE PROJCT                                                  AUT04920
C-----------------------------------------------------------------------AUT04930
C   THIS SUBROUTINE PRCJECTS INPUT CATA ACCORDING TO THE VALUE OF      |AUT04940
C   IREGTP.  IF:  IREGTP =1  ; THE PROJECTION IS LINEAR                |AUT04950
C                        2  ; THE PROJECTION IS LOGARITHMIC            |AUT04960
C                        3  ; THE PROJECTION IS EXPONENTIAL            |AUT04970
C                                                                      |AUT04980
C   THE LOG PRCJECTION IS CF THE FORM Y=A+B LOGX.  THE EXPONENTIAL     |AUT04990
C   PRCJECTICN IS OF THE FORM Y=A+BX**EXPCN; WHERE EXPON IS INPUT TO   |AUT05000
C   THE SUBRCUTINE AND LIES BETWEEN .4 AND 1.4, EXCLUDING 1.0.         |AUT05010
C-----------------------------------------------------------------------AUT05020
      REAL*8 YBASE,TCP,BCT,RSC,EXPCN                                     AUT05030
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR              AUT05040
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                 AUT05050
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT05060
     *ON,RSC,YSPROJ(60),ICCL2,IREGTP,K,N                                 AUT05070
C-----------------------------------------------------------------------AUT05080
C   PASS IN:  XSBASE(40),YBASE(40,26),L,N,K,IREGTP,ICOL2               |AUT05090
C                                                                      |AUT05100
C   PASS OUT: YSPRCJ(60),ALPHA,BETA,YBAR,RSQ                           |AUT05110
C                                                                      |AUT05120
C   WHERE:  XBASE = LOW RANGE EQUIVALENTS OF XYEAR AS CONVERTED BY     |AUT05130
C                   SUBRCUTINE MINVAL, THE X'S                         |AUT05140
C           YBASE = THE Y'S, ONE CCLUMN PER CCUNTY                     |AUT05150
C          YSPROJ = THE PRCJECTED Y'S, THE SAME VECTOR IS REUSED FOR   |AUT05160
C                   ALL CCUNTIES AFTER THE RESULTS ARE TRANSFERRED TO  |AUT05170
C                   TO MATRIX BESPRO VIA SUBROUTINE PROSTO             |AUT05180
C               L = THE NUMBER OF COUNTIES                             |AUT05190
C               N = THE NUMBER OF YEARS OF BASE CATA                   |AUT05200
C               K = THE NUMBER OF YEARS TO APPLY THE REGRESSION EQUA-  |AUT05210
C                   TICN TC AND THUS MAKE PRCJECTICNS FOR              |AUT05220
C          IREGTP = KEY TO THE TYPE CF REGRESSICN TO BE PERFORMED      |AUT05230
C           EXPCN = EXPCNENT TO BE USED IN AN EXPONENTIAL REGRESSION   |AUT05240
C           ICOL2 = PCINTER SHCWING WHICH OF THE COLUMNS IN YSBASE     |AUT05250
C                   (WHICH CCUNTY) IS BEING REGRESSED CR PRINTED       |AUT05260
C          XSBASE = VECTCR CCNTAINING TRANSFCRMED X'S, SUCH AS LOG X   |AUT05270
C          XSPRCJ = VECTOR CCNTAINING PRCJECTED X'S, USED TO CALCULATE |AUT05280
C                   THE PRCJECTED Y'S BY APPLYING THE REGRESSION       |AUT05290
C                   EQUATICN                                           |AUT05300
C-----------------------------------------------------------------------AUT05310
      SUMXY=0.000000000                                                  AUT05320
      SUMSQX=0.000000000                                                 AUT05330
      SUMX=0.000000000                                                   AUT05340
      SUMY=0.000000000                                                   AUT05350
      SUMSQY=0.00000000                                                  AUT05360
C--------------------------------                                       AUT05370
C   FIT TREND LINE TO BASE DATA.|                                      AUT05380
C--------------------------------                                       AUT05390
  623 DO 600 I=1,N                                                       AUT05400
      SUMY=SUMY+YBASE(I,ICCL2)                                           AUT05410
      SUMX=SUMX+XSBASE(I)                                                AUT05420
      SUMSQX=SUMSQX+XSBASE(I)**2                                         AUT05430
      SUMSQY=SUMSQY+YBASE(I,ICOL2)**2                                    AUT05440
  600 SUMXY=SUMXY+YBASE(I,ICOL2)*XSBASE(I)                               AUT05450
      SUMXSQ=SUMX**2                                                     AUT05460
      XBAR=SUMX/N                                                        AUT05470
      YBAR=SUMY/N                                                        AUT05480
      TOP=SUMXY-((SUMX*SUMY)/N)                                          AUT05490
      BETA=TOP/(SUMSQX-SUMXSQ/N)                                         AUT05500
      ALPHA=YBAR-(BETA*XBAR)                                             AUT05510
```

```
C-----------------------------                                        AUT05520
C   PROJECT BASE DATA K YEARS.|                                      AUT05530
C-----------------------------                                        AUT05540
      GO TO (624,625,626),IREGTP                                     AUT05550
  624 DO 709 I=1,K                                                   AUT05560
      XSPROJ(I)=XBASE(N)+I                                           AUT05570
  709 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)                                 AUT05580
      GO TO 640                                                      AUT05590
  626 DO 710 I=1,K                                                   AUT05600
      XSPROJ(I)=(XBASE(N)+I)**EXPON                                  AUT05610
  710 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)                                 AUT05620
      GO TO 640                                                      AUT05630
  625 DO 601 I=1,K                                                   AUT05640
      XSPROJ(I)=DLOG(XBASE(N)+I)                                     AUT05650
  601 YSPROJ(I)=ALPHA+BETA*XSPROJ(I)                                 AUT05660
C-----------------------------------------                           AUT05670
C   THIS SECTION COMPUTES THE VALUE OF RSQ.|                         AUT05680
C-----------------------------------------                           AUT05690
  640 TOP=(SUMXY/N-XBAR*YBAR)                                        AUT05700
      BOT=DSQRT((SUMSQX/N-XBAR**2)*(SUMSQY/N-YBAR**2))               AUT05710
      RSQ=(TOP/BOT)**2                                               AUT05720
  998 RETURN                                                         AUT05730
      END                                                            AUT05740
```


SUBROUTINE
PROJCT
COMMON
SUMXY=0.0
SUMSQX=0.0
SUMX=0.0
SUMY=0.0
SUMSQY=0.0
DO I=1,N
SUMY=SUMY+YBASE(I,ICOL2)
SUMX=SUMX+XSBASE(I)
SUMSQX=SUMSQX+XSBASE(I)**2
SUMSQY=SUMSQY+YBASE(I,ICOL2)**2
SUMXY=SUMXY+YBASE(I,ICOL2)*XSBASE(I)
SUMXSQ=SUMX**2
XBAR=SUMX/N
YBAR=SUMY/N
TOP=SUMXY-((SUMX*SUMY)/N)
BETA=TOP/(SUMSQX-SUMXSQ/N)
ALPHA=YBAR-(BETA*XBAR)
TO
2


2
IREGTP = 1 ?
T
F
DO I=1,K
624
XSPROJ(I)=XBASE(N)+I
YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
IREGTP = 2 ?
T
F
DO I=1,K
625
XSPROJ(I)=DLOG(XBASE(N)+I)
YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
IREGTP = 3 ?
T
DO I=1,K
626
XSPROJ(I)=(XBASE(N)+I)**EXPON
YSPROJ(I)=ALPHA+BETA*XSPROJ(I)
TOP=(SUMXY/N-XBAR*YBAR)
BOT=DSQRT((SUMSQX/N-XBAR**2)*(SUMSQY/N-YBAR**2))
RSQ=(TOP/BOT)**2
RETURN

VIII. Subroutine Minval

The subroutine replaces missing input data if requested, and automatically converts the input years to a form acceptable for regression. If IZERO, read in the main program, was YES then this subroutine replaces missing data or zeros in the input data. For example, if the data for 1950 and 1952 is available but not for 1951, and the replacement section of Minval was turned on (via IZERO=YES) then the program would calculate values for the missing data and fill it in before performing the regressions. The actual methods used to calculate the missing numbers are detailed in the comments on the Minval listing. Because of the way Minval calculates the missing data by looking at the data for the year ahead and behind it, no county can have two consecutive year's data missing or Minval will replace them incorrectly.

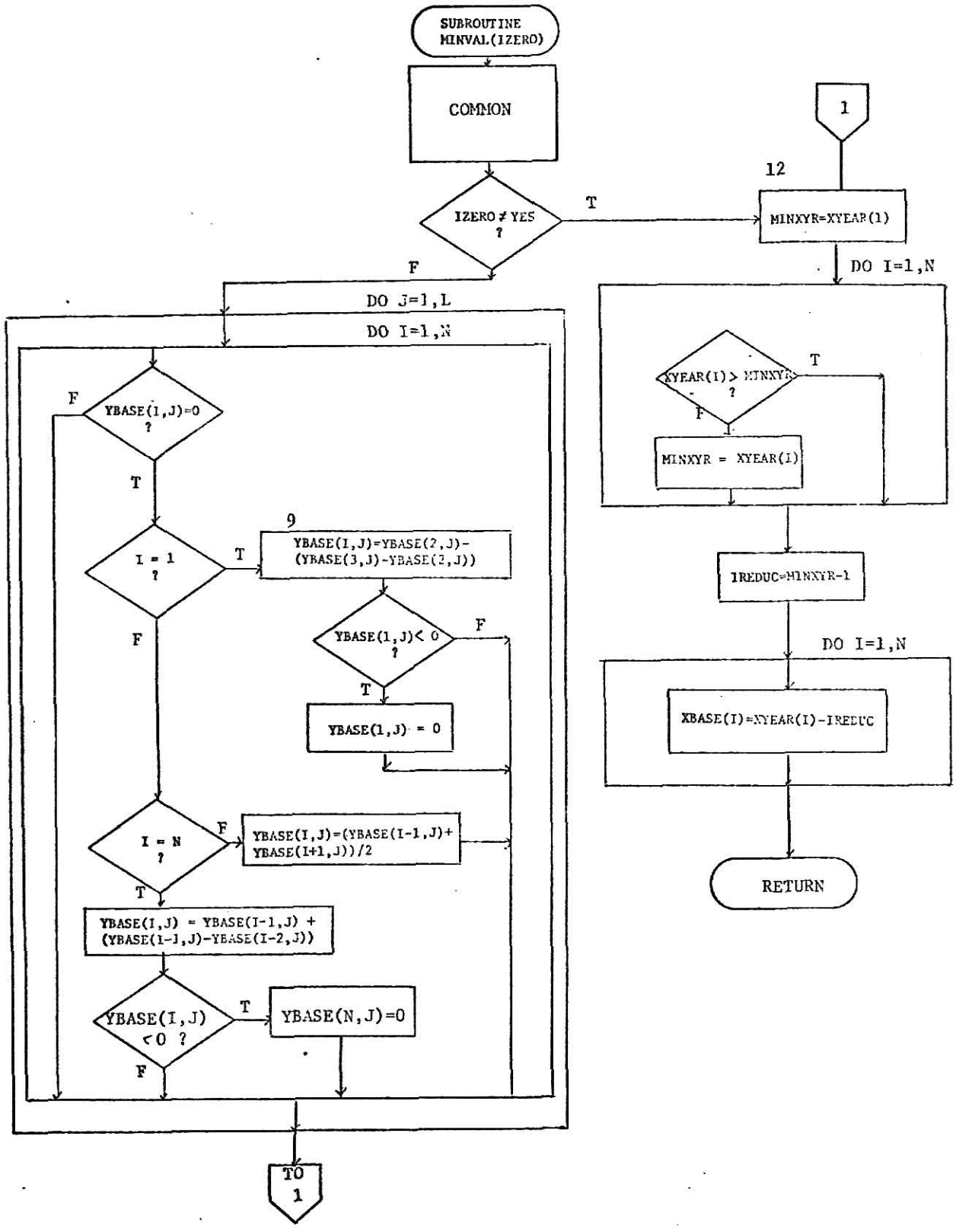
Minval also scales the input years for regression. The input years (XYEAR) are adjusted to integer numbers starting with 1. This is done so that the X and Y values used in the various regressions will be scaled correctly in relation to each other. The scaling is done by finding the smallest year in the input years (called MINXYR). This number minus one (called IREDUC) is then subtracted from each value in XYEAR and the scaled results are stored in vector XBASE.

The method of reducing the years to smaller scaled values allows the use of years that are greater than one year more than the previous. For example: 1930, 1927, 1945, 1970 is a perfectly acceptable stream of input years.

A flowchart of subroutine Minval follows a listing of it.

```
      SUBROUTINE MINVAL(IZERO)                                          AUT05750
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT05760
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR             AUT05770
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT05780
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT05790
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                AUT05800
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                AUT05810
      INTEGER*4 XYEAR,XSYEAR                                            AUT05820
      INTEGER*4 YES/'YES'/                                              AUT05830
C-----------------------------------------------------------------------AUT05840
C   THIS SECTION REPLACES ZEROS IN THE BASE DATA.  IF THE MISSING       |AUT05850
C   DATA IS THE FIRST YEAR'S DATA, IT IS REPLACED WITH THE SECOND       |AUT05860
C   YEAR'S DATA MINUS THE ABSOLUTE VALUE OF THE DIFFERENCE BETWEEN THE  |AUT05870
C   SECOND AND THIRD YEAR'S DATA.  IF THIS IS LESS THAN ZERO, THE FIRST |AUT05880
C   YEAR IS LEFT AS ZERO.                                               |AUT05890
C   IF THE MISSING DATA IS THE LAST YEAR, IT IS REPLACED WITH THE NEXT  |AUT05900
C   TO LAST YEAR'S DATA PLUS THE DIFFERENCE BETWEEN THE SECOND TO LAST  |AUT05910
C   AND THE THIRD TO LAST YEAR'S DATA.                                  |AUT05920
C   IF ANY OTHER YEAR IS MISSING, THE AVERAGE OF THE YEARS BEFORE AND   |AUT05930
C   AFTER IT IS USED IN PLACE OF THE MISSING DATA.                      |AUT05940
C   IF TWO YEAR'S DATA IN A ROW ARE MISSING, THE USER MUST CALCULATE    |AUT05950
C   ONE OF THEM AS THIS ROUTINE WILL ONLY REPLACE THE SECOND ONE.       |AUT05960
C-----------------------------------------------------------------------AUT05970
C***READ IN IZERO IN MAIN/ IF YES,TEST FOR ZEROS, IF NO SKIP THIS SECT.|AUT05980
C-----------------------------------------------------------------------AUT05990
      IF(IZERO.NE.YES) GO TO 12                                         AUT06000
      DO 5 J=1,L                                                        AUT06010
      DO 5 I=1,N                                                        AUT06020
      IF(YBASE(I,J).EQ.0) GO TO 8                                       AUT06030
      GO TO 5                                                           AUT06040
    8 IF(I.EQ.1) GO TO 9                                                AUT06050
      IF(I.EQ.N) GO TO 10                                               AUT06060
      YBASE(I,J)=(YBASE(I-1,J)+YBASE(I+1,J))/2                          AUT06070
      GO TO 5                                                           AUT06080
    9 YBASE(1,J)=YBASE(2,J)-(YBASE(3,J)-YBASE(2,J))                     AUT06090
      IF(YBASE(1,J).LT.0) YBASE(1,J)=0                                  AUT06100
      GO TO 5                                                           AUT06110
   10 YBASE(I,J)=YBASE(I-1,J)+(YBASE(I-1,J)-YBASE(I-2,J))               AUT06120
      IF(YBASE(N,J).LT.0) YBASE(N,J)=0                                  AUT06122
    5 CONTINUE                                                          AUT06130
C-----------------------------------------------------------------------AUT06140
C   THIS SECTION TRANSFORMS THE INPUT YEARS (XYEAR) INTO INTEGERS       |AUT06150
C   STARTING WITH 1 SO THE REGRESSION INPUT VALUES WILL BE SCALED COR-  |AUT06160
C   RECTLY.  THE SCALED RESULTS ARE PUT INTO XBASE.                     |AUT06170
C-----------------------------------------------------------------------AUT06180
   12 MINXYR=XYEAR(1)                                                   AUT06190
      DO 6 I=1,N                                                        AUT06200
      IF(XYEAR(I).GT.MINXYR) GO TO 6                                    AUT06210
      MINXYR=XYEAR(I)                                                   AUT06220
    6 CONTINUE                                                          AUT06230
      IREDUC=MINXYR-1                                                   AUT06240
      DO 7 I=1,N                                                        AUT06250
    7 XBASE(I)=XYEAR(I)-IREDUC                                          AUT06260
      RETURN                                                            AUT06270
      END                                                               AUT06280
```

Subroutine Minval Flowchart



IX. Subroutine Idist

This subroutine controls the printing of output from the program. This is done via setting controlling parameters and then calling subroutine Ititle to print title blocks on each page, subroutine Istat to print a statistics block at the end of each projection set, and subroutine Idata to print the base and projected data sets. Idata automatically compensates for districts with enough counties to require two pages in width to print them and adjusts lengthwise for any size base and projected data sets, up to the maximums.

Idist begins by calculating the total numberof lines of data to be printed. This is found by dividing the number of base years by five to give the number of blank lines to be inserted between them (IBSEC). Next, this is repeated for the projected data set. (IPSEC) The sum of those plus N (the number of years of base data) plus K (the number of years of projected data) plus one (to allow for the two lines between the base and projected data sets as IBSEC already allows for one) is equal to the total number of data lines of output. (TOTLIN)

Idist is designed so that the statistics block will never be printed on a page unless all 10 lines of the statistic block can be printed on that page. If the statistics block must be printed on another page, all by itself, the title block is printed first, but the year in the county name line is omitted. The year is dropped via Ititle when the flag IPRTDT is set to 1 by Idata. From the placement of the statistics block and the total number of lines of output, Idata determines the number of pages in length each projection set will be. This is determined using the facts that each page can contain up to 60 lines, the title block is 12 lines long, and the statistics block contains 10 lines. The logic used is shown in the second comment box in the subroutine listing.

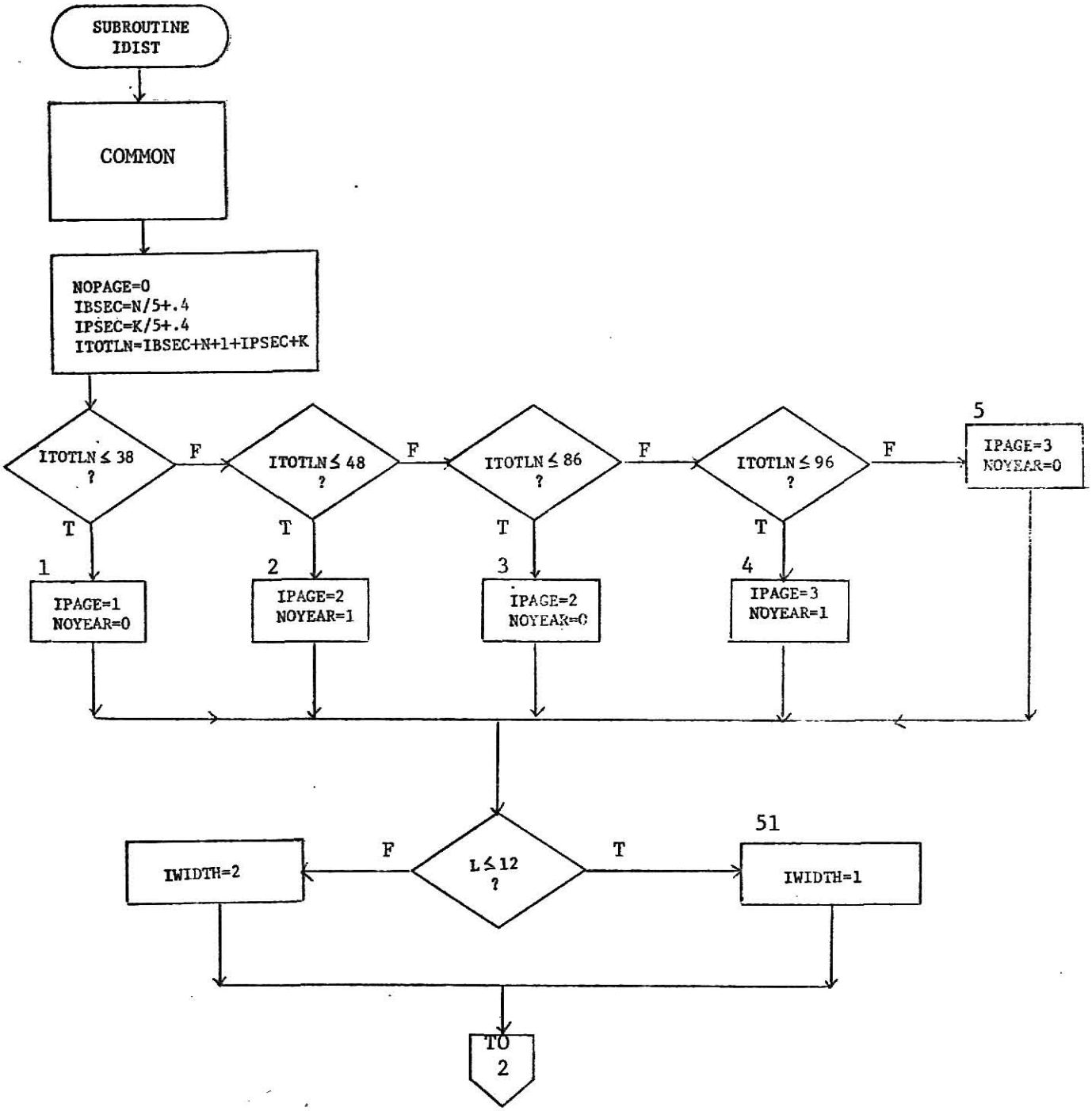
Once the number of pages in length a given projection are determined, the number of pages in width (1 or 2) are determined. If the number of counties is 12 or less, they and the district totals will all fit on one page in width. Otherwise, two pages in width must be used. If this is the case, the 2nd pages "in width" print after the pages making up the 1st page "in width." From the number of pages in width and the number of counties, IQUIT, IEND, and ISTART are set to control which counties are printed on which page in width.

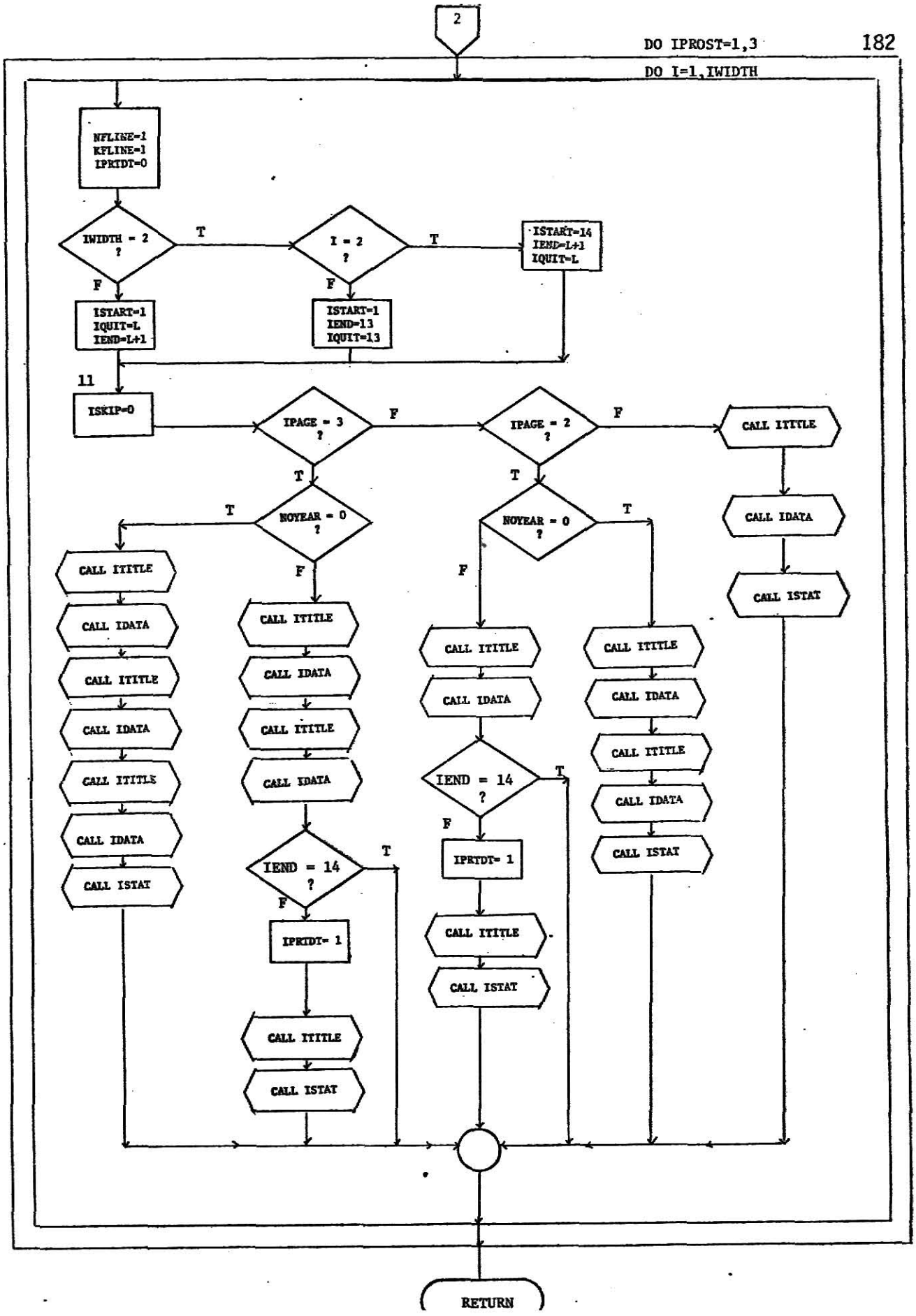
From the values of IPAGE, NOYEAR, and IWIDTH, Idist determines the sequence in which to call Ititle, Idata and Istat. The second page in width is run after the first, before the next projection set is printed, with the values of IQUIT, IEND, and ISTART changed appropriately so the correct counties will be printed.

```
      SUBROUTINE IDIST                                                  AUT06290
C-----------------------------------------------------------------------AUT06300
C   THIS SUBROUTINE CONTROLS THE PRINTING OF OUTPUT VIA THE CALLING     |AUT06310
C   OF SUBROUTINES ITITLE,IDATA, AND ISTAT.                             |AUT06320
C-----------------------------------------------------------------------AUT06330
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT06340
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSCX,XBAR             AUT06350
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT06360
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT06370
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                AUT06380
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)          AUT06390
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                AUT06400
      COMMON/C/IDISTR,IPRTOT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT06410
     1E5(31),LINE6(31),IQUIT                                            AUT06420
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPROST,ISKIP                   AUT06430
      COMMON/$PAGE/NOPAGE                                               AUT06440
      INTEGER XYEAR,XSYEAR                                              AUT06450
C-----------------------------------------------------------------------AUT06460
C       K = NUMBER OF YEARS OF DATA TO PROJECT. (MAX=60)                |AUT06470
C       L = NUMBER OF COUNTIES PER DISTRICT (MAX=25)                    |AUT06480
C       N = NUMBER OF BASE YEARS OF DATA (MAX=40)                       |AUT06490
C   IPAGE = NUMBER OF PAGES IN LENGTH 1,2,OR 3                          |AUT06500
C   IWIDTH= NUMBER OF PAGES IN WIDTH  1 OR 2                            |AUT06510
C   NOYEAR= 0 = PRINT STATISTICS AT THE BOTTOM OF THE LAST PAGE.        |AUT06520
C           1 = PRINT STATISTICS AT THE TOP OF THE LAST PAGE.           |AUT06530
C   IBSEC = NUMBER OF SPACES BETWEEN 5 YEAR GROUPS IN THE BASE DATA     |AUT06540
C   IPSEC = NUMBER OF SPACES BETWEEN 5 YEAR GROUPS IN THE PROJECTED     |AUT06550
C           DATA                                                        |AUT06560
C   ITOTLN= TOTAL NUMBER OF OUTPUT LINES EXCLUSIVE OF TITLE BLOCKS AND  |AUT06570
C           STATISTICS                                                  |AUT06580
C   NFLINE= FIRST LINE OF BASE DATA TO BE PRINTED WHEN IDATA IS CALLED  |AUT06590
C   KFLINE= FIRST LINE OF THE PROJECTED DATA TO BE PRINTED WHEN         |AUT06600
C           IDATA IS CALLED                                             |AUT06610
C   IPRTOT= SIGNALS ITITLE TO OMIT PRINTING THE YEAR IN THE TITLE BLOCK |AUT06620
C   ISKIP = SIGNAL TO IDATA THAT BASE DATA WAS ALREADY PRINTED ON A     |AUT06630
C           PREVIOUS CALL AND TO SKIP PRINTING IT THIS TIME             |AUT06640
C   IPROST= PROJECTION SET NUMBER FROM 1 TO 3 WHERE 1 IS THE BEST       |AUT06650
C           (BASED ON RSQ'S)                                            |AUT06660
C   ISTART= NUMBER OF THE COUNTY THAT THE OUTPUT PAGE STARTS WITH.      |AUT06670
C           (COLUMN NO. OF MATRICES BESPRO, OR DOPMAT WHEN USED IN      |AUT06680
C           SUBROUTINE ISTAT.)                                          |AUT06690
C     IEND= NUMBER OF COLUMN IN BESPRO THAT OUTPUT PAGE ENDS WITH.      |AUT06700
C           (= L+1 ON THE LAST PAGE IN WIDTH, TO ALLOW FOR THE DIS-     |AUT06710
C           TOTAL COLUMN.                                               |AUT06720
C    IQUIT= NUMBER OF THE COLUMN OF MATRIX DOPMAT THE OUTPUT PAGE'S     |AUT06730
C           STATISTIC BLOCK ENDS WITH.  (= L ON THE LAST PAGE IN WIDTH  |AUT06740
C           AS THE DISTRICT COLUMN HAS NO STATISTICS.                   |AUT06750
C   NOPAGE= FLAG FOR ITITLE NOT TO DO PAGING AS THE LAST PAGE OF OUT-   |AUT06760
C           PUT WAS FILLED AND CAUSED PAGING BY THE SYSTEM.             |AUT06770
C-----------------------------------------------------------------------AUT06780
      NOPAGE=0                                                          AUT06790
      IBSEC=N/5+.4                                                      AUT06800
      IPSEC=K/5+.4                                                      AUT06810
      ITOTLN=IBSEC+N+1+IPSEC+K                                          AUT06820
      IF(ITOTLN.LE.38) GO TO 1                                          AUT06830
      IF(ITOTLN.LE.48) GO TO 2                                          AUT06840
      IF(ITOTLN.LE.86) GO TO 3                                          AUT06850
      IF(ITOTLN.LE.96) GO TO 4                                          AUT06860
      GO TO 5                                                           AUT06870
C-----------------------------------------------------------------------AUT06880
```

```
C    IPAGE = 60 LINES   TITLE BLOCK = 12 LINES   STATISTICS = 10 LINES  |AUT06990
C    48 LINES/PAGE+TITLE BLOCK  38 LINES/PAGE+TITLE BLOCK+STATISTICS    |AUT06995
C                                                                       |AUT06910
C    ITOTLN.LE.38 = PRINT 1 PAGE WITH STATISTICS AT BOTTOM              |AUT06920
C    ITOTLN.LE.48 = PRINT 2ND PAGE WITH ONLY TITLE BLOCK (WITHOUT YEAR  |AUT06930
C                   IN IT) AND STATISTICS.                              |AUT06940
C    ITOTLN.LE.86 = PRINT 2 PAGES WITH STATISTICS AT BOTTOM OF 2ND PAGE |AUT06950
C    ITOTLN.LE.96 = PRINT 3RD PAGE WITH ONLY TITLE BLOCK (WITHOUT YEAR  |AUT06960
C                   IN IT) AND STATISTICS.                              |AUT06970
C    TO MAX OF 121= PRINT 3 PAGES WITH STATISTICS AT BOTTOM OF 3RD PAGE |AUT06980
C-------------------------------------------------------------------AUT06990
     1 IPAGE=1                                                         AUT07000
       NOYEAR=0                                                        AUT07010
       GO TO 50                                                        AUT07020
     2 IPAGE=2                                                         AUT07030
       NOYEAR=1                                                        AUT07040
       GO TO 50                                                        AUT07050
     3 IPAGE=2                                                         AUT07060
       NOYEAR=0                                                        AUT07070
       GO TO 50                                                        AUT07080
     4 IPAGE=3                                                         AUT07090
       NOYEAR=1                                                        AUT07100
       GO TO 50                                                        AUT07110
     5 IPAGE=3                                                         AUT07120
       NOYEAR=0                                                        AUT07130
    50 IF(L.LE.12) GO TO 51                                            AUT07140
       IWIDTH=2                                                        AUT07150
       GO TO 32                                                        AUT07160
    51 IWIDTH=1                                                        AUT07170
C-------------------------------------------------                     AUT07180
C    DO THE FOLLOWING ONCE FOR EACH PROJECTION SET |                   AUT07190
C    AND ONCE FOR EACH PAGE IN WIDTH.              |                   AUT07200
C-------------------------------------------------                     AUT07210
    32 DO 30 IPROST=1,3                                                AUT07220
       DO 30 I=1,IWIDTH                                                AUT07230
       NFLINE=1                                                        AUT07240
       KFLINE=1                                                        AUT07250
       IPRTOT=0                                                        AUT07260
       IF(IWIDTH.EQ.2) GO TO 10                                        AUT07270
       ISTART=1                                                        AUT07280
       IQUIT=L                                                         AUT07290
       IEND=L+1                                                        AUT07300
       GO TO 11                                                        AUT07310
    10 IF(I.EQ.2) GO TO 12                                             AUT07320
       ISTART=1                                                        AUT07330
       IEND=13                                                         AUT07340
       IQUIT=13                                                        AUT07350
       GO TO 11                                                        AUT07360
    12 ISTART=14                                                       AUT07370
       IEND=L+1                                                        AUT07380
       IQUIT=L                                                         AUT07390
    11 ISKIP=0                                                         AUT07400
       IF(IPAGE.EQ.3) GO TO 6                                          AUT07410
       IF(IPAGE.EQ.2) GO TO 7                                          AUT07420
C--------------------                                                  AUT07430
C    IPAGE=1,NOYEAR=0|                                                 AUT07440
C--------------------                                                  AUT07450
       CALL ITITLE                                                     AUT07460
       CALL IDATA                                                      AUT07470
       CALL ISTAT                                                      AUT07480
```

```
      GO TO 30                                                          AUT07490
    7 IF(NCYEAR.EQ.0) GO TO 8                                           AUT07500
C-------------------                                                    AUT07510
C   IPAGE=2,NOYEAR=1|                                                  AUT07520
C-------------------                                                    AUT07530
      CALL ITITLE                                                       AUT07540
      CALL IDATA                                                        AUT07550
C-----------------------------------------------------------------------AUT07552
C   DO NOT PRINT STATISTIC BLOCK IF DISTRICT IS THE ONLY COLUMN ON THE |AUT07554
C   PAGE AND ONLY THE DISTRICT'S STAT BLOCK WOULD BE PRINTED.          |AUT07555
C-----------------------------------------------------------------------AUT07556
      IF(IEND.EQ.14) GO TO 30                                           AUT07557
      IPRTOT=1                                                          AUT07560
      CALL ITITLE                                                       AUT07570
      CALL ISTAT                                                        AUT07580
      GO TO 30                                                          AUT07590
C-------------------                                                    AUT07600
C   IPAGE=2,NOYEAR=0|                                                  AUT07610
C-------------------                                                    AUT07620
    8 CALL ITITLE                                                       AUT07630
      CALL IDATA                                                        AUT07640
      CALL ITITLE                                                       AUT07650
      CALL IDATA                                                        AUT07660
      CALL ISTAT                                                        AUT07670
      GO TO 30                                                          AUT07680
C-------------------                                                    AUT07690
C   IPAGE=3,NCYEAR=1|                                                  AUT07700
C-------------------                                                    AUT07710
    6 IF(NOYEAR.EQ.0) GO TO 9                                           AUT07720
      CALL ITITLE                                                       AUT07730
      CALL IDATA                                                        AUT07740
      CALL ITITLE                                                       AUT07750
      CALL IDATA                                                        AUT07760
      IF(IEND.EQ.14) GO TO 30                                           AUT07767
      IPRTOT=1                                                          AUT07770
      CALL ITITLE                                                       AUT07780
      CALL ISTAT                                                        AUT07790
      GO TO 30                                                          AUT07800
C-------------------                                                    AUT07810
C   IPAGE=3,NCYEAR=0|                                                  AUT07820
C-------------------                                                    AUT07830
    9 CALL ITITLE                                                       AUT07840
      CALL IDATA                                                        AUT07850
      CALL ITITLE                                                       AUT07860
      CALL IDATA                                                        AUT07870
      CALL ITITLE                                                       AUT07880
      CALL IDATA                                                        AUT07890
      CALL ISTAT                                                        AUT07900
   30 CONTINUE                                                          AUT07910
      RETURN                                                            AUT07920
      END                                                               AUT07930
```


SUBROUTINE IDIST
COMMON
NOPAGE=0
IBSEC=N/5+.4
IPSEC=K/5+.4
ITOTLN=IBSEC+N+1+IPSEC+K
ITOTLN ≤ 38 ?
F
T
1
IPAGE=1
NOYEAR=0
ITOTLN ≤ 48 ?
F
T
2
IPAGE=2
NOYEAR=1
ITOTLN ≤ 86 ?
F
T
3
IPAGE=2
NOYEAR=0
ITOTLN ≤ 96 ?
F
T
4
IPAGE=3
NOYEAR=1
5
IPAGE=3
NOYEAR=0
L ≤ 12 ?
F
IWIDTH=2
T
51
IWIDTH=1
TO 2


2
DO IPROST=1,3
DO I=1,IWIDTH
NFLINE=1
KFLINE=1
IPRTDT=0
IWIDTH = 2 ?
T
F
I = 2 ?
T
F
ISTART=14
IEND=L+1
IQUIT=L
ISTART=1
IQUIT=L
IEND=L+1
ISTART=1
IEND=13
IQUIT=13
11
ISKIP=0
IPAGE = 3 ?
F
IPAGE = 2 ?
F
CALL ITITLE
CALL IDATA
CALL ISTAT
T
NOYEAR = 0 ?
T
F
CALL ITITLE
CALL IDATA
CALL ITITLE
CALL IDATA
CALL ITITLE
CALL IDATA
CALL ISTAT
CALL ITITLE
CALL IDATA
CALL ITITLE
CALL IDATA
IEND = 14 ?
T
F
IPRTDT= 1
CALL ITITLE
CALL ISTAT
CALL ITITLE
CALL IDATA
CALL ITITLE
CALL IDATA
CALL ISTAT
RETURN

X. Subroutine Ititle

This subroutine prints the title block at the top of each page of output. The titles were centered earlier by calls to subroutine Center mode in Main. Depending of the value of IPROST, each title gives the number of the projection set that follows it where ONE denotes the set containing the best projections in terms of r^2 and so on.

The district name read in the main is also put at the top of each page of output. If the output page contains the first 13 counties (or less) of a district the first half of LINE4 is printed above them, as it contains the names for those counties. Likewise, the second half of LINE4 is printed above counties 14 thru 25 (2nd page in width).

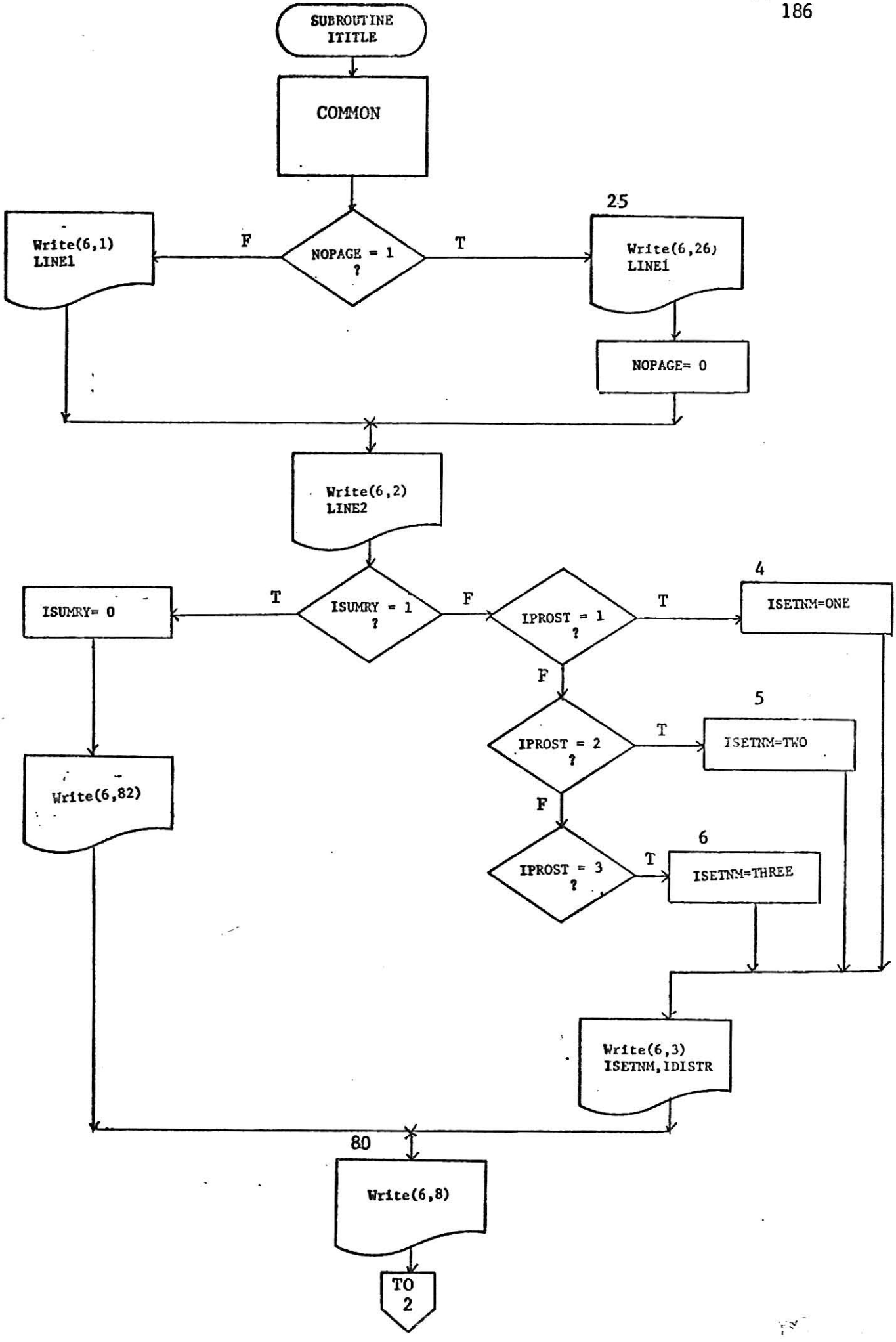
The flag NOPAGE controls the printing of the first title under different formats. If NOPAGE is zero, a format is used that causes paging before the title block is printed. If NOPAGE is 1, the format used does not cause paging.

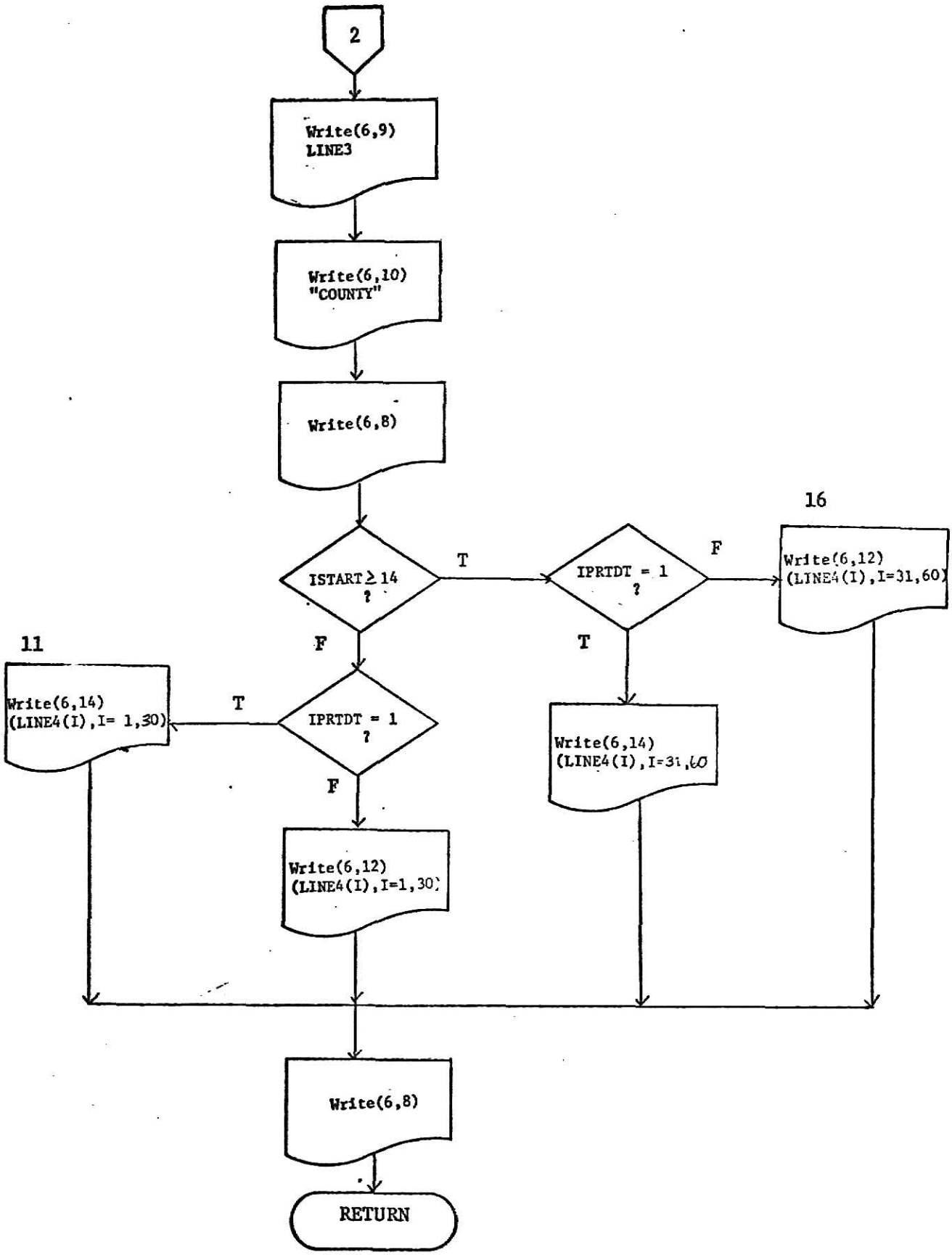
The flag ISUMRY controls the printing of the line containing the projection set and district numbers. If ISUMRY is 1, which it is when Ititle is called from subroutine Summry, the projection set and district number line is omitted. Otherwise, when ISUMRY=0, the line is printed.

A flowchart of subroutine Ititle follows a listing of it.

```
      SUBROUTINE ITITLE                                                 AUT07940
C-----------------------------------------------------------------------AUT07950
C   THIS SUBROUTINE PRINTS THE TITLE BLOCK ON EACH PAGE OF OUTPUT.      |AUT07960
C-----------------------------------------------------------------------AUT07970
      COMMON/C/IDISTR,IPRTOT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT07980
     1E5(31),LINE6(31),IQUIT                                            AUT07990
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPROST,ISKIP                   AUT08000
      COMMON/$PAGE/NOPAGE                                               AUT08010
      COMMON/$SUM/ISUMRY                                                AUT08020
      INTEGER*4 ONE/'1ST'/,TWO/'2ND'/,THREE/'3RD'/                      AUT08030
      IF(NOPAGE.EQ.1) GO TO 25                                          AUT08040
      WRITE(6,1) LINE1                                                  AUT08050
    1 FORMAT('1',33A4)                                                  AUT08060
      GO TO 20                                                          AUT08070
   25 WRITE(6,26) LINE1                                                 AUT08080
   26 FORMAT(1X,33A4)                                                   AUT08090
      NOPAGE=0                                                          AUT08100
   20 WRITE(6,2) LINE2                                                  AUT08110
    2 FORMAT('0',33A4)                                                  AUT08120
C---------------------------------                                      AUT08130
C   SELECT PROJECTION SET NUMBER. |                                     AUT08140
C---------------------------------                                      AUT08150
      IF(ISUMRY.EQ.1) GO TO 81                                          AUT08160
      GO TO (4,5,6),IPROST                                              AUT08170
    4 ISETNM=ONE                                                        AUT08180
      GO TO 7                                                           AUT08190
    5 ISETNM=TWO                                                        AUT08200
      GO TO 7                                                           AUT08210
    6 ISETNM=THREE                                                      AUT08220
    7 WRITE(6,3) ISETNM,IDISTR                                          AUT08230
    3 FORMAT('0',A3,' PROJECTION SET',102X,'DISTRICT ',A3)              AUT08240
      GO TO 80                                                          AUT08250
   81 ISUMRY=0                                                          AUT08260
      WRITE(6,82)                                                       AUT08270
   82 FORMAT(1X)                                                        AUT08280
   80 WRITE(6,8)                                                        AUT08290
    8 FORMAT(1X,132('-'))                                               AUT08300
      WRITE(6,9) LINE3                                                  AUT08310
    9 FORMAT(1X,33A4)                                                   AUT08320
      WRITE(6,10)                                                       AUT08330
   10 FORMAT('0',63X,'COUNTY')                                          AUT08340
      WRITE(6,8)                                                        AUT08350
C-------------------------------------------------------               AUT08360
C   IF ISTART > 14 = SECOND PAGE WIDTH OF OUTPUT SO USE  |              AUT08370
C   COUNTY NAMES IN SECOND HALF OF LINE4.                |              AUT08380
C-------------------------------------------------------               AUT08390
      IF(ISTART.GE.14) GO TO 15                                         AUT08400
C-------------------------------------------------------               AUT08410
C   WHEN IPRTOT = 1 DO NOT PRINT YEAR IN TITLE BLOCK.    |              AUT08420
C-------------------------------------------------------               AUT08430
      IF(IPRTOT.EQ.1) GO TO 11                                          AUT08440
      WRITE(6,12) (LINE4(I),I=1,30)                                     AUT08450
   12 FORMAT(4X,'YEAR',5X,30A4)                                         AUT08460
      GO TO 13                                                          AUT08470
   11 WRITE(6,14) (LINE4(I),I=1,30)                                     AUT08480
   14 FORMAT(13X,30A4)                                                  AUT08490
      GO TO 13                                                          AUT08500
   15 IF(IPRTOT.EQ.1) GO TO 16                                          AUT08510
      WRITE(6,12) (LINE4(I),I=31,60)                                    AUT08520
      GO TO 13                                                          AUT08530
```

```
16 WRITE(6,14) (LINE4(I),I=31,60)                                    AUT08540
13 WRITE(6,8)                                                        AUT08550
   RETURN                                                            AUT08560
   END                                                               AUT08570
```


SUBROUTINE ITITLE
COMMON
NOPAGE = 1 ?
F
T
Write(6,1) LINE1
25
Write(6,26) LINE1
NOPAGE= 0
Write(6,2) LINE2
ISUMRY = 1 ?
T
ISUMRY= 0
F
IPROST = 1 ?
T
4
ISETNM=ONE
F
IPROST = 2 ?
T
5
ISETNM=TWO
F
IPROST = 3 ?
T
6
ISETNM=THREE
Write(6,82)
Write(6,3) ISETNM,IDISTR
80
Write(6,8)
TO 2


2
Write(6,9)
LINE3
Write(6,10)
"COUNTY"
Write(6,8)
ISTART ≥ 14
?
T
IPRTDT = 1
?
F
16
Write(6,12)
(LINE4(I),I=31,60)
T
F
11
Write(6,14)
(LINE4(I),I= 1,30)
T
IPRTDT = 1
?
Write(6,14)
(LINE4(I),I=31,60
F
Write(6,12)
(LINE4(I),I=1,30)
Write(6,8)
RETURN

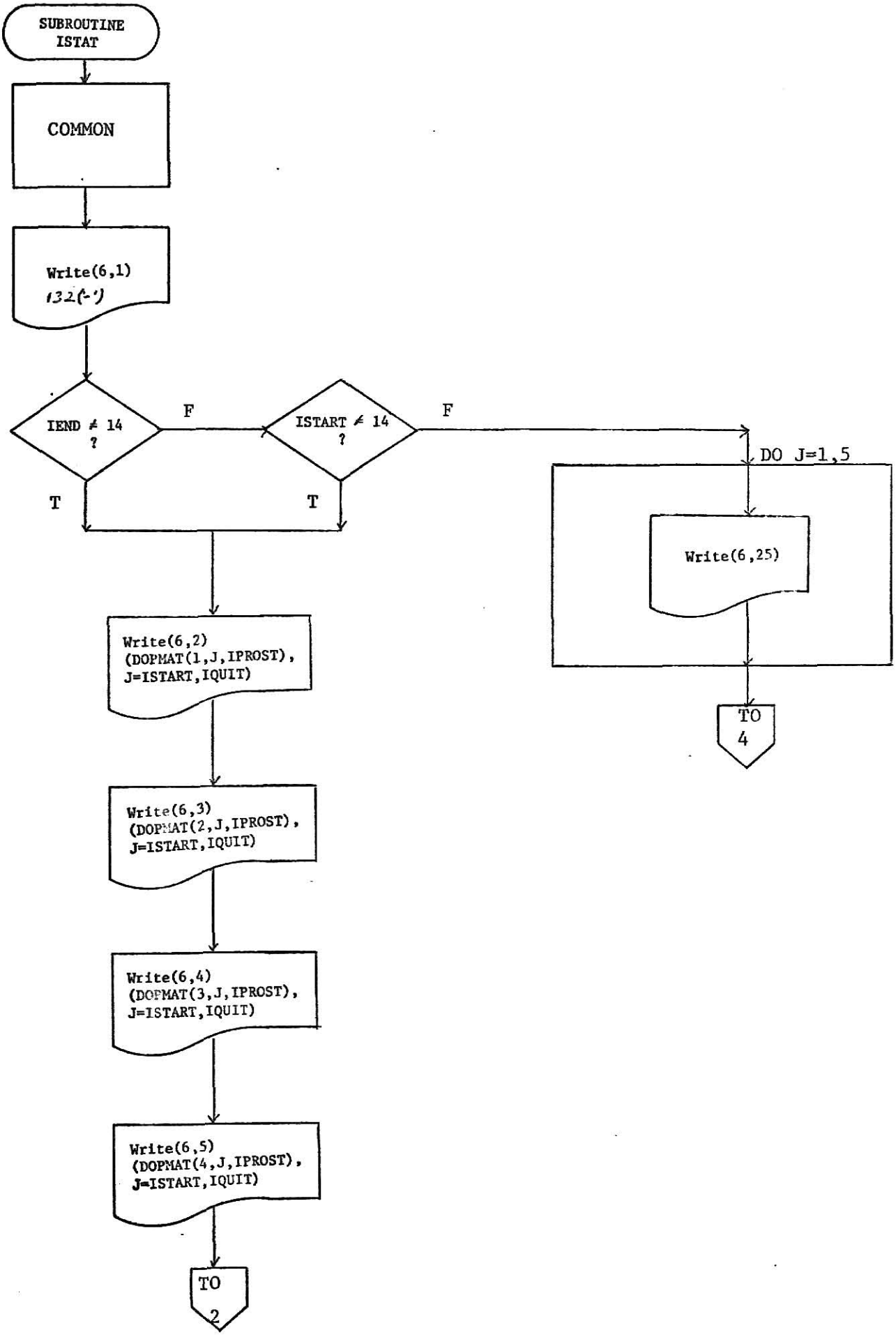
XI. Subroutine Istat

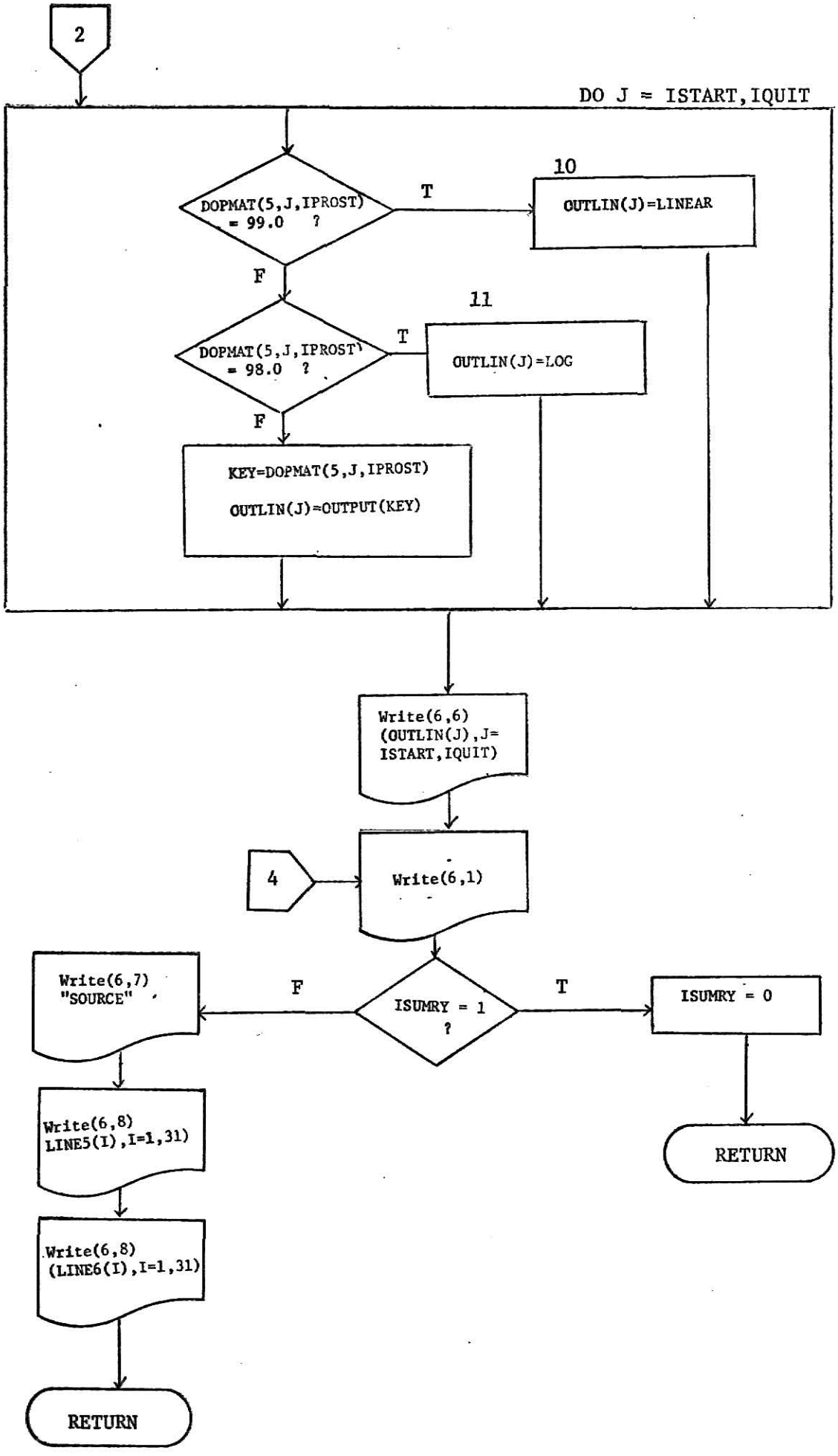
This subroutine prints the statistics block at the end of each set of projections. The statistics are generated by printing the contents of matrix dopmat. The last row of dopmat, which contains the projection type, is translated from a number into a character name, such as 1.4 EXP, before being printed. This is done by using the pronam number as a pointer to vector OUTPUT which contains the character name equivalents of the numbers. The character name is then assigned to vector OUTLIN which is printed in lieu of the fifth row of dopmat. If pronam is 99 or 98, it is translated directly by an if statement so that vector OUTPUT doesn't have to be 99 positions long.

After the projection type is printed, Istat prints two lines of footnotes. These are printed if ISUMRY equals zero. If not, when Istat is called from SUMMRY, the printing of the footnotes is skipped.

A flowchart of subroutine Istat follows a listing of it.

```
      SUBROUTINE ISTAT                                                  AUT08580
C-----------------------------------------------------------------------AUT08590
C   THIS SUBROUTINE PRINTS THE STATISTICS BLOCK AT THE END OF EACH      AUT08600
C   SET OF PROJECTIONS.                                                |AUT08610
C-----------------------------------------------------------------------AUT08620
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)          AUT08630
      COMMON/C/IDISTR,IPRTOT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT08640
     1E5(31),LINE6(31),IQUIT                                            AUT08650
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPROST,ISKIP                   AUT08660
      COMMON/$SUM/ISUMRY                                                AUT08680
      REAL*8 LINEAR/' LINEAR'/,LOG/'    LOG'/,OUTLIN(25)                AUT08690
      REAL*8 OUTPUT(16)/'0.1 EXP','0.2 EXP','0.3 EXP','0.4 EXP','0.5 EXPAUT08700
     1','0.6 EXP','0.7 EXP','0.8 EXP','0.9 EXP','1.0 EXP','1.1 EXP','1.2AUT08710
     2 EXP','1.3 EXP','1.4 EXP','1.5 EXP','1.6 EXP'/                    AUT08720
      WRITE(6,1)                                                        AUT08730
    1 FORMAT(1X,132('-'))                                               AUT08740
      IF(IEND.NE.14) GO TO 20                                           AUT08750
      IF(ISTART.NE.14) GO TO 20                                         AUT08760
      DO 26 J=1,5                                                       AUT08770
   26 WRITE(6,25)                                                       AUT08780
   25 FORMAT(1X)                                                        AUT08790
      GO TO 27                                                          AUT08800
   20 WRITE(6,2) (DOPMAT(1,J,IPROST),J=ISTART,IQUIT)                    AUT08810
    2 FORMAT(4X,'RSQ',6X,13(1X,F8.4))                                   AUT08820
      WRITE(6,3) (DOPMAT(2,J,IPROST),J=ISTART,IQUIT)                    AUT08830
    3 FORMAT(4X,'ALPHA',4X,13(1X,F8.2))                                 AUT08840
      WRITE(6,4) (DOPMAT(3,J,IPROST),J=ISTART,IQUIT)                    AUT08850
    4 FORMAT(4X,'BETA',5X,13(1X,F8.2))                                  AUT08860
      WRITE(6,5) (DOPMAT(4,J,IPROST),J=ISTART,IQUIT)                    AUT08870
    5 FORMAT(4X,'YBAR',5X,13(1X,F8.2))                                  AUT08880
C-----------------------------------------------------------------------AUT08890
C   THIS SECTION TRANSLATES PRONAM INTO A CHARACTER NAME.  THIS IS      |AUT08900
C   DONE BY USING THE VALUE OF PRONAM AS A POINTER TO AN ARRAY CON-     |AUT08910
C   TAINING THE CHARACTER NAME EQUIVALENTS OF THE NUMBERS.              |AUT08920
C-----------------------------------------------------------------------AUT08930
      DO 12 J=ISTART,IQUIT                                              AUT08940
      IF(DOPMAT(5,J,IPROST).EQ.99.0) GO TO 10                           AUT08950
      IF(DOPMAT(5,J,IPROST).EQ.98.0) GO TO 11                           AUT08960
      KEY=DOPMAT(5,J,IPROST)                                            AUT08970
      OUTLIN(J)=OUTPUT(KEY)                                             AUT08980
      GO TO 12                                                          AUT08990
   10 OUTLIN(J)=LINEAR                                                  AUT09000
      GO TO 12                                                          AUT09010
   11 OUTLIN(J)=LOG                                                     AUT09020
   12 CONTINUE                                                          AUT09030
      WRITE(6,6) (OUTLIN(J),J=ISTART,IQUIT)                             AUT09040
    6 FORMAT(4X,'PRO TYPE',2X,13(1X,A8))                                AUT09050
   27 WRITE(6,1)                                                        AUT09060
      IF(ISUMRY.EQ.1) GO TO 43                                          AUT09070
      WRITE(6,7)                                                        AUT09080
    7 FORMAT(4X,'SOURCE:')                                              AUT09090
      WRITE(6,8) (LINE5(I),I=1,31)                                      AUT09100
    8 FORMAT(8X,32A4)                                                   AUT09110
      WRITE(6,8) (LINE6(I),I=1,31)                                      AUT09120
      RETURN                                                            AUT09130
   43 ISUMRY=0                                                          AUT09140
      RETURN                                                            AUT09150
      END                                                               AUT09160
```


SUBROUTINE ISTAT
COMMON
Write(6,1)
132(-)
IEND ≠ 14 ?
F
T
ISTART ≠ 14 ?
F
T
DO J=1,5
Write(6,25)
TO 4
Write(6,2) (DOPMAT(1,J,IPROST), J=ISTART,IQUIT)
Write(6,3) (DOPMAT(2,J,IPROST), J=ISTART,IQUIT)
Write(6,4) (DOPMAT(3,J,IPROST), J=ISTART,IQUIT)
Write(6,5) (DOPMAT(4,J,IPROST), J=ISTART,IQUIT)
TO 2


2
DO J = ISTART,IQUIT
DOPMAT(5,J,IPROST) = 99.0 ?
T
10
OUTLIN(J)=LINEAR
F
DOPMAT(5,J,IPROST) = 98.0 ?
T
11
OUTLIN(J)=LOG
F
KEY=DOPMAT(5,J,IPROST)
OUTLIN(J)=OUTPUT(KEY)
Write(6,6) (OUTLIN(J),J= ISTART,IQUIT)
4
Write(6,1)
ISUMRY = 1 ?
F
T
Write(6,7) "SOURCE"
ISUMRY = 0
Write(6,8) LINE5(I),I=1,31)
Write(6,8) (LINE6(I),I=1,31)
RETURN
RETURN

XII. Subroutine Idata

This subroutine prints the base and projected data for each projection set. The first half of Idata prints the base data, which is the same for each projection set. The second half of the subroutine prints the projections for each projection set, with the projection set number being given by the value of IPROST.

Count is kept of each line of data printed and each space made so that the statistical block won't be printed on a page unless all 10 lines of the block can be printed on the same page. ICOUNT keeps track of the number of lines printed on a page. It is reset to zero each time this subroutine is entered.

In order to start printing with the correct next line of data on the next page, the number of the row of the data matrix that printing resumes with must be recorded before exiting this subroutine. This is done via NFLINE and KFLINE. NFLINE is set if the subroutine was printing the base data when exited. If the subroutine was printing the projected data set, KFLINE is set with the starting line and ISKIP is set to 1 so the part that prints the base data will be skipped over the next time Idata is called.

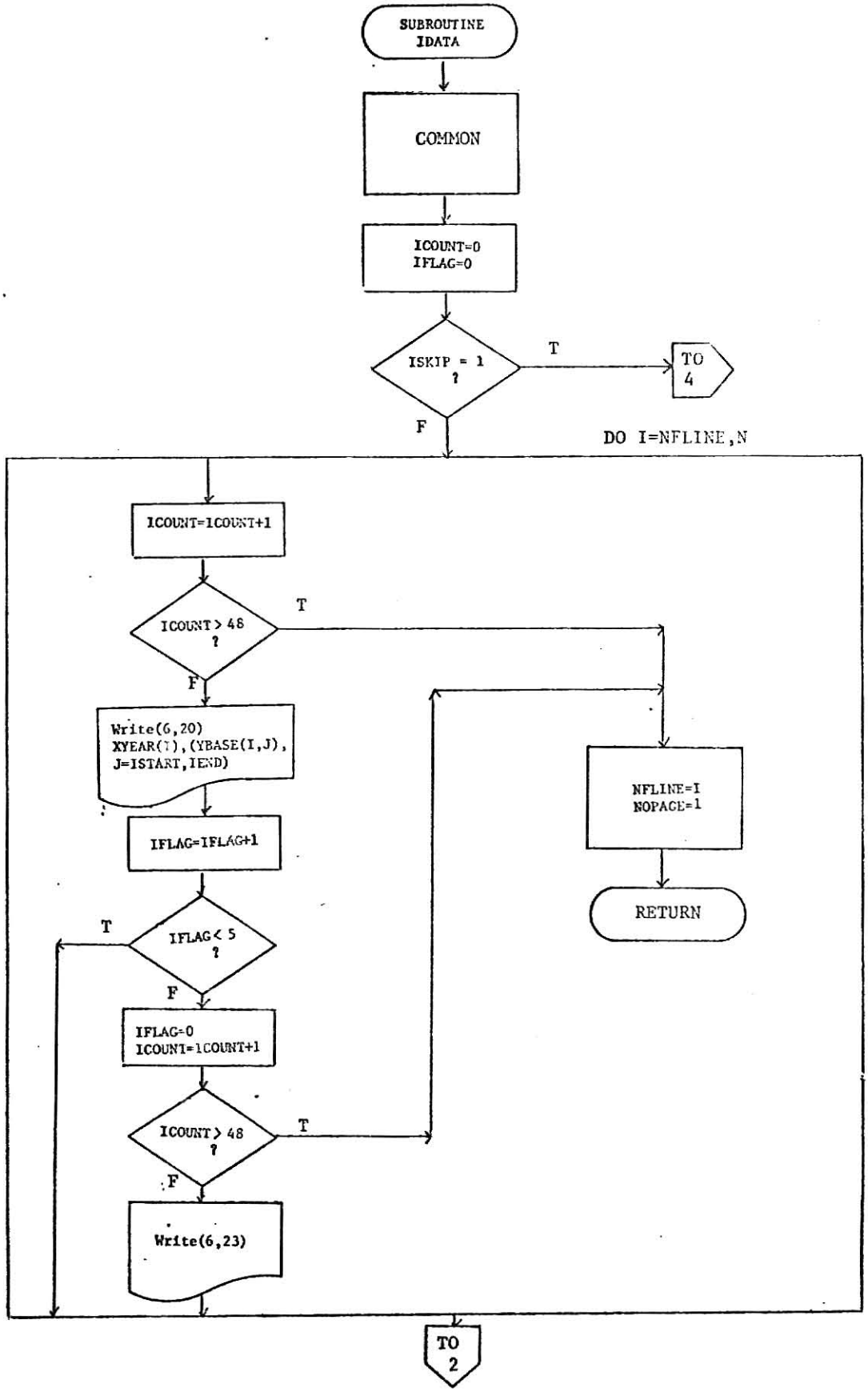
IFLAG is used so that every five years of data output a space is left. Two spaces are put between the base and projected data sets.

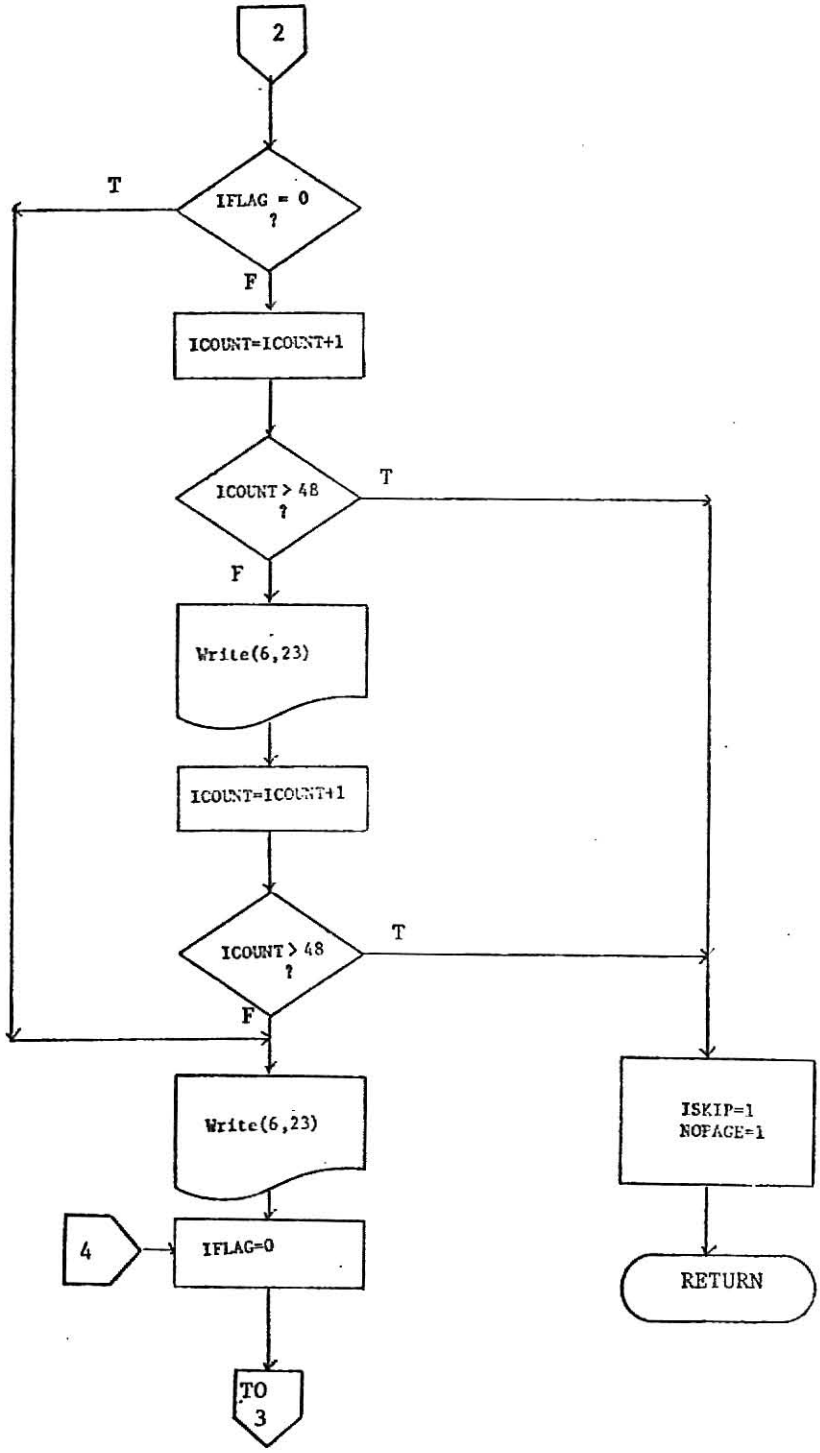
A flowchart of subroutine Idata follows a listing of it.

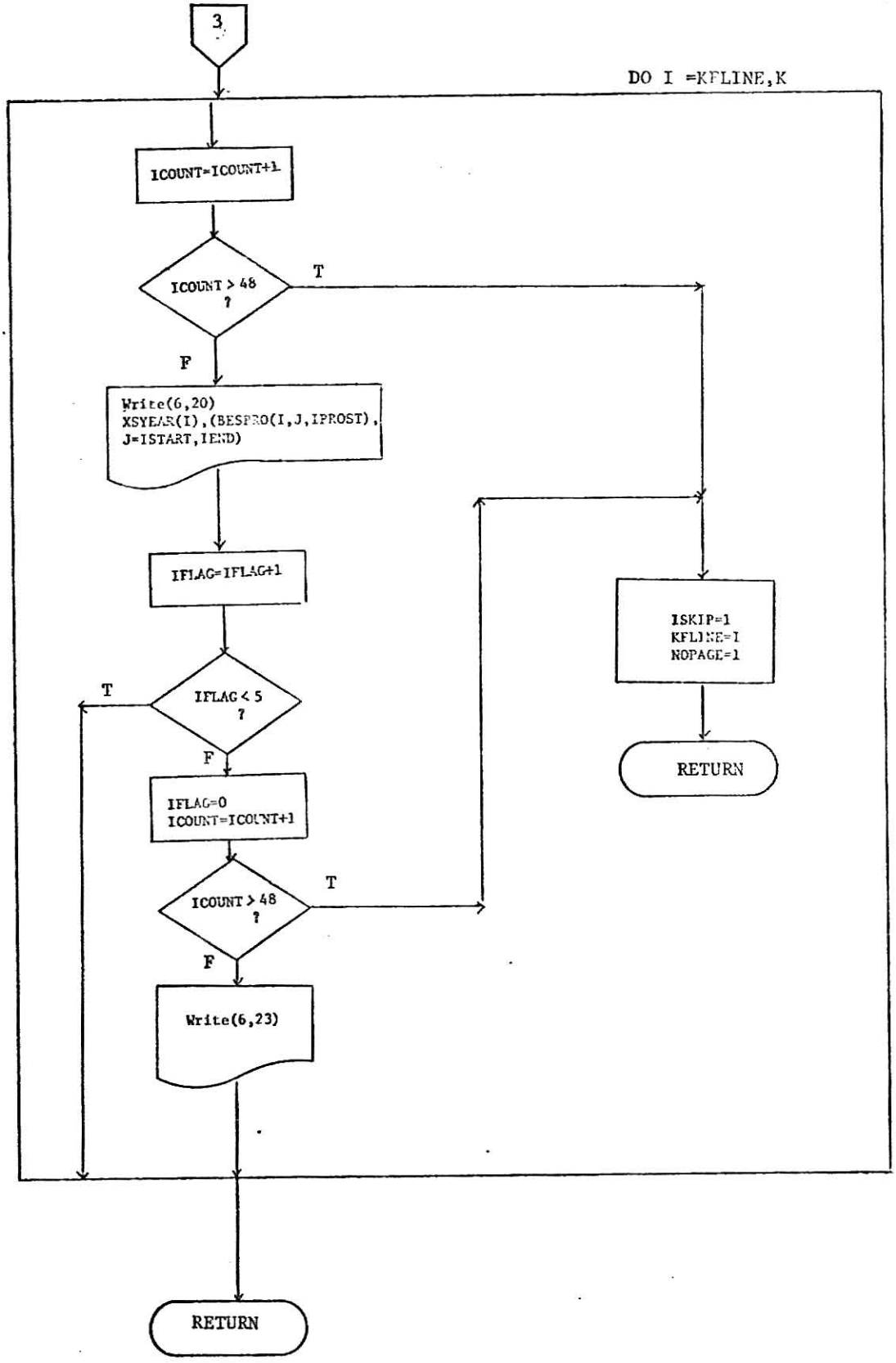
```
      SUBROUTINE IDATA                                                  AUT09170
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT09180
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR             AUT09190
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT09200
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT09210
     *ON,RSQ,YSPROJ(60),ICCL2,IREGTP,K,N                                AUT09220
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)          AUT09230
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                AUT09240
      COMMON/C/IDISTR,IPRTCT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT09250
     1E5(31),LINE6(31),ICUIT                                            AUT09260
      COMMON/G/ISTART,IEND,NFLINE,KFLINE,IPROST,ISKIP                   AUT09270
      COMMON/$PAGE/NOPAGE                                               AUT09280
      INTEGER XYEAR,XSYEAR                                              AUT09290
C---------------------------------------                                AUT09300
C   WRITE BASE DATA WITH DISTRICT TOTALS|                               AUT09310
C---------------------------------------                                AUT09320
      ICOUNT=0                                                          AUT09330
      IFLAG=0                                                           AUT09340
      IF(ISKIP.EQ.1) GO TO 29                                           AUT09350
      DO 21 I=NFLINE,N                                                  AUT09360
      ICOUNT=ICOUNT+1                                                   AUT09370
      IF(ICOUNT.GT.48) GO TO 26                                         AUT09380
      WRITE(6,20) XYEAR(I),(YBASE(I,J),J=ISTART,IEND)                   AUT09390
   20 FORMAT(4X,I4,5X,13(1X,F8.0))                                      AUT09400
      IFLAG=IFLAG+1                                                     AUT09410
      IF(IFLAG.LT.5) GO TO 21                                           AUT09420
      IFLAG=0                                                           AUT09430
      ICOUNT=ICOUNT+1                                                   AUT09440
      IF(ICOUNT.GT.48) GO TO 26                                         AUT09450
      WRITE(6,23)                                                       AUT09460
   23 FORMAT(1X)                                                        AUT09470
   21 CONTINUE                                                          AUT09480
C-------------------------------------------------------------------------AUT09490
C   IF IFLAG=0 IT MEANS A SPACE WAS JUST PUT IN SO ONLY PUT ONE MORE    |AUT09500
C   SPACE BETWEEN THE BASE AND PROJECTED DATA /CTHERWISE PUT TWO SPACES|AUT09510
C-------------------------------------------------------------------------AUT09520
      IF(IFLAG.EQ.0) GO TO 24                                           AUT09530
      ICOUNT=ICOUNT+1                                                   AUT09540
      IF(ICOUNT.GT.48) GO TO 27                                         AUT09550
      WRITE(6,23)                                                       AUT09560
      ICOUNT=ICOUNT+1                                                   AUT09570
      IF(ICOUNT.GT.48) GO TO 27                                         AUT09580
   24 WRITE(6,23)                                                       AUT09590
C-------------------------------------------                            AUT09600
C   WRITE PROJECTED DATA WITH DISTRICT TOTALS|                          AUT09610
C-------------------------------------------                            AUT09620
   29 IFLAG=0                                                           AUT09630
      DO 25 I=KFLINE,K                                                  AUT09640
      ICOUNT=ICOUNT+1                                                   AUT09650
      IF(ICOUNT.GT.48) GO TO 28                                         AUT09660
      WRITE(6,20) XSYEAR(I),(BESPRO(I,J,IPROST),J=ISTART,IEND)          AUT09670
      IFLAG=IFLAG+1                                                     AUT09680
      IF(IFLAG.LT.5) GO TO 25                                           AUT09690
      IFLAG=0                                                           AUT09700
      ICOUNT=ICOUNT+1                                                   AUT09710
      IF(ICOUNT.GT.48) GO TO 28                                         AUT09720
      WRITE(6,23)                                                       AUT09730
   25 CONTINUE                                                          AUT09740
      RETURN                                                            AUT09750
   26 NFLINE=I                                                          AUT09760
```

```
      NOPAGE=1                                                    AUT09770
      RETURN                                                      AUT09780
   27 ISKIP=1                                                     AUT09790
      NOPAGE=1                                                    AUT09800
      RETURN                                                      AUT09810
   28 ISKIP=1                                                     AUT09820
      NOPAGE=1                                                    AUT09830
      KFLINE=1                                                    AUT09840
      RETURN                                                      AUT09850
      END                                                         AUT09860
```

Subroutine Idata Flowchart




2
IFLAG = 0 ?
T
F
ICOUNT=ICOUNT+1
ICOUNT > 48 ?
T
F
Write(6,23)
ICOUNT=ICOUNT+1
ICOUNT > 48 ?
T
F
Write(6,23)
ISKIP=1
NOPAGE=1
4
IFLAG=0
RETURN
TO 3


3
DO I =KFLINE,K
ICOUNT=ICOUNT+1
ICOUNT > 48 ?
T
F
Write(6,20)
XSYEAR(I),(BESPRO(I,J,IPROST),
J=ISTART,IEND)
IFLAG=IFLAG+1
IFLAG < 5 ?
T
F
IFLAG=0
ICOUNT=ICOUNT+1
ICOUNT > 48 ?
T
F
Write(6,23)
ISKIP=1
KFLINE=1
NOPAGE=1
RETURN
RETURN

XIII. Subroutine Prosto

This subroutine stores the projections and holds them until time to print the best three. The first three projections and their associated statistics (RSQ, XBAR etc.) are automatically stored in order by RSQ with the projection with the highest RSQ going in the 1st column of matrix BESPRO and so on. (The statistics going into matrix DOPMAT.) Since the output matrix and the statistics matrix are initialized to zero before each data set is input, if less than three projections are run (due to the goodness of fit of the first one or two projections) then zeros will be printed for each projection (up to two) not run, when the best three projections are printed.

After the first three projections are stored, if subroutine EXPFLW requests more projections, then as each projection is made, its r^2 is tested against the r^2 of the three projections already stored in BESPRO plus the value of IGAP.

IGAP normally set to zero, can be used to spread out the projections. For example, if the r^2 of the third projection is .50 and IGAP equals 5, then fourth projections r^2 will have to exceed .55 in order to replace the third projection. Using IGAP this way would prevent selection of projections of, for example, .9981, .9984, .9985 unless the first three projections made were the best and they were close together as IGAP doesn't affect the storage of the first three projections.

If the RSQ of the latest (YPROJ) is less than the RSQ of the projection already stored in column 3 of BESPRO (the third best projection, as measured by r^2, so far made) plus IGAP, then nothing happens and the subroutine returns. If the RSQ of the latest projection is better than the r^2 of the projection in the third column of BESPRO plus IGAP, the last projection's RSQ is tested against the 2nd columns associated RSQ plus IGAP. If found

better, against the 1st column's projections r^2 plus IGAP.

From the above, the subroutine decides if the r^2 of the latest projection is between the 1st and 2nd or 2nd and 3rd best projections' r^2's. If it fits between the 1st and 2nd projection, then the 2nd column of BESPRO and DOPMAT is put into the 3rd column displacing what formerly was the 3rd best projection. Then the latest projection is stored in the 2nd column of BESPRO and its associated statistics are put into the second column of DOPMAT.

If the newest projections RSQ is better than the formerly best projection's RSQ (in the 1st column of BESPRO) then the newest projection is put there with the existing 1st and 2nd best projection being shuffled one column to the right--becoming the 2nd and 3rd best projections and displacing what was formerly the third best projection.

If the newest projection is only a better projection than the 3rd best projection, the newest projection replaces it directly and no shuffling of the other existing projections occurs. In this case and the others above, when data in a column of BESPRO is shifted, the data in the same column number in DOPMAT is also shifted by an equal amount.

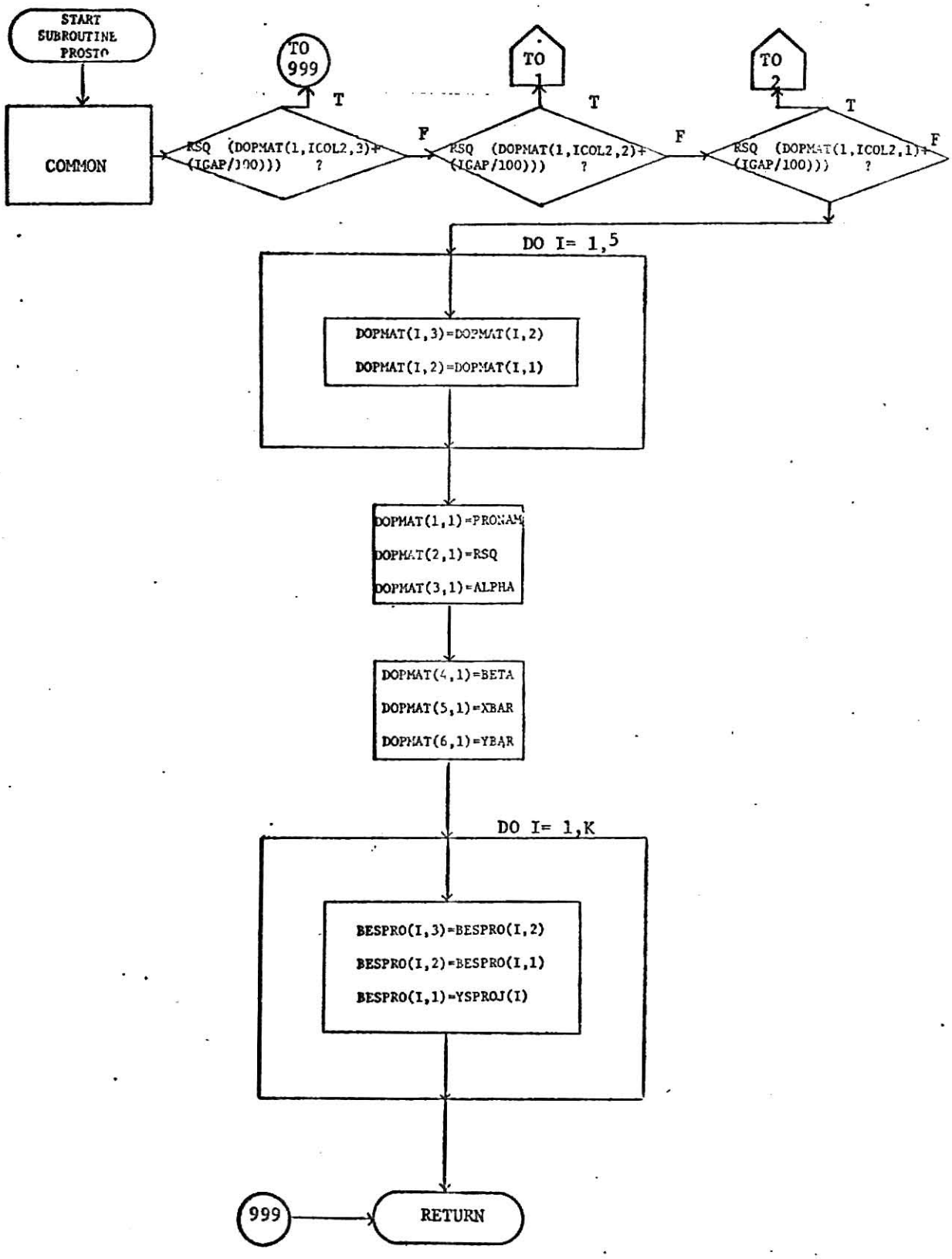
The above processes occur each time a projection is made. Thus, after the last projection is made, BESPRO contains the best three projections from among all the projections made for a given data set.

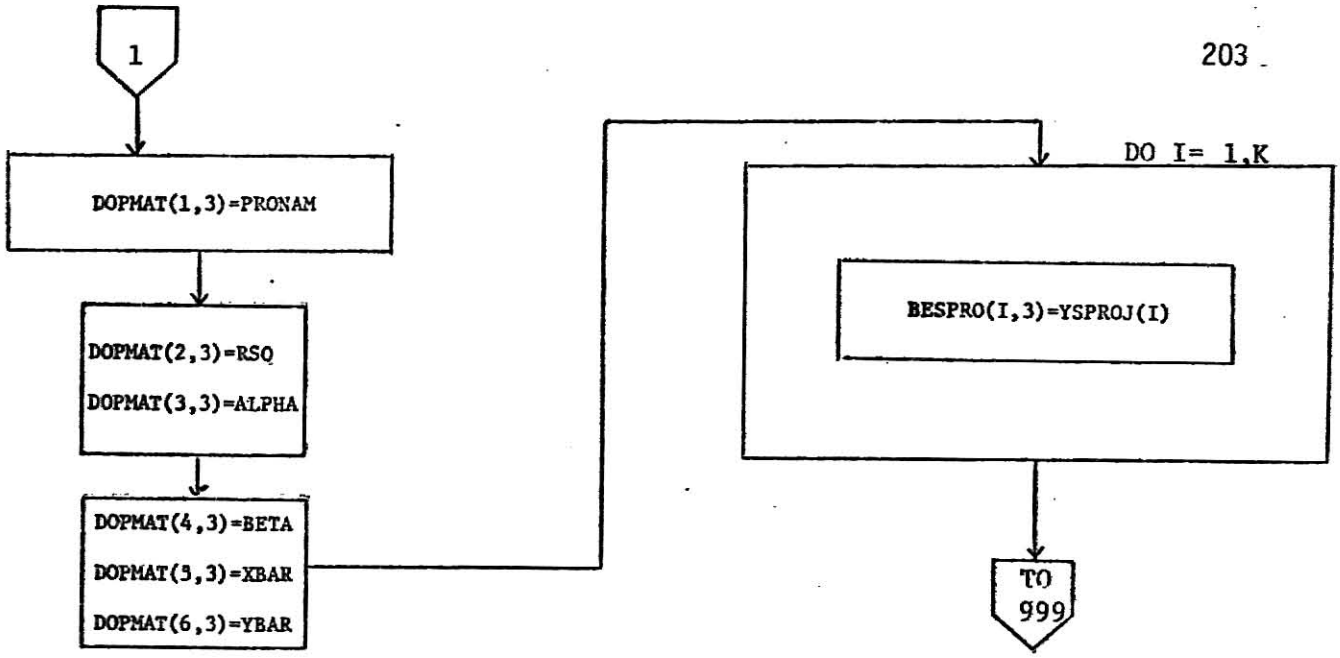
A flowchart of subroutine Prosto follows a listing of it.

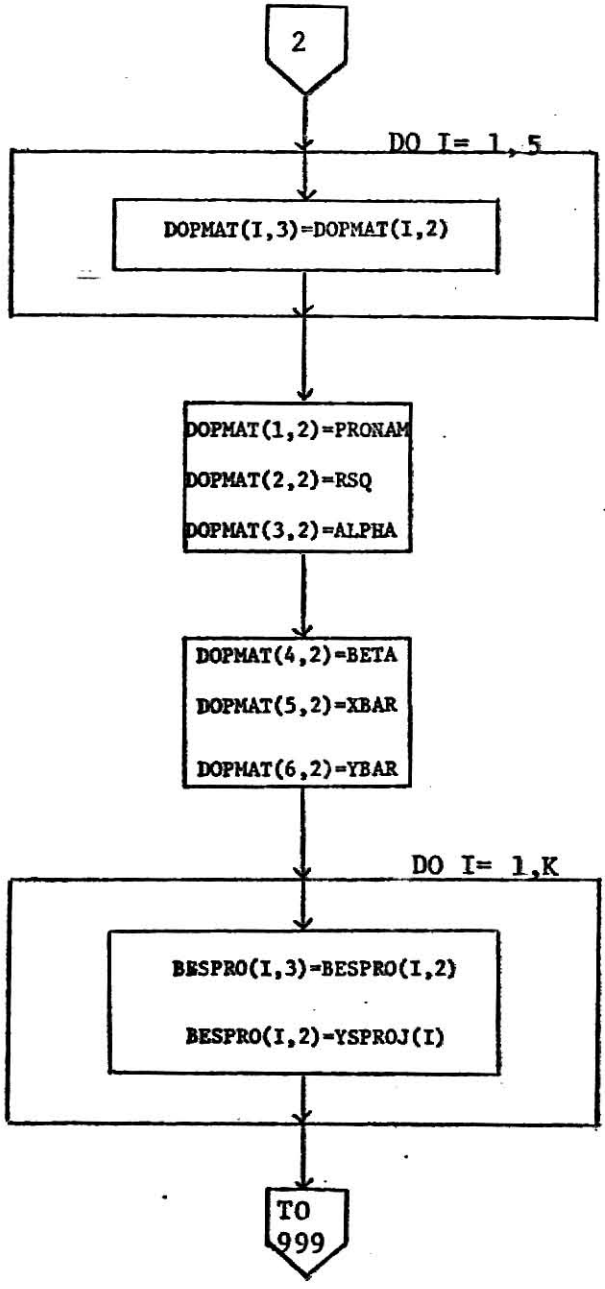
```
      SUBROUTINE PROSTC                                                 AUT09870
C-----------------------------------------------------------------------AUT09880
C   THIS SUBROUTINE KEEPS THE BEST THREE PROJECTIONS AND THEIR          |AUT09890
C   ASSOCIATED DOPE VECTORS (CONTAINING RSQ, ALPHA, BETA, YBAR,         |AUT09900
C   AND PRONAM.)  THEY ARE STORED IN MATRIX DOPMAT (FOR THE DOPE VEC-   |AUT09910
C   TORS) AND IN MATRIX BESPRO FOR THE BEST THREE PROJECTIONS.  THE     |AUT09920
C   PROJECTIONS ARE FILED IN THE MATRIX IN ORDER OF THEIR RSQ, WITH     |AUT09930
C   THE PROJECTION WITH THE HIGHEST RSQ GOING INTO THE FIRST PLANE      |AUT09940
C   OF THE MATRIX AND SO ON.  NEW PROJECTIONS ARE COMPARED TO THOSE     |AUT09950
C   ON FILE IN THIS SUBROUTINE AND IF THEIR RSQ EXCEEDS THAT OF THE     |AUT09960
C   LOWEST RSQ-PROJECTION ON FILE, THE NEW PROJECTION IS STORED IN      |AUT09970
C   THE PLANE  BEFITTING ITS RSQ RANK AND ONE OF THE OTHER PROJEC-      |AUT09980
C   TIONS IS DROPPED AND THE OTHER PROJECTIONS AND THEIR ASSOCIATED     |AUT09990
C   DOPE VECTORS ARE MOVED OVER TO MAKE ROOM FOR THE NEW PROJECTION     |AUT10000
C   AND ITS DOPE VECTOR, AS DICTATED BY ITS RSQ RANK, SHOULD GO.        |AUT10010
C   **THUS THE PRINTING OF BESPRO WOULD GIVE THE BEST THREE PROJEC-     |AUT10020
C   TIONS FOR EACH COUNTY (COLUMN OF BESPRO).                           |AUT10030
C                                                                       |AUT10040
C   PRONAM = NAME OF THE TYPE OF PROJECTION JUST MADE BY REGRES.        |AUT10050
C            PRONAM IS REPRESENTED BY A NUMBER, THE KEY TO WHICH IS:    |AUT10060
C                                                                       |AUT10070
C   -------KEY------------------PRONAM------------------KEY--------     |AUT10080
C                                                                       |AUT10090
C            99 = LINEAR                                                |AUT10100
C            98 = LOGARITHMIC                                           |AUT10110
C            OTHERWISE:  10 * EXPONENT    (EX: 1.4->PRONAM=14           |AUT10120
C-----------------------------------------------------------------------AUT10130
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT10140
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQX,XBAR                    AUT10150
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT10160
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT10170
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                AUT10180
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)          AUT10190
C-----------------------------------------------------------------     AUT10200
C   IGAP IS THE MINIMUM REQUIRED RSQ DIFFERENCE BETWEEN RSQ'S.|         AUT10210
C        THE VALUE IS DETERMINED IN THE MAIN BY READING.      |         AUT10220
C-----------------------------------------------------------------     AUT10230
      IF(RSQ.LE.(DOPMAT(1,ICOL2,3)+(IGAP/100))) GO TO 999               AUT10240
      IF(RSQ.LE.(DOPMAT(1,ICOL2,2)+(IGAP/100))) GO TO 1                 AUT10250
      IF(RSQ.LE.(DOPMAT(1,ICOL2,1)+(IGAP/100))) GO TO 2                 AUT10260
C-------------------------------------------                            AUT10270
C   PUT NEW PROJECTION IN 1ST PROJECTION SET.|                          AUT10280
C-------------------------------------------                            AUT10290
      DO 3 I=1,5                                                        AUT10300
      DOPMAT(I,ICOL2,3)=DOPMAT(I,ICOL2,2)                               AUT10310
    3 DOPMAT(I,ICOL2,2)=DOPMAT(I,ICOL2,1)                               AUT10320
      DOPMAT(1,ICOL2,1)=RSQ                                             AUT10330
      DOPMAT(2,ICOL2,1)=ALPHA                                           AUT10340
      DOPMAT(3,ICOL2,1)=BETA                                            AUT10350
      DOPMAT(4,ICOL2,1)=YBAR                                            AUT10360
      DOPMAT(5,ICOL2,1)=PRONAM                                          AUT10370
      DO 4 I=1,K                                                        AUT10380
      BESPRO(I,ICOL2,3)=BESPRO(I,ICOL2,2)                               AUT10390
      BESPRO(I,ICOL2,2)=BESPRO(I,ICOL2,1)                               AUT10400
    4 BESPRO(I,ICOL2,1)=YSPROJ(I)                                       AUT10410
      GO TO 999                                                         AUT10420
C-------------------------------------------                            AUT10430
C   PUT NEW PROJECTION IN 2ND PROJECTION SET.|                          AUT10440
C-------------------------------------------                            AUT10450
    2 DO 5 I=1,5                                                        AUT10460
```

```
    5 DOPMAT(I,ICOL2,3)=DOPMAT(I,ICOL2,2)                          AUT10470
      DOPMAT(1,ICOL2,2)=RSQ                                        AUT10480
      DOPMAT(2,ICOL2,2)=ALPHA                                      AUT10490
      DOPMAT(3,ICOL2,2)=BETA                                       AUT10500
      DOPMAT(4,ICOL2,2)=YBAR                                       AUT10510
      DOPMAT(5,ICOL2,2)=PRCNAM                                     AUT10520
      DO 6 I=1,K                                                   AUT10530
      BESPRO(I,ICOL2,3)=BESPRO(I,ICOL2,2)                          AUT10540
    6 BESPRC(I,ICOL2,2)=YSPRCJ(I)                                  AUT10550
      GO TO 999                                                    AUT10560
C-------------------------------------------                       AUT10570
C   PUT NEW PROJECTION IN 3RD PROJECTION SET.|                     AUT10580
C-------------------------------------------                       AUT10590
    1 DOPMAT(1,ICOL2,3)=RSQ                                        AUT10600
      DOPMAT(2,ICOL2,3)=ALPHA                                      AUT10610
      DOPMAT(3,ICOL2,3)=BETA                                       AUT10620
      DOPMAT(4,ICOL2,3)=YBAR                                       AUT10630
      DOPMAT(5,ICOL2,3)=PRCNAM                                     AUT10640
      DO 7 I=1,K                                                   AUT10650
    7 BESPRC(I,ICOL2,3)=YSPRCJ(I)                                  AUT10660
  999 RETURN                                                       AUT10670
      END                                                          AUT10680
```

Flowchart for Subroutine Prosto




1
DOPMAT(1,3)=PRONAM
DOPMAT(2,3)=RSQ
DOPMAT(3,3)=ALPHA
DOPMAT(4,3)=BETA
DOPMAT(5,3)=XBAR
DOPMAT(6,3)=YBAR
DO I= 1,K
BESPRO(I,3)=YSPROJ(I)
TO 999


2
DO I= 1, 5
DOPMAT(I,3)=DOPMAT(I,2)
DOPMAT(1,2)=PRONAM
DOPMAT(2,2)=RSQ
DOPMAT(3,2)=ALPHA
DOPMAT(4,2)=BETA
DOPMAT(5,2)=XBAR
DOPMAT(6,2)=YBAR
DO I= 1,K
BESPRO(I,3)=BESPRO(I,2)
BESPRO(I,2)=YSPROJ(I)
TO 999

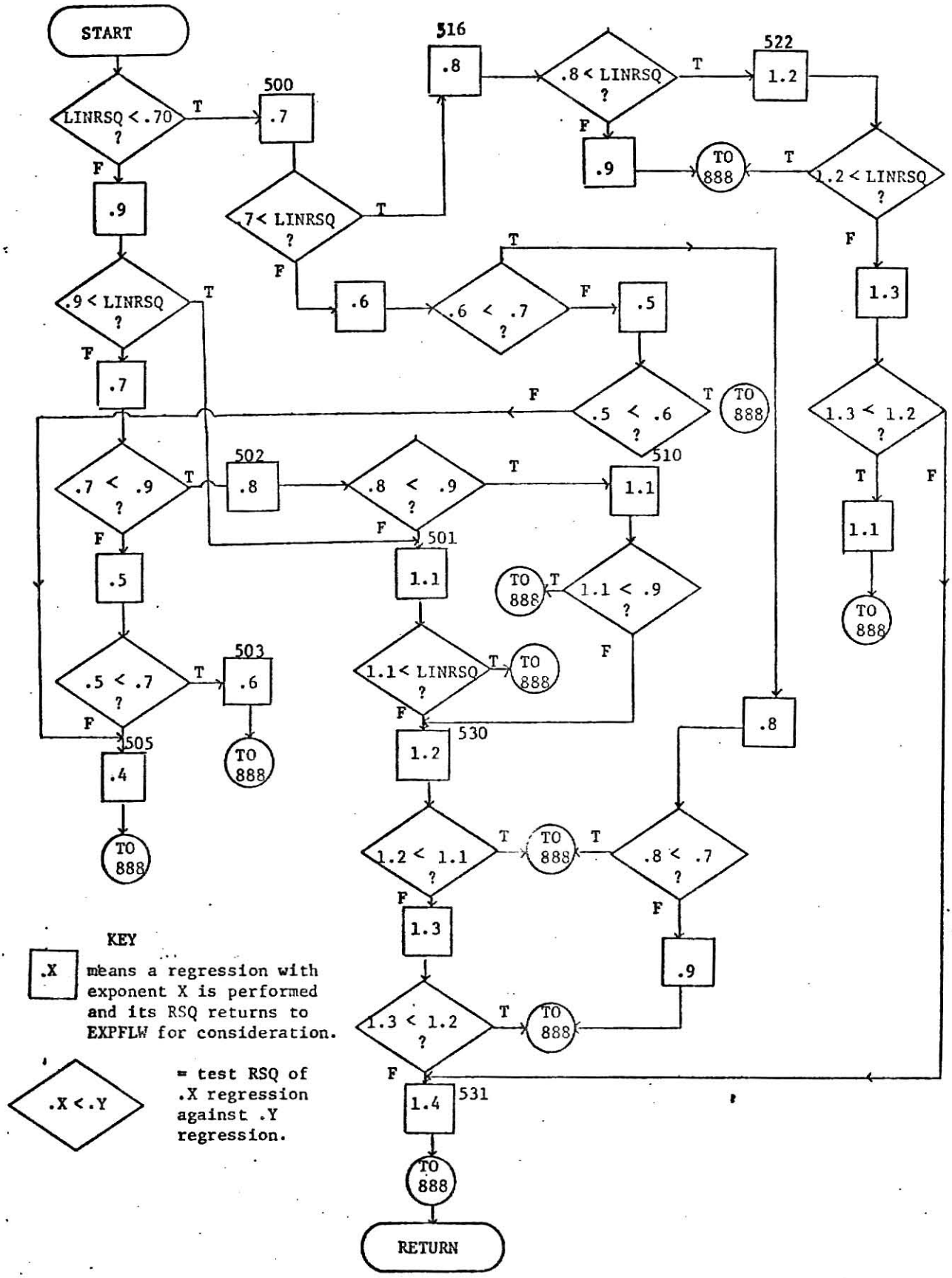
XIV. Subroutine Expflw

Expflw is a subroutine that determines the order that different exponential projections with varying exponents are to be performed. Expflw then controls the regression by establishing the exponent and then calling subroutine REGRES to which the exponent is passed.

The order the regressions of various exponents are to be performed in is determined by examining the RSQ of the linear projection or else the previously performed exponential regression's RSQ (such as ERSQ12, etc.). Expflw then uses a built in decision tree to decide the next regression. The first decision is whether the linear projection was a good fit. If so, then Expflw tries exponents that are close to 1.0 first in order to achieve a higher RSQ. If a linear projection was a bad fit, it is assumed that the trend best fitting the base data may be curvilinear so the exponents tried first are further from 1.0.

Once Expflw has determined its starting point then all the exponents between 0.4 and 1.4 are tried in regressions either implicitly or explicitly. For example, if the linear r^2 (LINRSQ) was .78 and the r^2 of a 0.7 exponent regression is less than .78 then all the exponents below 0.7 can also be eliminated implicitly. Expflw would then try the exponents between 0.7 and 1.0 if they were better than 1.0 (in terms of RSQ) it would stop, otherwise the exponents above 1.0 would be tired. The overall decision tree used by Expflw is shown on the following page in a simplified flowchart. A longer flowchart follows a listing of the subroutine.

SIMPLE FLOWCHART FOR EXPFLW

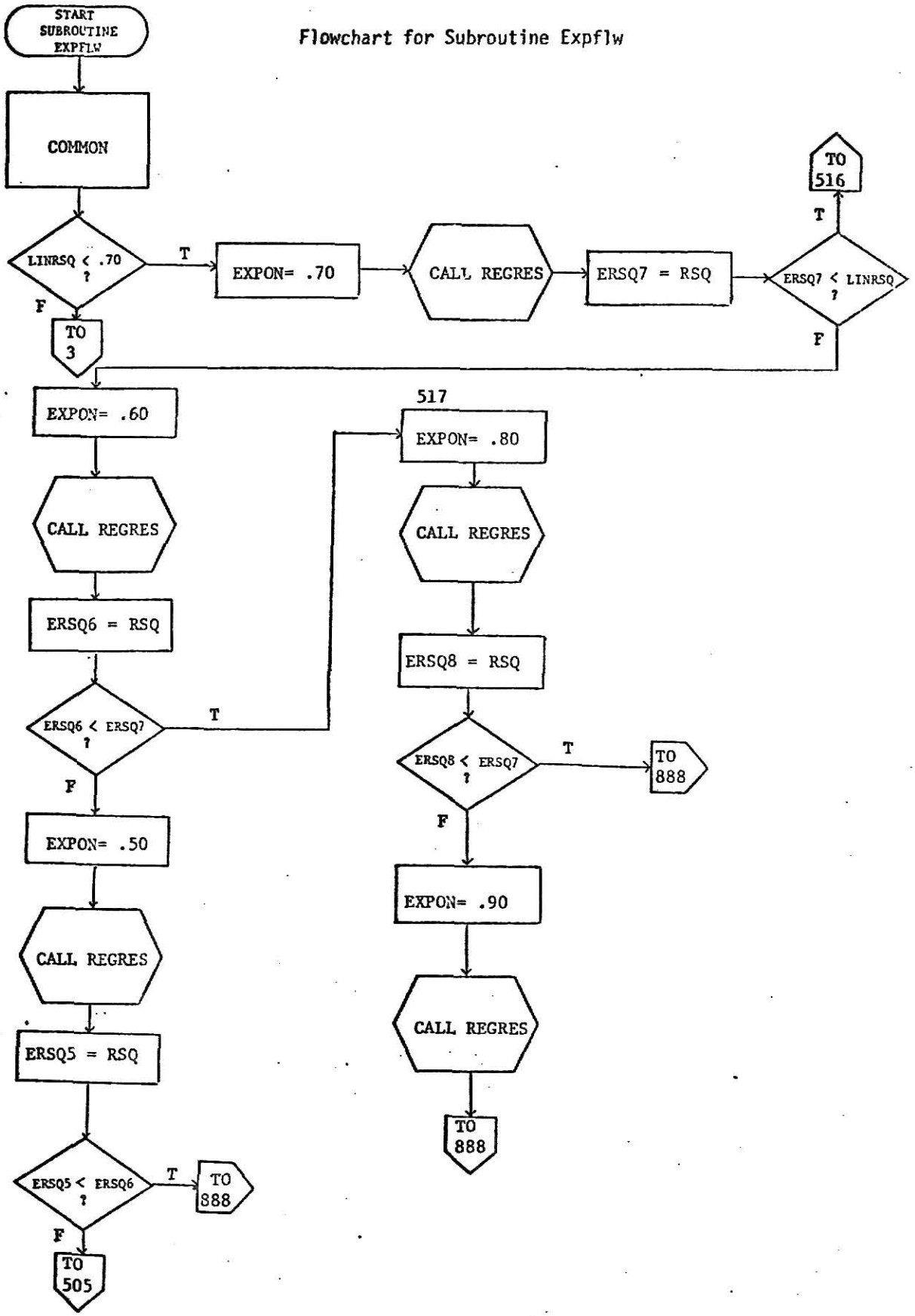


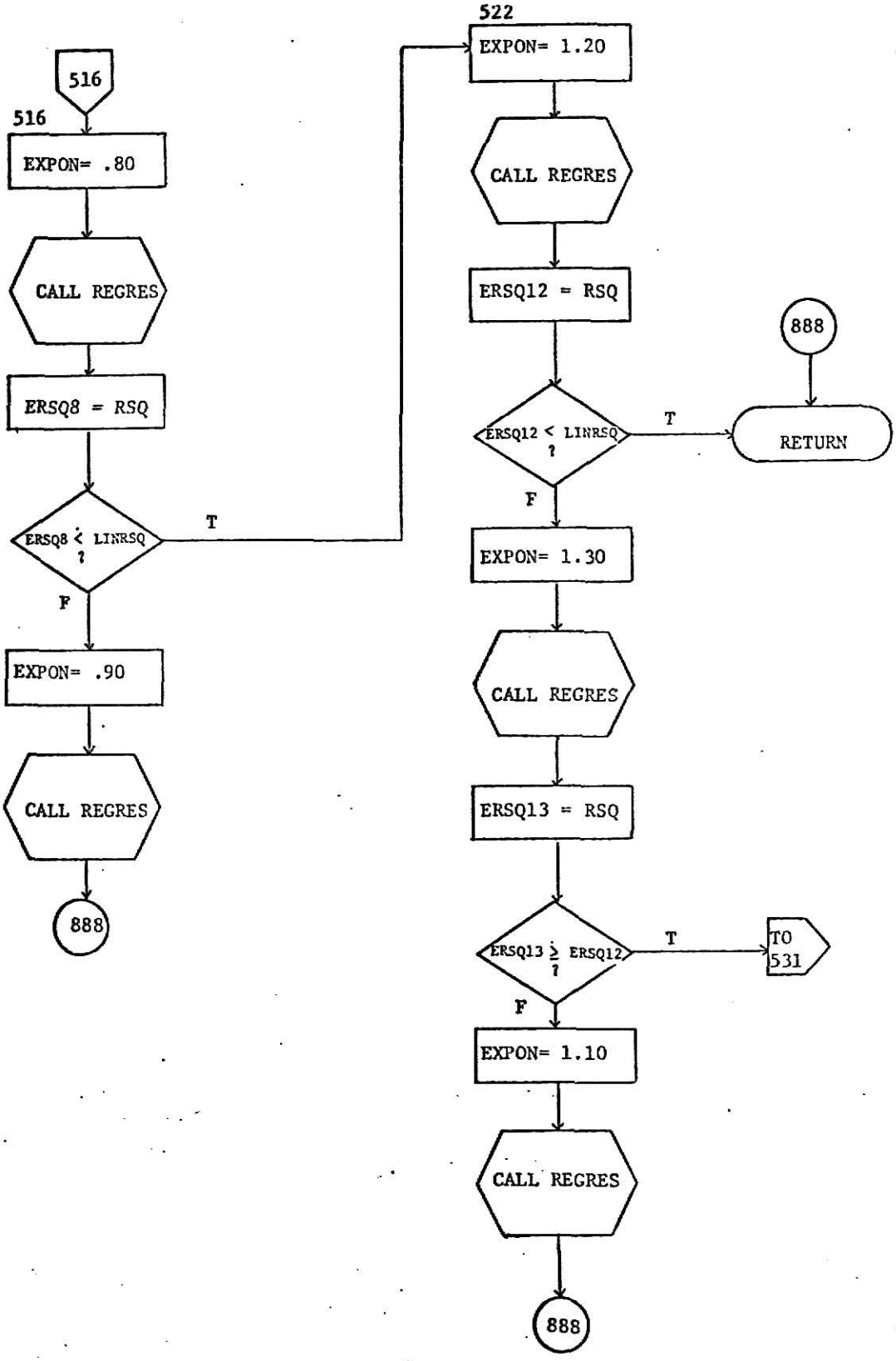
```
C-------------------------------------------------------------------------AUT10690
C   THIS SUBROUTINE CONTAINS THE LOGIC FOR PICKING THE BEST EXPONENTIAL |AUT10700
C   PROJECTION.  THIS IS DONE BY PICKING THE NEXT PROJECTION TO TRY     |AUT10710
C   BASED ON THE RSQ OF THE BEST PROJECTION MADE TO DATE.  THIS SUB-    |AUT10720
C   ROUTINE ASSUMES THAT IF THE LINEAR RSQ (LINRSQ) WAS NOT ABOVE 0.7   |AUT10730
C   THAT THAT THE TREND WOULD COME NEARER TO AN EXPONENTIAL PROJECTION  |AUT10740
C   THAT WAS FURTHER AWAY FROM 1.0 AND THUS THE SUBROUTINE LOOKS FIRST  |AUT10750
C   AT EXPONENTIAL PROJECTIONS WHERE THE EXPONENT IS FURTHER AWAY FROM  |AUT10760
C   1.0.  OTHERWISE (WHEN THE RSQ OF THE LINEAR PROJECTION IS ABOVE     |AUT10770
C   .70) THE EXPONENTIAL PROJECTIONS WITH AN EXPONENT NEAR 1.0 ARE      |AUT10780
C   TRIED FIRST IN AN EFFORT TO SAVE THE NUMBER OF PROJECTIONS NEC-     |AUT10790
C   CESSARY.                                                            |AUT10800
C-------------------------------------------------------------------------AUT10810
C                          SYMBOL DEFINITION                            |AUT10820
C                                                                       |AUT10830
C   LINRSQ = R SQUARE OF THE LINEAR REGRESSION.                         |AUT10840
C                                                                       |AUT10850
C   LOGRSQ = R SQUARE OF THE LOGARITHMIC REGRESSION.                    |AUT10860
C                                                                       |AUT10870
C   REGRES = SUBROUTINE WHICH PERFORMS THE FOLLOWING:                   |AUT10880
C            LINEAR      WHEN IREGTP = 1.0                              |AUT10890
C            LOGARITHMIC             = 2.0                              |AUT10900
C            EXPONENTIAL             = 3.0,                             |AUT10910
C            TYPES OF REGRESSION.  IREGTP IS THE KEY TO THE TYPE OF     |AUT10920
C            REGRESSION REGRES IS TO PERFORM.                           |AUT10930
C                                                                       |AUT10940
C   EXPON  = VALUE PASSED IN COMMON TO SUBROUTINE REGRES TO DETER-      |AUT10950
C            MINE THE EXPONENT USED IN EXPONENTIAL REGRESSION           |AUT10960
C                                                                       |AUT10970
C   ERSQ7  = RSQ FOR AN EXPONENTIAL PROJECTION USING THE EXPONENT       |AUT10980
C     ETC.   .7.  ERSQ12 MEANS THE EXPONENT 1.2.                        |AUT10990
C-------------------------------------------------------------------------AUT11000
      SUBROUTINE EXPFLW                                                 AUT11010
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR             AUT11020
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT11030
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT11040
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT11050
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                AUT11060
      COMMON/$EXP/LINRSQ                                                AUT11070
      REAL*4 LINRSQ                                                     AUT11080
      IF(LINRSQ.LT.0.70) GO TO 500                                      AUT11090
      EXPON=.90000000000                                                AUT11100
      CALL REGRES                                                       AUT11110
      ERSQ9=RSQ                                                         AUT11120
      IF(ERSQ9.LT.LINRSQ) GO TO 501                                     AUT11130
      EXPON=.700000000                                                  AUT11140
      CALL REGRES                                                       AUT11150
      ERSQ7=RSQ                                                         AUT11160
      IF(ERSQ7.LT.ERSQ9) GO TO 502                                      AUT11170
      EXPON=.500000000                                                  AUT11180
      CALL REGRES                                                       AUT11190
      ERSQ5=RSQ                                                         AUT11200
      IF(ERSQ5.LT.ERSQ7) GO TO 503                                      AUT11210
  505 EXPON=.400000000                                                  AUT11220
      CALL REGRES                                                       AUT11230
      GO TO 888                                                         AUT11240
  503 EXPON=.6000000000                                                 AUT11250
      CALL REGRES                                                       AUT11260
      GO TO 888                                                         AUT11270
  510 EXPON=1.100000000                                                 AUT11280
```

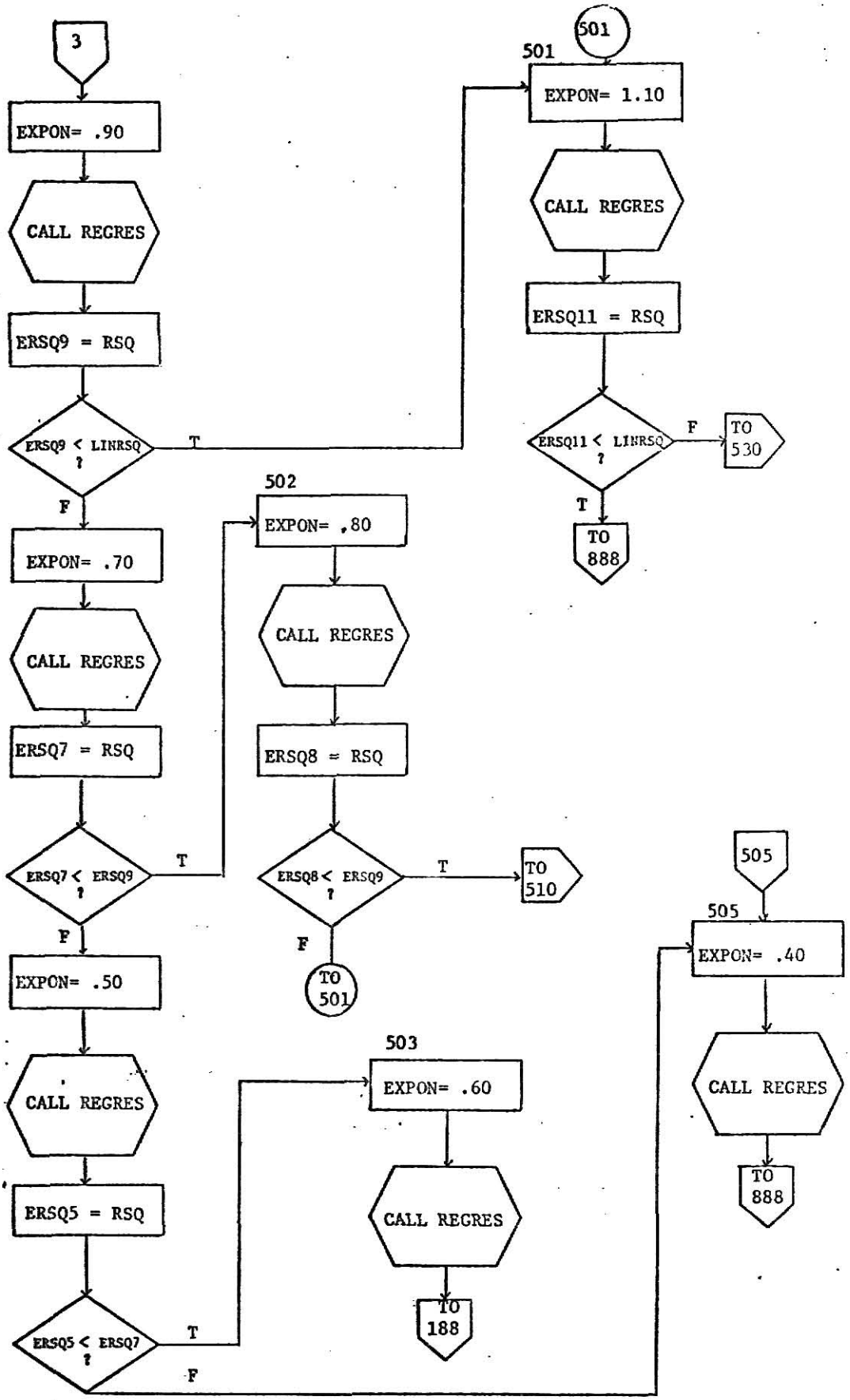
```
      CALL REGRES                                                    AUT11290
      ERSQ11=RSQ                                                     AUT11300
      IF(ERSQ11.LT.ERSQ9) GO TO 888                                  AUT11310
      GO TO 530                                                      AUT11320
  502 EXPON=.800000000                                               AUT11330
      CALL REGRES                                                    AUT11340
      ERSQ8=RSQ                                                      AUT11350
      IF(ERSQ8.LT.ERSQ9) GO TO 510                                   AUT11360
  501 EXPON=1.100000000                                              AUT11370
      CALL REGRES                                                    AUT11380
      ERSQ11=RSQ                                                     AUT11390
      IF(ERSQ11.LT.LINRSQ) GO TO 888                                 AUT11400
  530 EXPON=1.200000000                                              AUT11410
      CALL REGRES                                                    AUT11420
      ERSQ12=RSQ                                                     AUT11430
      IF(ERSQ12.LT.ERSQ11) GO TO 888                                 AUT11440
      EXPON=1.300000000                                              AUT11450
      CALL REGRES                                                    AUT11460
      ERSQ13=RSQ                                                     AUT11470
      IF(ERSQ13.LT.ERSQ12) GO TO 888                                 AUT11480
  531 EXPON=1.400000000                                              AUT11490
      CALL REGRES                                                    AUT11500
      GO TO 888                                                      AUT11510
  500 EXPON=.700000000                                               AUT11520
      CALL REGRES                                                    AUT11530
      ERSQ7=RSQ                                                      AUT11540
      IF(ERSQ7.LT.LINRSQ) GO TO 516                                  AUT11550
      EXPON=.600000000                                               AUT11560
      CALL REGRES                                                    AUT11570
      ERSQ6=RSQ                                                      AUT11580
      IF(ERSQ6.LT.ERSQ7) GO TO 517                                   AUT11590
      EXPON=.500000000                                               AUT11600
      CALL REGRES                                                    AUT11610
      ERSQ5=RSQ                                                      AUT11620
      IF(ERSQ5.LT.ERSQ6) GO TO 888                                   AUT11630
      GO TO 505                                                      AUT11640
  517 EXPON=.800000000                                               AUT11650
      CALL REGRES                                                    AUT11660
      ERSQ8=RSQ                                                      AUT11670
      IF(ERSQ8.LT.ERSQ7) GO TO 888                                   AUT11680
      EXPON=.900000000                                               AUT11690
      CALL REGRES                                                    AUT11700
      GO TO 888                                                      AUT11710
  516 EXPON=.800000000                                               AUT11720
      CALL REGRES                                                    AUT11730
      ERSQ8=RSQ                                                      AUT11740
      IF(ERSQ8.LT.LINRSQ) GO TO 522                                  AUT11750
      EXPON=.900000000                                               AUT11760
      CALL REGRES                                                    AUT11770
      GO TO 888                                                      AUT11780
  522 EXPON=1.200000000                                              AUT11790
      CALL REGRES                                                    AUT11800
      ERSQ12=RSQ                                                     AUT11810
      IF(ERSQ12.LT.LINRSQ) GO TO 888                                 AUT11820
      EXPON=1.300000000                                              AUT11830
      CALL REGRES                                                    AUT11840
      ERSQ13=RSQ                                                     AUT11850
      IF(ERSQ13.GE.ERSQ12) GO TO 531                                 AUT11860
      EXPON=1.100000000                                              AUT11870
      CALL REGRES                                                    AUT11880
```

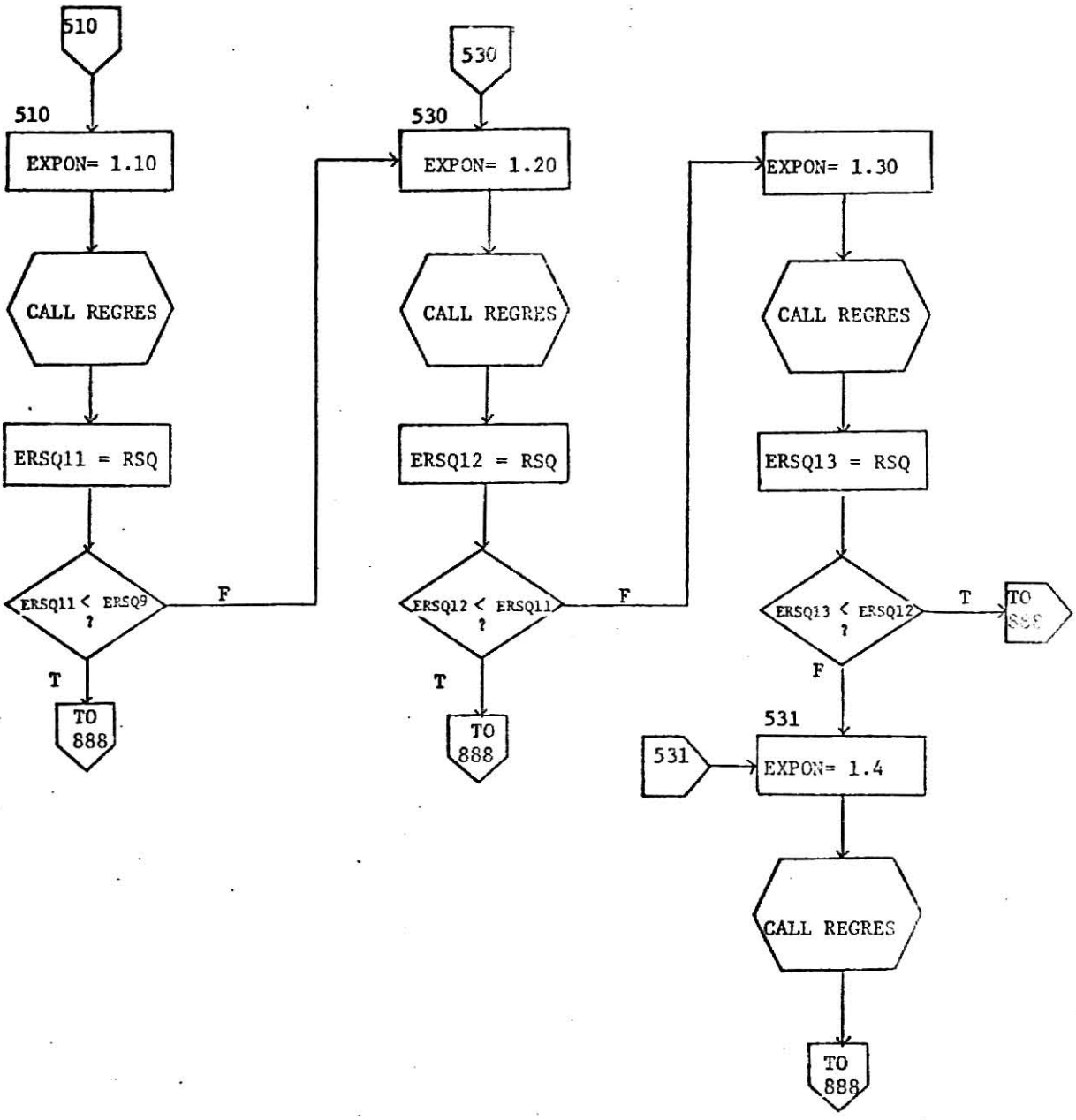
```
888 RETURN                                                    AUT11890
    END                                                       AUT11900
```

Flowchart for Subroutine Expflw




516
516
EXPON= .80
CALL REGRES
ERSQ8 = RSQ
ERSQ8 < LINRSQ ?
T
F
EXPON= .90
CALL REGRES
888
522
EXPON= 1.20
CALL REGRES
ERSQ12 = RSQ
888
ERSQ12 < LINRSQ ?
T
RETURN
F
EXPON= 1.30
CALL REGRES
ERSQ13 = RSQ
ERSQ13 ≥ ERSQ12 ?
T
TO 531
F
EXPON= 1.10
CALL REGRES
888


3
EXPON= .90
CALL REGRES
ERSQ9 = RSQ
ERSQ9 < LINRSQ ?
T
F
EXPON= .70
CALL REGRES
ERSQ7 = RSQ
ERSQ7 < ERSQ9 ?
T
F
EXPON= .50
CALL REGRES
ERSQ5 = RSQ
ERSQ5 < ERSQ7 ?
T
F
501
EXPON= 1.10
CALL REGRES
ERSQ11 = RSQ
ERSQ11 < LINRSQ ?
F
TO 530
T
TO 888
502
EXPON= .80
CALL REGRES
ERSQ8 = RSQ
ERSQ8 < ERSQ9 ?
T
TO 510
F
TO 501
503
EXPON= .60
CALL REGRES
TO 188
505
EXPON= .40
CALL REGRES
TO 888


510
510
EXPON= 1.10
CALL REGRES
ERSQ11 = RSQ
ERSQ11 < ERSQ9 ?
F
T
TO 888
530
530
EXPON= 1.20
CALL REGRES
ERSQ12 = RSQ
ERSQ12 < ERSQ11 ?
F
T
TO 888
EXPON= 1.30
CALL REGRES
ERSQ13 = RSQ
ERSQ13 < ERSQ12 ?
T
TO 888
F
531
531
EXPON= 1.4
CALL REGRES
TO 888

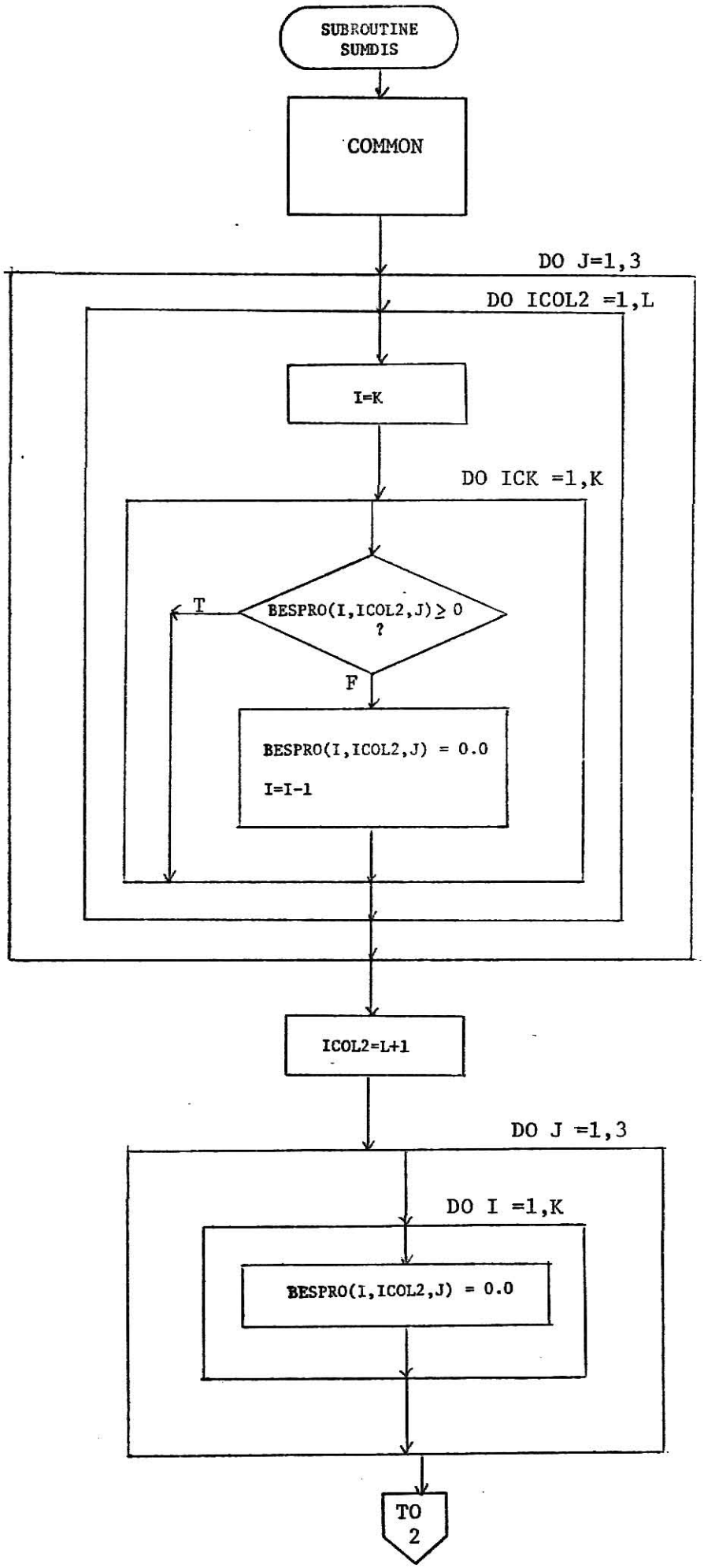
XV. Subroutine Sumdis

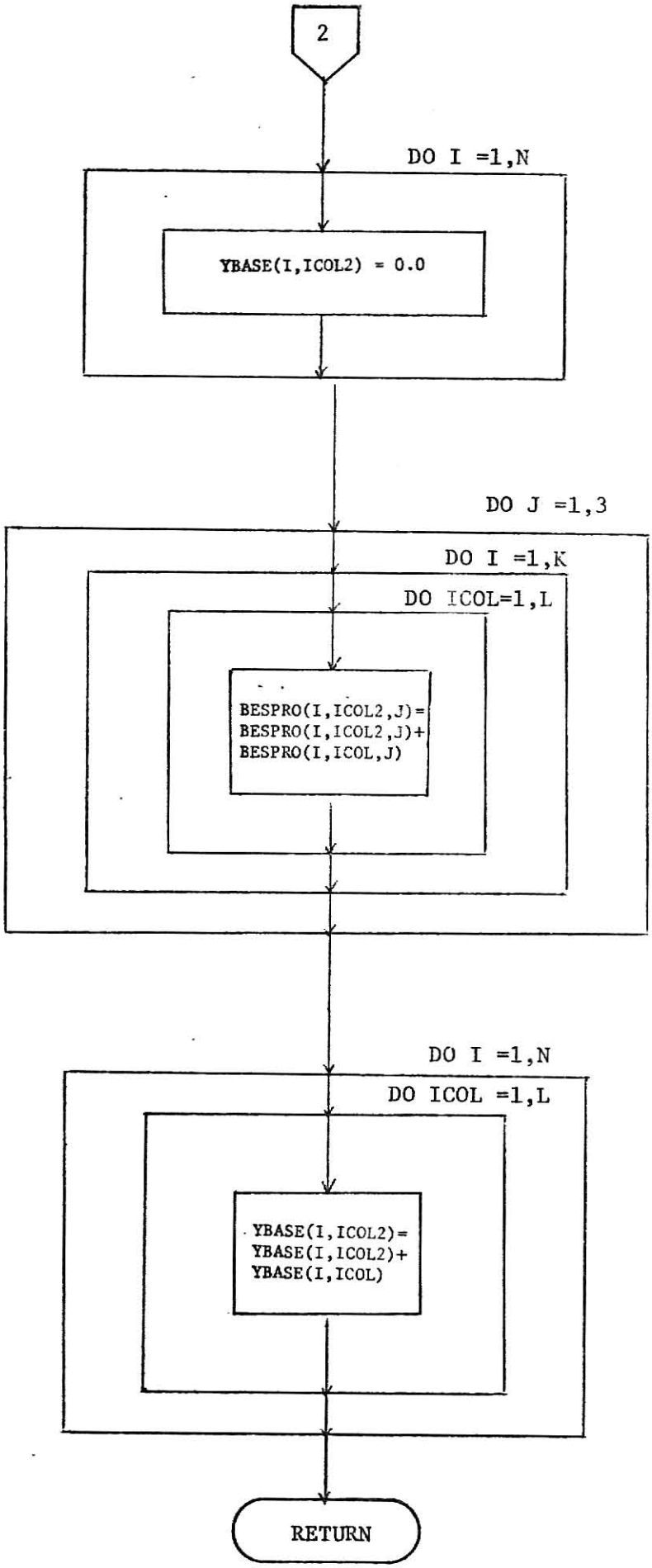
The subroutine calculates the district totals. First, however, it tests the projections to see if any counties projections went negative. Since negative production isn't possible, the negative figures are all converted to zero. As the most likely place to find negative production would be the last projected year, the program starts there and searches each projection backwards. When it encounters a positive number or zero it stops looking backwards since with a non-cyclical, non-polynomial projection no negative projected values would proceed a positive value.

Once each county's projections have been checked and corrected for negative production, the district totals are calculated. This is done by summing the first L columns of the matrices holding the base data and projected data, a row at a time. The sum is put in columns L + 1. This is done for each set of projected data.

A flowchart of subroutine Sumdis follows a listing of it.

```
      SUBROUTINE SUMDIS                                                 AUT11910
      REAL*8 YBASE,TOP,BOT,RSQ,EXPON                                    AUT11920
      REAL*8 XSPROJ(60),XSBASE,SUMX,SUMY,SUMSQY,SUMSQX,XBAR             AUT11930
      REAL*8 YBAR,ALPHA,BETA,SUMXY,XBASE                                AUT11940
      COMMON/$PROJ/XBASE(40),XSBASE(40),ALPHA,BETA,YBAR,YBASE(40,26),EXPAUT11950
     *ON,RSQ,YSPROJ(60),ICOL2,IREGTP,K,N                                AUT11960
      COMMON/REGR/XYEAR(40),XSYEAR(60),L                                AUT11970
      COMMON/$PROST/IGAP,PRONAM,DOPMAT(5,25,3),BESPRO(60,26,3)          AUT11980
C-----------------------------------------------------------------------AUT11990
C   TEST FOR NEGATIVES AND CONVERT TO ZERO AS NEGATIVE PRODUCTION IS    |AUT12000
C   NOT FEASIBLE.                                                       |AUT12010
C-----------------------------------------------------------------------AUT12020
      DO 6 J=1,3                                                        AUT12030
      DO 6 ICOL2=1,L                                                    AUT12040
      I=K                                                               AUT12050
      DO 5 ICK=1,K                                                      AUT12060
C-----------------------------------------------------------------------AUT12070
C   CHANGE THE FOLLOWING STEP IF CYCLICAL REGRESSION IS ADDED, AS THIS  |AUT12080
C   SECTION SEARCHES FROM THE BOTTOM UP FOR A POSITIVE NUMBER AND WHEN  |AUT12090
C   ONE IS FOUND STOPS REPLACING NEGATIVES WITH ZERO.  IN CYCLICAL RE-  |AUT12100
C   GRESSION NEGATIVES COULD BE FOUND THROUGHOUT THE PROJECTIONS.       |AUT12110
C-----------------------------------------------------------------------AUT12120
      IF(BESPRO(I,ICOL2,J).GE.0) GO TO 6                                AUT12130
      BESPRO(I,ICOL2,J)=0.0                                             AUT12140
      I=I-1                                                             AUT12150
    5 CONTINUE                                                          AUT12160
    6 CONTINUE                                                          AUT12170
C---------------------------------                                      AUT12180
C   SET DISTRICT COLUMNS TO ZERO. |                                     AUT12190
C---------------------------------                                      AUT12200
      ICOL2=L+1                                                         AUT12210
      DO 1 J=1,3                                                        AUT12220
      DO 1 I=1,K                                                        AUT12230
    1 BESPRO(I,ICOL2,J)=0.0                                             AUT12240
      DO 3 I=1,N                                                        AUT12250
    3 YBASE(I,ICOL2)=0.0                                                AUT12260
C---------------------------------------                               AUT12270
C   SUM ROWS OF BASE AND PROJECTED DATA. |                              AUT12280
C   MATRICES TO GET DISTRICT TOTALS.     |                              AUT12290
C---------------------------------------                               AUT12300
      DO 2 J=1,3                                                        AUT12310
      DO 2 I=1,K                                                        AUT12320
      DO 2 ICOL=1,L                                                     AUT12330
    2 BESPRO(I,ICOL2,J)=BESPRO(I,ICOL2,J)+BESPRO(I,ICOL,J)              AUT12340
      DO 4 I=1,N                                                        AUT12350
      DO 4 ICOL=1,L                                                     AUT12360
    4 YBASE(I,ICOL2)=YBASE(I,ICOL2)+YBASE(I,ICOL)                       AUT12370
      RETURN                                                            AUT12380
      END                                                               AUT12390
```


SUBROUTINE SUMDIS
COMMON
DO J=1,3
DO ICOL2 =1,L
I=K
DO ICK =1,K
BESPRO(I,ICOL2,J) ≥ 0 ?
T
F
BESPRO(I,ICOL2,J) = 0.0
I=I-1
ICOL2=L+1
DO J =1,3
DO I =1,K
BESPRO(I,ICOL2,J) = 0.0
TO 2

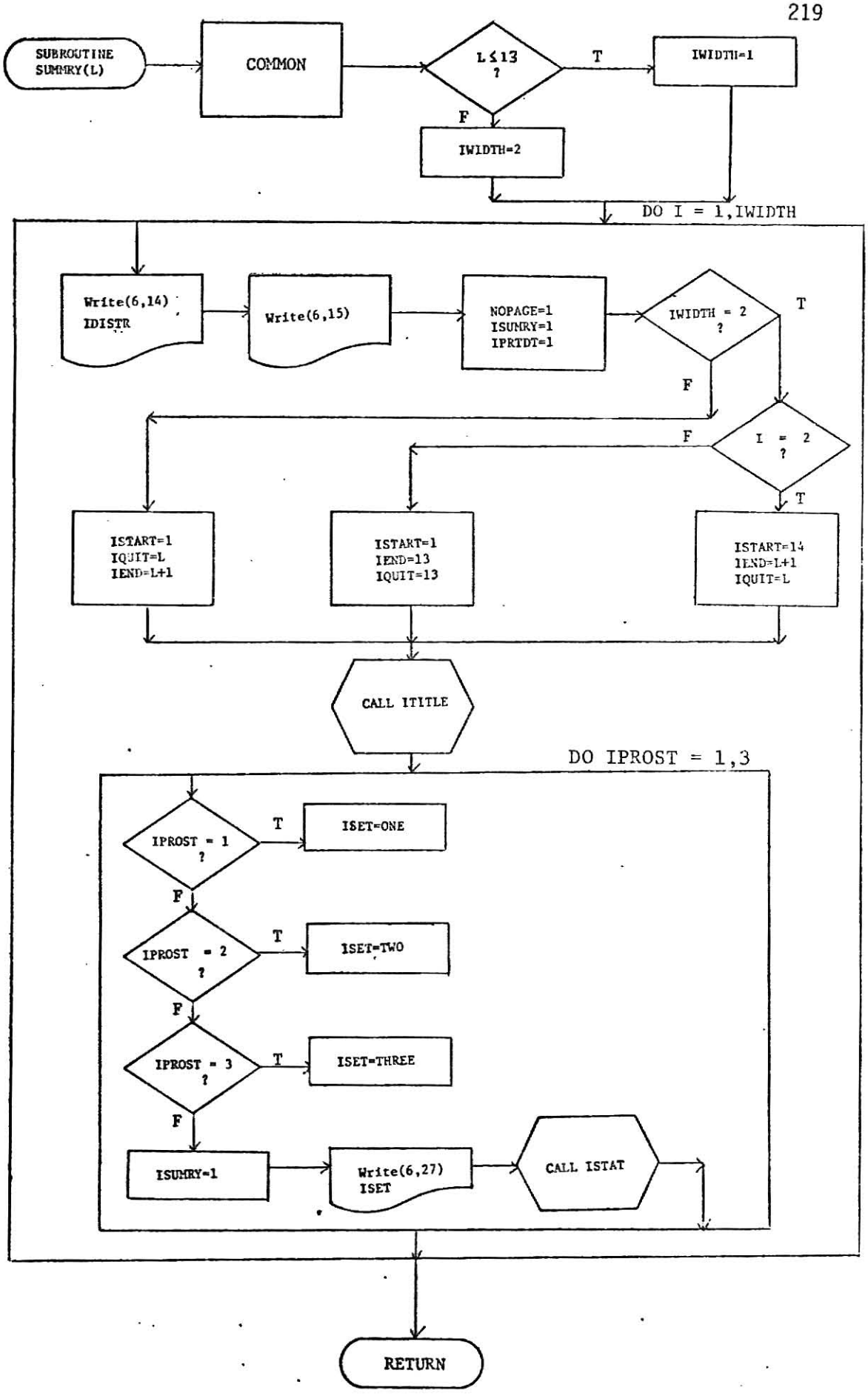

2
DO I =1,N
YBASE(I,ICOL2) = 0.0
DO J =1,3
DO I =1,K
DO ICOL=1,L
BESPRO(I,ICOL2,J)=
BESPRO(I,ICOL2,J)+
BESPRO(I,ICOL,J)
DO I =1,N
DO ICOL =1,L
YBASE(I,ICOL2)=
YBASE(I,ICOL2)+
YBASE(I,ICOL)
RETURN

XVI. Subroutine Summry

This subroutine prints a copy of all statistics for a district, for each projection set, immediately after the district title page. The statistics are repeated at the end of each projection set, but this subroutine puts them all together on one or two pages for easy comparative purposes.

Summry works very similar to Idist in that it defines parameters and then calls Ititle and Istat to do most of the actual printing. The setting of the flag ISUMRY to 1 causes Ititle to omit printing the line containing the projection set and district number. It causes ISTAT to omit printing the footnotes. This is done because the use of ISTAT and ITITLE in SUMMRY doesn't require the printing of these omitted lines.

```
      SUBROUTINE SUMMRY(L)                                                AUT12400
C-----------------------------------------------------------------------AUT12402
C   THIS SUBROUTINE PRINTS SUMMARY STATISTICS FOR EACH DISTRICT.          |AUT12404
C-----------------------------------------------------------------------AUT12406
      COMMON/C/IDISTR,IPRTDT,LINE1(33),LINE2(33),LINE3(33),LINE4(65),LINAUT12410
     1E5(31),LINE6(31),IQUIT                                              AUT12420
      COMMON/G/ISTART,IEND,KFLINE,KFLINE,IPROST,ISKIP                     AUT12430
      COMMON/$PAGE/NOPAGE                                                 AUT12440
      COMMON/$SUM/ISUMRY                                                  AUT12450
      INTEGER*4 ONE/'1ST'/,TWO/'2ND'/,THREE/'3RD'/                        AUT12460
      IF(L.LE.13) GO TO 51                                                AUT12470
      IWIDTH=2                                                            AUT12480
      GO TO 32                                                            AUT12490
   51 IWIDTH=1                                                            AUT12500
   32 DO 30 I=1,IWIDTH                                                    AUT12510
      WRITE(6,14) IDISTR                                                  AUT12520
   14 FORMAT('1',48X,'SUMMARY STATISTICS FOR DISTRICT ',A3/'+',43X,35('_AUT12530
     *'))                                                                 AUT12540
      WRITE(6,15)                                                         AUT12550
   15 FORMAT('-')                                                         AUT12560
      NOPAGE=1                                                            AUT12570
      ISUMRY=1                                                            AUT12580
      IPRTDT=1                                                            AUT12590
      IF(IWIDTH.EQ.2) GO TO 10                                            AUT12600
C-----------------------------------------------------------------------AUT12602
C   SET PARAMETERS FOR WRITING 1ST PAGE OF OUTPUT.                        |AUT12604
C-----------------------------------------------------------------------AUT12606
      ISTART=1                                                            AUT12610
      IQUIT=L                                                             AUT12620
      IEND=L+1                                                            AUT12630
      GO TO 11                                                            AUT12640
   10 IF(I.EQ.2) GO TO 12                                                 AUT12650
C-----------------------------------------------------------------------AUT12652
C   SET PARAMETERS FOR WRITING 2ND PAGE OF OUTPUT.                        |AUT12654
C-----------------------------------------------------------------------AUT12656
      ISTART=1                                                            AUT12660
      IEND=13                                                             AUT12670
      IQUIT=13                                                            AUT12680
      GO TO 11                                                            AUT12690
   12 ISTART=14                                                           AUT12700
      IEND=L+1                                                            AUT12710
      IQUIT=L                                                             AUT12720
   11 CALL ITITLE                                                         AUT12730
C-----------------------------------------------------------------------AUT12732
C   WRITE ONE PAGE WIDTH OF EACH PROJECTION SET'S STATISTICS, ALL ON      |AUT12734
C   PAGE.                                                                 |AUT12736
C-----------------------------------------------------------------------AUT12738
      DO 30 IPROST=1,3                                                    AUT12740
      IF(IPROST.EQ.1) ISET=ONE                                            AUT12750
      IF(IPROST.EQ.2) ISET=TWO                                            AUT12760
      IF(IPROST.EQ.3) ISET=THREE                                          AUT12770
      ISUMRY=1                                                            AUT12780
      WRITE(6,27) ISET                                                    AUT12790
   27 FORMAT('-',A3,' PROJECTION SET')                                    AUT12800
      CALL ISTAT                                                          AUT12810
   30 CONTINUE                                                            AUT12820
      RETURN                                                              AUT12830
      END                                                                 AUT12840
```


SUBROUTINE SUMMRY(L)
COMMON
L≤13 ?
T
IWIDTH=1
F
IWIDTH=2
DO I = 1,IWIDTH
Write(6,14) IDISTR
Write(6,15)
NOPAGE=1
ISUMRY=1
IPRTDT=1
IWIDTH = 2 ?
T
F
I = 2 ?
F
T
ISTART=1
IQUIT=L
IEND=L+1
ISTART=1
IEND=13
IQUIT=13
ISTART=14
IEND=L+1
IQUIT=L
CALL ITITLE
DO IPROST = 1,3
IPROST = 1 ?
T
ISET=ONE
F
IPROST = 2 ?
T
ISET=TWO
F
IPROST = 3 ?
T
ISET=THREE
F
ISUMRY=1
Write(6,27) ISET
CALL ISTAT
RETURN

REFERENCES

1)Barr, Anthony J., Goodnight, James H., Sall, John P., Helwig, Jane T., A User's Guide to SAS 76, SAS Institute, Inc., Raleigh, North Carolina, 1976.

2)Box, G. E., and Jenkins, G. M., Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco, 1970.

3)Daniel, Cuthbert, and Wood, Fred. S., Fitting Equations to Data, Wiley--Interscience, New York, 1971.

4)Dawson, J. A., "Curve fitting by least squares methods., Computer Applications in the Natural and Social Sciences, No.3, Aug. 1969.

5)Dixon, William J., ed., Biomedical Computer Programs, University of California Press, Berkley, 1975.

6)Hamburg, Morris, Statistical Analysis for Decision Making, Harcourt, Brace and World, New York, 1970.

7)IBM, System/360 Scientific Subroutine Package Version III, Programmer's Manual, IBM, Kingston, New York, 1968.

8)Kemp, K. E., AARDVARK, Department of Statistics, Statistical Computing Laboratory, Kansas Agricultural Experiment Station, Contribution 209, Kansas State University, Manhattan, Kansas, 1973.

9)Newbold, Paul, "The principles of the Box-Jenkins approach., Oper. Research Quarterly, 26, No.2, July 1975.

10)Nie, Norman H., Hull, C. Hadai, Jenkins, Jean G., Streinbrenner, Karin, and Bent, Dale, SPSS--Statistical Package for the Social Sciences, Mc-Graw Hill, Inc., New York, 1975.

11)Ostle, Bernard and Mensing, Richard W., Statistics in Research, 3rd ed., Iowa State University Press, 1975.

12)Phillips, Richard, Tiao, Joe M., and Schruben, Leonard W., User's Guide to Computerized System for Agribusiness Development, Food and Feed Grain Institute, Kansas State University, Manhattan, 1975.

13)Rosenblatt, Murray, Ed., Proceedings of the Symposium on Time Series Analysis, John Wiley & Sons, Inc., New York, 1963.

14)Schruben, Leonard, Turner, Walter, Tiao, Joe, Jeon, Doyle, and Carlson, June, How Extending Navigation of the Trinity River, Texas

Would Affect Transportation Costs of Wheat, Department of Agricultural Economics, Agricultural Experiment Station, Contribution 632-a, Kansas State University, Manhattan, Kansas, May, 1977.

15)Smillie, K. W., Statpack 2--An APL Statistical Package, 2nd ed., Department of Computing Science, Pub. No. 17, The University of Alberta, Canada, 1969.

16)Thornber, Hodson, B34T--A Stepwise Regression Program, Center for Mathmatical Studies in Business and Economics, University of Chicago, Illinois, 1966.

A COMPUTERIZED SYSTEM FOR PROJECTING CROP DATA ON A COUNTY AND DISTRICT BASIS

by

WALTER LYNN TURNER

B. S., Kansas State University, 1974
M. B. A., Kansas State University, 1976

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Agricultural Economics

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1977

The techniques used in the development of a computerized system for projecting crop data are discussed. Two computer programs were actually developed. One, an interactive terminal program, is included for use in developing further projection methods and the projection of one data set at a time. The other program, handles multiple county data sets and summarizes the results by crop reporting district.

Briefly, the computer programs subject a data set (or sets) to various projection models and then select the best three projections, on the basis of R^2, and print them. The projection models included in the system are 1)linear $Y = a + bx$, 2)logarithmic $Y = a + b(\log x)$, and 3)exponential $Y = a + b(x^c)$, where c is an exponent between 0.4 and 1.4, inclusive. The model selection is under control of an internal decision tree.

The implications of this computer system are discussed. In addition, a real-world application of the system and a discussion of the results are presented.