1976

TRANSLATION OF THE SPARKS PREPROCESSOR FROM FORTRAN TO SPARKS

by

RICHARD MANSON STROUD

B.S., UNIVERSITY OF TEXAS, ARLINGTON, 1972

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1978

Approved by:

Myron A. Calhoun

MAJOR PROFESSOR

Document
LD
2668
.R4
1978
S86
C. 2

TABLE OF CONTENTS

ILLEGIBLE

THE FOLLOWING DOCUMENT (S) IS ILLEGIBLE DUE TO THE PRINTING ON THE ORIGINAL BEING CUT OFF

ILLEGIBLE

CHAPTER 1

INTRODUCTION

1.0 Purpose.

The purpose of this report is to describe the translation of the SPARKS preprocessor from FORTRAN to the SPARKS language. The report is organized such that the chapters dealing with the documentation of the preprocessor and the updated user's guide can be detached. The preprocessor written in SPARKS is included at appendix A. The FORTRAN translation of that code is at appendix B.

1.1 Explanation of SPARKS Preprocessor.

The sole purpose of the SPARKS preprocessor is to translate SPARKS statements into FORTRAN. Why SPARKS? And for that matter, why FORTRAN? Answering the second question first, FORTRAN has become established as the primary language for scientific and engineering computation. It also has the distinction of being the earliest 'high' level programming language, developed about 1957 at IBM. Because of its age, FORTRAN has many positive attributes such as:

a. It is almost always available and its compilers are often good.

b. There is a language standard which allows a degree of portability not obtainable with other languages.

c. There are extensive subroutine libraries available.

d. There is a large labor force already familiar with

FORTRAN.

However, FORTRAN has certain negative aspects such as the fact that it was developed before the concepts of structured programming became established. It is essentially an unstructured and nonmodular language. Now, why SPARKS? The SPARKS language adds the improved syntactical constructs to FORTRAN which allow structured and modular code. Thus, the SPARKS preprocessor provides a simple means to obtain a nicely structured language while preserving the virtues of FORTRAN.

1.2 History of SPARKS.

SPARKS originated as a language for describing algorithms in the textbook Fundamentals of Data Structures by Ellis Horowitz and Sartaj Sahni, published by Computer Science Press, Woodland Hills, California. The SPARKS preprocessor which grew out of this prototype language was orginally written in SPARKS. The program was then hand translated into FORTRAN to produce the first working model. During the hand translation and implementation, the language was revised and no longer exactly matches the orginal textbook version.

1.3 Translation of the Preprocessor.

Rewriting the SPARKS preprocessor in SPARKS brings to the translation program itself the structure for easy

readability and understanding and the necessary modularity to make modifying the program easier. The modularity of SPARKS can be illustrated by the ease in which the 20 seperate cases of the main program case statement can be rearranged. The cases can be moved about within the case statement by simply taking the cards belonging to that case and replacing them where desired. This same action in the FORTRAN copy would be a major undertaking as many labelled statements and GO TO's would have to be changed.

1.4 Conclusion.

The measure of a computer language, good or bad, is the ease with which it allows a programmer to describe a real world problem in a natural way. The SPARKS language improves FORTRAN toward this end. The preprocessor itself, written in SPARKS, is a program much easier to read, understand, and modify.

CHAPTER 2

TRANSLATION INTO SPARKS

2.0 Introduction.

The writting of the translator in SPARKS involved two distinct steps. The overall structure of a block of the FORTRAN was recognized and then its sub-structures found. The preprocessor was hand translated to SPARKS using this top-down approach.

2.1 Translation of FORTRAN Statements.

The first step in the translation process was to recognize the overall structure of each routine. For example, the large main routine consists of a single loop forever with a case statement within the loop. The loop covers 12 pages of code and the case statement has 20 seperate cases over those 12 pages of code. Once the overall structure of the routine was discovered, the individual constructions within the routine were recognized and converted to equilivant SPARKS code. This method of finding the structures (large to small) was followed to produce the entire translator in handwritten SPARKS code. Shown in the following paragraphs is how the actual FORTRAN code was recognized as convertable to a particular SPARKS construction. Also discussed, are those instances where the machine translation of the SPARKS code will not reproduce the exact FORTRAN of the orginal. The FORTRAN IF statement

was generally easy to convert to SPARKS as shown with the nested IF statement example.

ORIGINAL FORTRAN

```
      IF(INDFLG.NE.BKWRD) GO TO 12000
         SSTRT=SSTRT-INCR
         IF(CURFLG.EQ.NEW) CALL SPRNT
         NESTNO=NESTNO-1
         BLKNO=BLKNO+1
12000 CONTINUE
```

TRANSLATION TO SPARKS

```
      IF INDFLG.EQ.BKWRD THEN
         SSTRT=SSTRT-INCR
         IF CURFLG.EQ.NEW THEN
            CALL SPRNT
         ENDIF
         NESTNO=NESTNO-1
         BLKNO=BLKNO+1
      ENDIF
```

The inner IF statement is an example where the machine translation will not reproduce the orginal FORTRAN code. SPARKS has no translation to the inner IF construction. The

following illustrates the preprocessor conversion.

SPARKS

```
IF CURFLG.EQ.NEW THEN
   CALL SPRNT
ENDIF
```

MACHINE TRANSLATION TO FORTRAN

```
      IF(.NOT.(CURFLG.EQ.NEW)) GO TO 11000
         CALL SPRNT
11000 CONTINUE
```

Although the meaning of the original FORTRAN and the SPARKS preprocessor generated FORTRAN is the same, it is apparent that care must be taken when hand translating to SPARKS. The relational operator may or may not require negation depending on the type of FORTRAN IF to be translated. A particular sequence of code with the FORTRAN IF can signal the SPARKS IF/THEN/ELSE construction. The following FORTRAN sequence clearly indicates the IF/THEN/ELSE construct. ECOND is a logical.

FORTRAN

```
IF(.NOT.ECOND) GO TO 1920
   CALL ERROR(2,DOO,2,1,0,0)
   TOEND=TOEND+1
   GO TO 1910
```

```
1920 TCEND=TOEND+2
1910 CONTINUE
```

TRANSLATION TC SPARKS

```
IF ECOND THEN
   CALL ERROR(2,DOO,2,1,0,0)
   TOEND=TCEND+1
ELSE
   TOEND=TCEND+2
ENDIF
```

Again, the relational operator can be changed or negated to reverse the THEN and ELSE cases but care must be used so as not to change the meaning of the orginal code.

Many nested IF statements or IF statements which branch to another IF on the true branch are convertable to the SPARKS case statement. The improved readability and modularity of SPARKS is nicely shown by the CASE example.

FORTRAN

```
IF(STTOP.NE.FFOR).GO TO 2090
   LAB1=STACK(TCP-5)
   LAB2=STACK(TCP-2)
   LAB3=STACK(TCP-1)
   TCP=TOP-6
```

```
            LBPTR=LBPTR-20
            GO TO 2000
2090 IF(STTOP.NE.LLOOP) GO TO 2080
            LAB1=STACK(TOP-3)
            LAB2=STACK(TOP-2)
            LAB3=STACK(TOP-1)
            TOP=TOP-5
            LBPTR=LBPTR-15
            GO TO 2000
2080 IF(STTOP.NE.WHILE) GO TO 2070
            LAB1=STACK(TOP-3)
            LAB2=STACK(TOP-2)
            LAB3=STACK(TOP-1)
            TOP=TOP-15
            LBPTR=LBPTR-15
            GO TO 2000
2070 CONTINUE
            CALL ERROR(3,REPT,6,4,REPT,6)
2000 CONTINUE
```

TRANSLATION TO SPARKS

```
CASE
   :STTOP.EQ.FFOR:
      LAB1=STACK(TOP-5)
      LAB2=STACK(TOP-2)
      LAB3=STACK(TOP-1)
```

```
            TOP=TOP-6
            LBPTR=LBPTR-20
         :STTOP.EQ.LLOOP:
            LAB1=STACK(TOP-3)
            LAB2=STACK(TOP-2)
            LAB3=STACK(TOP-1)
            TOP=TOP-5
            LBPTR=LBPTR-15
         :STTOP.EQ.WHLE:
            LAB1=STACK(TOP-3)
            LAB2=STACK(TOP-2)
            LAB3=STACK(TOP-1)
            TOP=TOP-5
            LBPTR=LBPTR-15
         :ELSE:
            CALL ERROR(3,REPT,6,4,REPT,6)
      ENDCASE
```

SPARKS allows four loop constructions: The FOR loop, the LOOP/REPEAT, the WHILE loop, and the LOOP/UNTIL. The task of recognizing a loop construction in FORTRAN, other than the obvious DO loop, involves finding a test condition and a backward GO TO. The following examples show how the SPARKS loop constructions were used to convert the FORTRAN to SPARKS. Treating the easier DO loop first, it is replaced by the SPARKS FOR loop.

FORTRAN

```
DC 1980 I=1,5
   L=LAB3+I-1
1980 FBUF(I)=LABEL(L)
```

TRANSLATION TC SPARKS

```
FCR I=1 TO 5 DC
   L=LAB3+I-1
   FBUF(I)=LABEL(L)
REPEAT
```

This is another case where the machine translation of the SPARKS will not reproduce the original FCRTRAN. There is no SPARKS construction that converts tc a FCRTRAN DO loop. The SPARKS FOR loop is more powerful than shown above in that it will allow incrementation by any integer, either positive or negative. The above illustraticn shows the default incrementation of +1. The SPARKS preprocessor handles the translaticn of the FOR loop in an interesting way. The example below is of a negative increrenting loop and its FCRTRAN translaticn.

SPARKS

```
FCR I=10 TC 2 BY -2 DO
   J=LAB3+I-1
REPEAT
```

MACHINE TRANSLATION TO FORTRAN

```
      I=10
      GO TO 100
  102 I=I+(-2)
  100 IF(.NOT.(I-(2))*(-2).LE.0) GO TO 101
      J=LAB3+I-1
      GO TO 102
  101 CONTINUE
```

The example illustrates that although the FORTRAN DO loop can be easily replaced by a SPARKS FOR loop, the machine translation of the FOR loop into FORTRAN in no way resembles the original code. The other loop structures were also used in the translation to replace the FORTRAN conditional tests and backward GO TO's. A WHILE loop conveniently replaces the FORTRAN code shown below.

FORTRAN

```
 3401 IF(SPIN(TOBEG).NE.BLANK) GO TO 3402
         TOBEG=TOBEG+1
         IF(TOBEG.GT.IMAX) GO TO 3403
         GO TO 3401
 3403 TOEND=IMAX+1
 3402 CONTINUE
```

TRANSLATION TO SPARKS

```
WHILE SPIN(TOBEG).EQ.BLANK DO
   TOBEG=TOBEG+1
   IF TOBEG.GT.IMAX THEN
      TOEND=IMAX+1
      EXIT
   ENDIF
REPEAT
```

The SPARKS construction EXIT, used above, allows premature leaving of a loop structure. Similarly, the SPARKS CYCLE statement is used to transfer to the bottom of a loop from anywhere within the loop structure. The SPARKS LOOP/REPEAT and the EXIT are used to translate the following FORTRAN code.

FORTRAN

```
1090 IF(K.GT.IMAX) GO TO 2140
        IF(SPIN(K).NE.BLANK) GO TO 1091
           K=K+1
           GO TO 1090
1091    I=ISNUM(SPIN(K))
        GO TO 2150
2140    I=0
2150 CONTINUE
```

TRANSLATION TO SPARKS

```
LOOP
   IF K.GT.IMAX THEN
      I=0
      EXIT
   ENDIF
   IF SPIN(K).NE.BLANK THEN
      I=ISNUM(SPIN(K))
      EXIT
   ENDIF
   K=K+1
REPEAT
```

The last SPARKS loop structure, the LOOP/UNTIL is used to convert the FORTRAN below.

FORTRAN

```
10651 TTOP=TTOP-STFRM(TTOP)
      IF(TTOP.GT.0) GO TO 10652
         CALL ERROR(3,EEEND,3,4,EEEND,3)
         ECOND=.TRUE.
         RETURN
10652 CONTINUE
      STTOP=STACK(TTOP)
      IF(STTOP.NE.SUBFN) GO TO 10651
```

TRANSLATION TO SPARKS

```
LCOP
   IF TTOP.LE.0 THEN
      CALL ERROR(3,EEEND,3,4,EEEND,3)
      ECOND=.TRUE.
   ENDIF
   STTOP=STACK(TTOP)
UNTIL STTOP.EQ.SUBFN REPEAT
```

Some of the FORTRAN code was modified during the original hand translation from SPARKS to FORTRAN to the extent that it did not match any SPARKS constructions. The subroutine DETNX is cited as an example. This subroutine's overall structure was not convertable to any single SPARKS construction. Essentially, the intent of the FORTRAN code had to be determined exactly and SPARKS code generated as required. The overall construction of the subroutine in SPARKS consists of two nested LOOP-REPEAT constructions which bear little resemblence to the original FORTRAN code. There are other instances where it was necessary to rearrange the SPARKS code completely in order to eliminate retaining the FORTRAN GO TO. The completed translation to SPARKS does not, in any case, retain the FORTRAN IF statement, the GO TO in any form, or the DO loop in the program. Of course, the statements common to both FORTRAN and SPARKS were retained to avoid punching duplicate cards. The first SPARKS deck varied less than four per cent in size

from the original FORTRAN deck.

2.2 Implementation.

Loading and debugging of the SPARKS language preprocessor was the next step in the translation process. It was decided that the SPARKS language preprocessor should be loaded and debugged one subroutine at a time. This procedure would side-step the problem of trying to find errors in the 1845 card SPARKS deck. The original FORTRAN language preprocessor was copied onto a temporary disk file. The SPARKS preprocessor originally brought to Kansas State University was not disturbed and provided the means to translate the new SPARKS language version to FORTRAN. The new subroutines were processed through and linked to the FORTRAN copy on the temporary disk. After each subroutine was converted, a test deck was run. Each subroutine was completely debugged before another was added. When all subroutines had been replaced, the main routine was replaced and the temporary disk was complete with the machine translation of the new SPARKS language preprocessor. At this point, two copies of the SPARKS deck were made. One copy was retained as a record copy and one copy was provided to another Graduate Student, John J. Martin, for use in his Master's project.

During the initial loading, two errors were discovered

in the original FORTRAN version of SPARKS: Large programs could not be loaded as there was a error in the method of reclaiming the label pointer positioning. Consequentially, large programs would over-drive the label table and the program would terminate unsuccessfully. Also, the SCAN1 subroutine which normally stops on a semicolon incorrectly translated format and data statements containing semicolons and produced fatal FORTRAN errors. These errors made it impossible to load the entire preprocessor as a single program, and FORTRAN flags (F in column 1) had to be placed around all format and data statements. It was found that John Martin had already developed corrections for these two errors as a part of his Master's project and those corrections were incorporated in the working deck.

2.3 Program Improvements.

Once a copy of the preprocessor written in SPARKS was obtained, the code was examined in detail. Many occurances of redundant code were found as well as several unused variables. The following variables were eliminated from the program:

a. HECH-was not used.

b. COOLN-duplicate variable name, not needed

d. NINE-retained as a subroutine local variable, only.

e. CODEO-was not used.

f. CODE9-was not used.

g. IBLD-eliminated as unnecessary.

h. ITIN, ITOT, ITRD, INRD, ITERS, ITPRF, INPRS, INPRF, INFPR, INSPR, INSN1-statistics gathering variables, all eliminated.

The variable STBEG, which marks the beginning of all input statements, was set once in each separate case of the Main Program. It was noted that the beginning of the line is determined prior to entering the Main Routine CASE statement. Therefore, STBEG could be set prior to entry into the CASE statement and 19 redundant cards removed from the Main Routine. The following block of code was found to appear frequently in the Main Routine:

```
FSTRT=FSTRT+INCR
IF FSTRT.GT.IMAX THEN
   FSTRT=FSTRT-INCR
ENDIF
```

The block was rewritten as follows saving two unnecessary statements.

```
IF FSTRT+INCR.LT.IMAX THEN FSTRT=FSTRT+INCR ; ENDIF
```

It was found that blanks were placed in the first six columns of the FORTRAN output buffer (FBUF) by almost every separate case of the Main Routine by repetition of the following code:

```
FOR I=1 TO 6 DO
```

```
FBUF(I)=BLANK
REPEAT
```

A location in the first subroutine, DETNX, was located where the buffer could be filled for each output line. The above code was then eliminated from the Main Routine. Instances were also found where identical flags were set either in both the true and false branches of an IF statement or repeated in consecutive IF statements. In all of these instances, the flags to be set were moved so as to eliminate repetitious code.

Examination of the CYCLE and EXIT statement-handling routines in the Main Routine CASE statement revealed that the code was practically identical for both statements. A subroutine was created to process both the CYCLE and EXIT statements. The only code remaining in the Main Routine for the EXIT and CYCLE statements is a call to the CHECK Subroutine and a call to the new subroutine, TYHAND.

Several occurences were found where the SPARKS code could be improved by rearranging and changing the construction used. Reference is made to the CASE conversion example of section 2.1, where it can be seen that the code for the LOOP case and the WHILE case is identical. Also, the ELSE case was redundant as the CHECK Subroutine had been called prior to entry into the case. With these

observations, the CASE can be replaced by the IF-THEN-ELSE construction using much less code.

```
IF STTOP.EQ.FFOR THEN
   LAB1=STACK(TOP-5)
   LAB2=STACK(TOP-2)
   LAB3=STACK(TOP-1)
   TOP=TOP-6
   LBPTR=LBPTR-20
ELSE
   LAB1=STACK(TOP-3)
   LAB2=STACK(TOP-2)
   LAB3=STACK(TOP-1)
   TOP=TOP-5
   LBPTR=LBPTR-15
ENDIF
```

The last items to be investigated were the incorrect error messages, the method of declaring integers and arrays, and the alignment of COMMON blocks. The error messages output by the preprocessor were either incorrect or had missing words or letters. The problem found was that the error variables had been equivalenced to positions in the program's reserved word table, but apparently the ordering within the table had been changed and subsequently the equivalences were now to the wrong positions. The problem

was solved by eliminating the equivalence statements and assigning values to the ERROR subroutine's variables by DATA statements in the BLOCK DATA routine. The integer declarations were made originally by long lists of integer statements listing every variable and array. Also, the arrays were dimensioned in the integer statements. Each variable and array was again listed in the COMMON blocks. This double listing of the variables in each routine could be eliminated by the use of the implicit integer feature. It should be noted that the program is no longer ANSI Standard FORTRAN with the inclusion of the implicit integer feature. The dimensioning of the arrays was included in the COMMON blocks. A closer look at the named COMMON blocks indicated that they were not optimized for the routines that used them; there were many occurences where a named common of up to 21 variables was included in a routine which used only one of the variables. Thus, in addition to adding the dimensioning to the common blocks, they were re-aligned and/or split so that each routine included fewer common statements.

2.4 Conclusion.

The resulting SPARKS language preprocessor is a 1470 line program versus the 1875 lines of the original FORTRAN version. It has one less case in the main routine and one additional subroutine.

CHAPTER 3

PROGRAM DOCUMENTATION

3.0 Introduction.

The SPARKS preprocessor is a 1470 line program consisting of a Main Routine, a Block Data, 11 Subroutines, and three Integer Functions. The preprocessor written in the SPARKS language is at APPENDIX A. The machine translation of this program into FORTRAN is at APPENDIX B. The module access diagram for the program is shown in Figure 1. The data flow of the preprocessor is in Figure 2. The global variables used by the program and the routines in which they are used are included in Paragraph 3.2 of this chapter. The common blocks and the variables they include are shown in Paragraph 3.3. Figures 1 and 2 are given on the next two pages.

FIGURE 1

Module Access Diagram for the SPARKS Preprocessor.

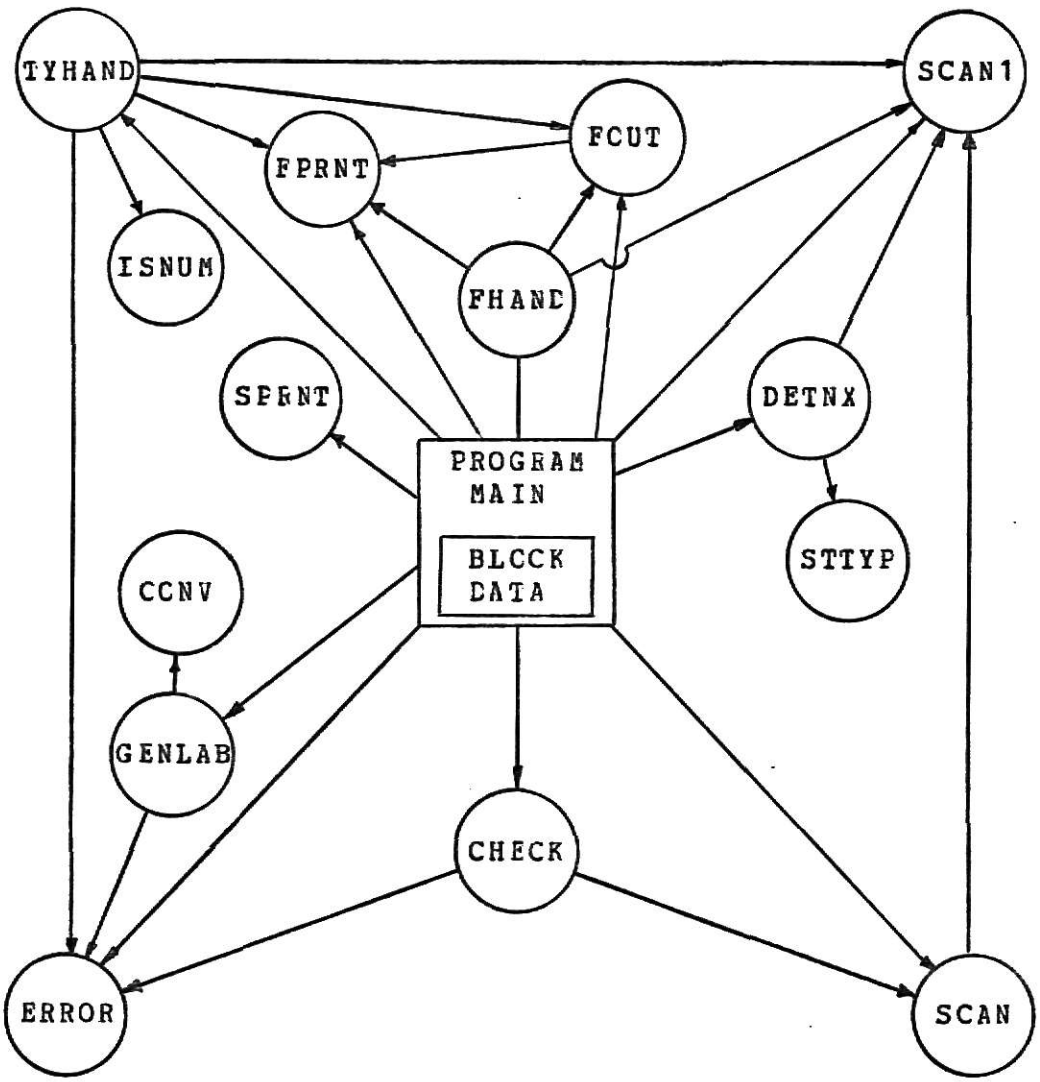
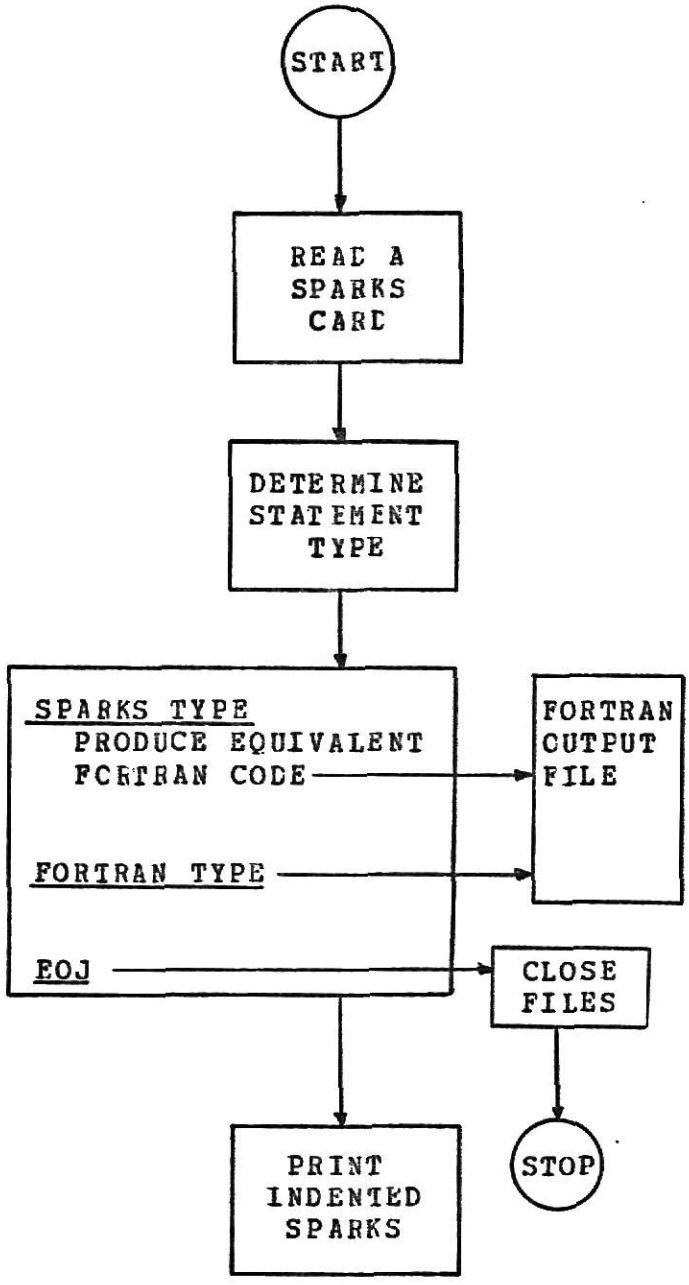


FIGURE 2

Data Flow Diagram for the SPARKS Preprocessor.



CHAPTER 3

3.1 Main Routine and Subroutine Specifications.

In the paragraphs to follow, each routine of the program is outlined as to its function, the common blocks it includes and pseudo code showing what the routine does.

MAIN ROUTINE

FUNCTION: The main routine causes a card to be read (subroutine DETNX reads the card) and the type of statement to be determined (again by subroutine DETNX which calls STTYP). The main then processes the statements by type. The FORTRAN output is generated within the main routine CASE statement which is organized by statement type. The SPARKS output is accomplished at the end of the main for each input line after the FORTRAN for that line has been generated. The main routine includes the following common blocks: INPUT, SPRINF, FRTINF, CENTRL, CTPAT, INPAT, STTYP1, STTYP2, MNMIC, KEYWD, TABLES, and TYP.

ABSTRACT CODE

```
BEGIN
   Initialize program options
   LOOP
     Cause a SPARKS card to be read (Call to DETNX).
     Determine statement type (Call to DETNX, STTYP).
     If a real, integer or logical, check for
        subroutine or function.  If found, set type to
        SUBFN.
     CASE
```

```
: type is FORTRAN :
  Pass the FORTRAN to the FORTRAN file.
: type is double slash :
  Process SPARKS comment.
  Translate comment to FORTRAN, pass to FORTRAN
     file.
 : type is an IF :
  Place IF on stack.
  Translate to a FORTRAN IF and output to
     FORTRAN file.
 : type is an ENDIF :
  Check stack top for an IF statement, if
     none found then error, CYCLE.
  Generate necessary labels and CONTINUES and
     output to the FORTRAN file.
 : type is an ELSE :
  Check stack top for CASE or IF statement, if
     none found then error, CYCLE.
  Convert to FORTRAN and output to FORTRAN
     file.
 : type is a FOR :
  Place FOR on stack.
  Search for key words TO, BY, and DO, if
     missing then error, CYCLE.  Convert
     to FORTRAN and output to FORTRAN file.
 : type is a REPEAT :
```

```
      Check stack top for an iterative statement;
         if none found then error, CYCLE.
      Convert to FORTRAN, output to FORTRAN file.
    : type is a CASE :
      Put CASE on stack.
      Convert to FORTRAN, output to FORTRAN file.
    : type is a COLON :
      Check stack for a CASE statement, if none
         found then error, CYCLE.
      Convert to FORTRAN, output to FORTRAN file.
    : type is an ENDCASE :
      Check stack top for a CASE statement, if
         none found then error, CYCLE.
      Convert to FORTRAN, output to FORTRAN file.
    : type is a LOOP :
      Put LOOP on stack.
      Convert to FORTRAN, output to FORTRAN file.
    : type is a WHILE :
      Put WHILE on stack.
      Convert to FORTRAN, output to FORTRAN file.
    : type is a CONTINUE :
      Convert to FORTRAN, output to FORTRAN file.
    : type is an EXIT or CYCLE :
      Check stack top for iterative statement, if
         none found then error, CYCLE.
      Convert to FORTRAN, output to FORTRAN file.
```

```
        : type is an UNTIL :
          Check stack top for ICCP statement,
             if none found then error, CYCLE.
          Convert to FORTRAN, output to FORTRAN file.
        : type is a SUBROUTINE or FUNCTION :
          Stack the SUBFN.
          Re-initialize label numbers, skip to new
             page, output to FORTRAN file.
        : type is an END :
          Check stack top for PROGRAM, SUBROUTINE, or
             FUNCTION; if not found then error, CYCLE.
          Output to FORTRAN file.
        : type is a FORTRAN comment :
          Check for C in column 1 and output to FORTRAN
             file.
        : type is an EOJ :
          Close the files.
          STOP.
        ENDCASE
        Set SPARKS output indentation and output
           to SPARKS print.
     REPEAT
  END
```

SUBROUTINE DEINX

FUNCTION: The subroutine reads a card and returns to

the main with the statement type. The FORTRAN flag (F in column 1) is also handled by this routine. Code between the flags is passed directly to the FORTRAN file without processing. The subroutine includes the following common blocks: INPUT, FRTINF, STTYP2 and MNMIC.

ABSTRACT CODE

```
BEGIN
   LOOP
      If the end of line pointer exceeds 72,
         EXIT.
      If a semicolon is found then CYCLE.
      If a C is found in column 1 then type
         is a COMMENT, RETURN.
      Look for a blank or end condition (by a call
         to SCAN1).
      CASE
        : No blanks prior to end condition:
          Type is FORTRAN, RETURN.
        : Column six is not blank :
          Type is a continuation card, RETURN.
        : A blank is found prior to END condition :
          Call to Subroutine STTYP to determine
             type.
          RETURN
        : A label is found :
```

```
                Pass the label to the FORTRAN file.
            ENDCASE
        REPEAT
        Read a card.
        Output FORTRAN if FORTRAN flags are found.
    REPEAT
END
```

SUBROUTINE STTYP

FUNCTION: The subroutine searches the SPARKS keyword table to find the type of statement. The subroutine is called only when a word is found surrounded by blanks which indicates a possible SPARKS statement. If the keyword is not found in the table, the type is assumed to be FORTRAN. The keyword table is initialized in the BLOCK DATA. STTYP includes the following common blocks: INPUT, STTYP2, AND TABLES.

ABSTRACT CODE

```
BEGIN
    Initialize local variables.
        Use binary search of key word table. If
        first letter match is found, check length
        character by character. If length matches
        then assign a value to TYPE and RETURN.
    If a match is not found then TYPE is assigned
        as FORTRAN.
```

RETURN

END

SUBROUTINE SCAN1

FUNCTICN: THe subroutine scans the input card for a requested character that is passed in. It either returns with the character found or with an END condition. SCAN1 includes the following common blocks: INPUT and STTYP2.

ABSTRACT CODE

```
BEGIN
   WHILE the line pointer is less than 72 DO
      Search for the input character.
      If an end condition is found (or a semicolon),
         RETURN with a TRUE end condition.
      If the input character is found, RETURN.
    REPEAT
   RETURN with a true end condition.
END
```

SUBROUTINE SCAN

FUNCTICN: The subroutine searches the input string for a pattern that is passed to it. If the pattern is found, the subroutine returns with a false end condition. If the pattern is not found the end condition is returned true. SCAN includes the following common blocks: INPUT and STTYP2.

ABSTRACT CODE

```
BEGIN
   LOOP
      Match the input pattern , a character at a time.
      If the pattern is found then end condition
         is returned false.
      Else the end condition is returned true.
   REPEAT
END
```

SUBROUTINE SPRNT

FUNCTION: The subroutine prints the processed statements as SPARKS output. A 131 character line is used to output the SPARKS code. SPRNT includes the following common blocks: INPUT, STTYP2, SPTINF, FRTINF, NMNIC, and TYP.

ABSTRACT CODE

```
BEGIN
   Fill the first six columns of output buffer.
   Set beginning of line to be output.
   Set end of line to be output.
   Write line to printer.
   Reset first six columns of output buffer
      to blanks.
END
```

SUBROUTINE FHAND

FUNCTION: The subroutine handles the FORTRAN statements. Fhand insures that column six is proper for the statement being processed, causes any incomplete FORTRAN statements to be completed, and causes the FORTRAN to be output. FHAND includes the following common blocks:INPUT, STTYP2, FRTINF, and TYP.

ABSTRACT CODE

```
BEGIN
   LOOP
      Check for a semicolon or end condition.
      If type is a continuation, copy the first six
         columns from input.
      Complete and statements by calling FCUT.
      RETURN when an end condition is found.
   REPEAT
END
```

SUBROUTINE FCUT

FUNCTION: The subroutine completes FORTRAN statements by putting the label in columns one through five and introducing a C in column one if the line is a comment. If a blank line is found, the routine returns with blank line true. The routine causes FORTRAN to be output by FPRNT. FCUT includes the following common blocks: CENTR, FRTINF, STTYP2, NMNIC, TYP, and FORPRT.

ABSTRACT CODE

```
BEGIN
   LOOP UNTIL the line has been completely scanned.
      If the line pointer is at the end then
         cause the line to be printed (call FPRNT).
      Fill the output buffer with the label, if any.
      If a continuation then fill column six.
      If a comment put a C in column one.
      Check for a blank line, if found then
         RETURN blank line true.
   REPEAT
END
```

SUBROUTINE FPRNT

FUNCTION: The subroutine writes the FORTRAN to the file that will be passed to the FORTRAN compiler for execution. FPRNT includes the following common blocks: INPUT,MNMIC, SPRINF, FRTINF, and FORPRT.

ABSTRACT CODE

```
BEGIN
   Check line pointers, if null output, RETURN.
   Complete filling the output buffer, columns 1
      to 80.
   Check for blank line, if found RETURN.
   Write line to FORTRAN output file.
   Reset line pointers.
   RETURN
```

```
END
```

INTEGER FUNCTION ISNUM

FUNCTION: The function compares the input argument to the digits 0,1,2,...,9. If the input parameter is the alphanumeric representation of a number, the function returns a one. If the comparison fails, a zero is returned. ISNUM has no common blocks.

ABSTRACT CODE

```
BEGIN
  Set up comparison digets 0,1,2,...,9.
  LOOP
     Compare input parameter to each digit in turn.
     If a match is found, then set function to one
        and RETURN.
  REPEAT
  Set function to zero.
  RETURN
END
```

INTEGER FUNCTION CONV

FUNCIION: The Integer Function CONV converts a number into its character representation so that it can be printed. CONV has no common blocks.

ABSTRACT CODE

```
BEGIN
```

Set digits zero to nine in Holerith Fields.

Assign the function the digit in printable form.

RETURN

END

INTEGER FUNCTION GENLAB

FUNCTION: The function generates a label and stores that label in the label array. If there is a user defined label, that label is is also stored in the label array. The function returns with a pointer to the label generated. GENLAB includes the following common blocks: INPUT and CENTRL.

ABSTRACT CODE

BEGIN

If there is not a user defined label then generate a label.

Put the label in the label array.

RETURN with the pointer to the label.

If the number of labels over-drives the label pointer then error.

If there is a user defined label then put that label in the label array.

RETURN with a pointer to the label.

END

SUBROUTINE CHECK

FUNCTICN: The subroutine does the necessary checking :f SPARKS statements. SPARKS keywords which begin SPARKS :onstructions are stacked for checking later when the end :ondition for that construction is enccuntered. The :ubrcutine checks the stack top tc insure the statement :eing checked is used in the correct context. CHECK :ncludes the following common blocks: INPUT, PRTINF, :PRINF, OTPAT, STTYP1, CENTR, INPAT, TABLES, and KEYWD.

ABSTRACT CODE

```
BEGIN
   Check stack, if empty then error.
   CASE
      : SPARKS type is a CCICN or ENDCASE :
         If the stack top is a CASE then RETURN
            else error.
      : SPARKS type is an ELSE or ENDIF :
         If the stack top is an IF then RETURN
            else error.
      : SPARKS type is an UNTIL :
         If the stack top is a LOOP then
            RETURN, else error.
      : SPARKS type is an END :
         If the stack top is a program, subroutine,
            or function then RETURN, else error.
         Unstack until a PRCGRAM, SUBROUTINE, or
            FUNCTION is found, close the block.
```

```
            If the stack goes empty while unstacking and
               statement not found, then error.
         : SPARKS type is a REPEAT, EXIT, or CYCLE :
            If the stack top is an iterative statement
               then RETURN, else error.
      ENDCASE
      RETURN
   END
```

SUBROUTINE ERROR

FUNCTION: The Subroutine outputs sparks error messages, based on input arguements. Error contains the data statements necessary to build the appropriate error messages. ERROR includes the following common blocks: INPUT, STTYP2, and MNMIC.

ABSTRACT CODE

```
   BEGIN
      Write the beginning of the error message.
      Match the input arguements to the error message data
         table.
      Write the remaining part of the error message.
   END
```

SUBROUTINE TYHAND

FUNCTION: The subroutine handles the processing of the EXIT and CYCLE statements. It causes the appropriate

FORTRAN code to be output to implement the intent of the EXIT and CYCLE. TYHAND includes the following common blocks: INPUT, SPRINF, FRTINF,OTPAT, CENTR, and STTYP2.

ABSTRACT CODE

```
BEGIN
   Check for a label, if present then
      error, delete.
   Convert the EXIT or CYCLE to FORTRAN by
      generating a GO TO the appropriate
      LOOP label.
   RETURN
END
```

3.2 Program Variables.

The global variables of the program are listed below along with the routines in which they appear. The arrays are listed after the variables. Local loop counters and temporary 'hold value' variables are not included.

Global Variables

Variable	Using Routines
BLANK	MAIN, BLOCK DATA, DETNX, SCAN, SPRNT, FHAND, FOUT.
BLNKLN (LOGICAL)	FPRNT, FOUT.
BKWRD	MAIN, BLOCK DATA.
BLKNO	MAIN, SPRNT.
CCASE	MAIN, BLOCK DATA.
CEE	MAIN, BLOCK DATA, SPRNT, FOUT.
CCLN	MAIN, BLOCK DATA.
COLON	MAIN, BLOCK DATA, CHECK.
CCMT	MAIN, BLOCK DATA, DETNX, SPRNT, FOUT.
CONT	MAIN, BLOCK DATA, DETNX, SPRNT,

	FHAND.
CRDR	BLOCK DATA, DETNX.
CURFLG	MAIN, BLOCK DATA, TYHAND.
DBLSL	MAIN, BLOCK DATA, SITYP.
ECOND (LOGICAL)	MAIN, DETNX, SCAN1, SCAN, FHAND, CHECK.
EELSE	MAIN, BLOCK DATA, CHECK.
EEND	MAIN, BLOCK DATA, CHECK.
EENDIF	MAIN, BLOCK DATA, CHECK.
EEXIT	MAIN, BLOCK DATA, CHECK.
ENCASS	MAIN, BLOCK DATA, CHECK.
EOJ	MAIN, BLOCK DATA.
FFILE	BLOCK DATA, DETNX, FPRNT.
FFOR	MAIN, BLOCK DATA, CHECK.
FMAX	MAIN, FPRNT, FOUT.
FOTRN	MAIN, BLOCK DATA, DETNX, STTYP.
FPTR	MAIN, FPRNT, FOUT.
FRWRD	MAIN, BLOCK DATA.
FSTRT	MAIN, FPRNT, CHECK.
IIF	MAIN, BLOCK DATA, CHECK.
INDFLG	MAIN, BLOCK DATA, FOUT, FPRNT.
IMAX	MAIN, DETNX, SCAN1, SCAN.
LBFLG (LOGICAL)	MAIN, DETNX, SPRNT, FHAND, GENLAB.
LBMAX	MAIN, CHECK, GENLAB.
LBPTR	MAIN, CHECK, GENLAB.
NCYCLE	MAIN, BLOCK DATA, CHECK.
NESTNO	MAIN, SPRNT.
NEUTR	MAIN, BLOCK DATA.
NUMB	MAIN, BLOCK DATA, GENLAB.
OLD	MAIN, BLOCK DATA.
PRTR	MAIN, BLOCK DATA, DETNX.
PULOUT	MAIN, BLOCK DATA.
RINLO	MAIN, BLOCK DATA.
REPT	MAIN, BLOCK DATA, CHECK.
SEMI	BLOCK DATA, DETNX, SCAN1, FHAND.
SSTRT	SPRNT, CHECK.
STBEG	MAIN, SPRNT.
STEND	MAIN, SPRNT.
STMAX	MAIN, BLOCK DATA.
STMNO	MAIN, SPRNT, FPRNT.
SUBFN	MAIN, BLOCK DATA, CHECK.
TOBEG	MAIN, DETNX, STTYP, SCAN, FHAND.
TCEND	MAIN, DETNX, STTYP, SCAN, FHAND, CHECK.
TCP	MAIN, STTYP, CHECK.
TPITR	MAIN, CHECK.
TYPE	MAIN, DETNX, STTYP, SPRNT, FHAND, FOUT.
UNTL	MAIN, BLOCK DATA, CHECK.
WHLE	MAIN, BLOCK DATA, CHECK.

Local Variables

MAIN ROUTINE	BAD (LOGICAL), CCYCLE, CURFLG, ECOND1 (LOGICAL) EX2A, EX2B, INCR, INDFLG, ITEMP, LAB, LAB2, LAB3, LAB4, CNE, SMAX.
SUBROUTINE DETNX	EFF.
SUBROUTINE STTYP	BOTTOM, LENGTH, MID, PNT.
SUBROUTINE SCAN1	CHAR.
SUBROUTINE SCAN	ISET, LEN, PAT.
SUBROUTINE SPRNT	FIL, LEN1, LEN2, OVFL (LOGICAL), SEND.
SUBROUTINE FOUT	NINE, PAT, STRT.
SUBROUTINE CHECK	STYPE.
SUBROUTINE ERROR	LEN, LEN1, LEN2.

GLOBAL ARRAYS

BLNKS(131)	MAIN, BLOCK DATA, SPRNT, FPRNT.
BYE(2)	MAIN, BLOCK DATA.
CASS(4)	MAIN, BLOCK DATA, CHECK.
CCOLON(5)	MAIN, BLOCK DATA, CHECK.
CCYCLE(5)	MAIN, BLOCK DATA, CHECK.
CGOTO(9)	MAIN, BLOCK DATA.
CONTI(8)	MAIN, BLOCK DATA.
DBSLH(2)	MAIN, BLOCK DATA.
DO(2)	MAIN, BLOCK DATA.
EEEND(3)	BLOCK DATA, CHECK.
EEEXT(4)	BLOCK DATA, CHECK.
ELZ(4)	MAIN, BLOCK DATA.
ENDCASS(7)	MAIN, BLOCK DATA, CHECK.
ENIF(5)	MAIN, BLOCK DATA, CHECK.
EQL(1)	MAIN, BLOCK DATA.
FBUF(80)	MAIN, DETNX, FHAND, FPRNT, FOUT.
GOTO(6)	MAIN, BLOCK DATA.
IIIF(2)	BLOCK DATA, CHECK.
IFNOT(9)	MAIN, BLOCK DATA.
ITERAT(9)	BLOCK DATA, CHECK.
LABEL(200)	MAIN, BLOCK DATA, DETNX, FOUT, CHECK, GENLAB.
LEO(5)	MAIN, BLOCK DATA.
LLOOP(4)	BLOCK DATA, CHECK.
MINUS(1)	MAIN, BLOCK DATA.
ONE(1)	MAIN, BLOCK DATA.
PLU(1)	MAIN, BLOCK DATA.
RREPT(6)	BLOCK DATA, STTYP.
RESRD(200)	BLOCK DATA, STTYP.
RSTAR(2)	MAIN, BLOCK DATA.
SPIN(80)	MAIN, BLOCK DATA, DETNX,

	STTYP, SCAN1, SCAN, SPRNT, FHAND, FPRNT, GENLAB, ERRCH.
STACK(200)	MAIN, CHECK.
STFRM(6)	BLOCK DATA, CHECK.
THENN(4)	MAIN, BLOCK DATA.
TOO(2)	MAIN, BLOCK DATA.
TRES(23)	BLOCK DATA, STTYP.
UUNTL(5)	MAIN, BLOCK DATA, CHECK.

3.3 COMMON Blocks.

The COMMON blocks and the variables they include are listed below.

/INPUT/ SRIN(80), TOBEG, TOEND, STBEG, STEND, IMAX, LBFLG

/SPRINF/ SSTRT, INCR, STMNO, BLKNO, NESTNO, INDFLG, CURFLG, SMAX

/FRTINF/ FBUF(80), FPTR, FMAX, FSTRT, BINKS(131)

/CENTRL/ STACK(200), LABEL(200), TOP, LBMAX, LBPTR, STFRM(6), TPITR, NEUTR, BKWRD, FRWRD, NEW, CLD, PULOUT, STMAX, NUMB

/OTPAT/ CGOTO(9), CONTI(8), EQL(1), GOTO(6), IFNOT(9), IEO(5), MINUS(1), CNE(1), PLU(1), RSTAR(2)

/INPAT/ CCLN(1), ELZ(4), DBSLH(2), DOO(2), BYE(2), THENN(4), TOO(2)

/STTYP1/ CCASE, CCLON, DBLSL, EELSE, EEND, ECJ, EEXIT, FFOR, IIF, LLOOP, RREPT, UNTL, WHLE, SUBFN, BINLO, NCYCLE, ENCAS, EENDIF

/STTYP2/ COMT, CONT, FOTRN, BLANK, SEMI

/NMNIC/ CEE, PRTR, FFILE, CRDR

/KEYWD/ EEEND(3), EEEXT(4), CCYCLE(5), UUNTL(5), CASS(4), IIIF(2), ITERAR(9), ENCASS(7), ENIF(5), REPT(6), CCOLON(5)

CHAPTER 3

/TABLES/ TRES(23), RESRD(200)

/TYP/ TYPE

/FORTPRT/ BLNKLN

CHAPTER 4

SPARKS USERS MANUAL

4.0 INTRODUCTION.

FORTRAN users, SPARKS is meant for you! As a FORTRAN user, you are taking advantage of a language that has been in use since 1957. You have, no doubt, come to rely on FORTRAN's established portability, highly-developed compilers, and wide-spread availability. You also know you are not alone, as there is an established large labor force familiar with FORTRAN. However, as you thumb through your FORTRAN program, haven't you wondered how nice it would be if you didn't have to keep jumping about following GO TO's? And, what of all the talk going around now about program structure and modularity? Don't you even feel a little bit guilty? You don't want to give up all of the advantages of FORTRAN just to elimate GO TO's and get a little structure? Now you can guess why SPARKS is for you.

What is SPARKS? It is essentially a preprocessor for FORTRAN which inexpensively augments the language with improved syntactical constructs and other useful features. A FORTRAN preprocessor is a program which translates statements written in a language X into FORTRAN. In this case language X is SPARKS. SPARKS is classified as a preprocessor instead of a compiler because the source and the target language language have many statements in common. A translator such as SPARKS has many advantages. Most

importantly, it preserves a close connection with FORTRAN. Also many installations do not have a nicely structured language available and the translator provides a simple means for supplementing the exsisting FORTRAN capability.

4.1 SAMPLE PROGRAM.

Is a little convincing needed for you to give up the GO TO? Consider writting a program which searches for an integer X in a sorted array of integers A(N), N less than or equal to 100. The output is the integer J which is either zero if X is not found or A(J)=X, 0<J<N+1. The method used is the well known binary search algorithm. A reasonable attempt to write the program in FORTRAN follows:

```
      SUBROUTINE BINS(A,N,X,J)
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(100)
      BOT=1
      TOP=N
      J=0
  100 IF(BOT.GT.TOP) RETURN
      MID=(BOT+TOP)/2
      IF(X.GE.A(MID)) GO TO 101
      TOP=MID-1
      GO TO 100
  101 IF(X.EQ.A(MID)) GO TO 102
      BOT=MID+1
      GO TO 100
```

```
      102 J=MID
          RETURN
          END
```

Now, a look at this algorithm written in SPARKS.

```
          SUBROUTINE BINS(A,N,X,J)
          IMPLICIT INTEGER (A-Z)
          DIMENSION A(100)
          BOT=1 ; TOP=N ; J=0
          WHILE BOT.LE.TOP DO
             MID=(BOT+TOP)/2
             CASE
                : X.LT.A(MID) : TOP=MID-1
                : X.GT.A(MID) : BOT=MID+1
                : ELSE        : J=MID ; RETURN
             ENDCASE
          REPEAT
          RETURN
          END
```

The difference between the two algorithms is not dramatic, but it is significant. The WHILE and CASE statements allow the algorithm to be described in a more natural way. The program can be read from top to bottom without jumping up and down the page. When such improvements are consistently adopted in a large software project, the resulting code is bound to be easier to comprehend. Like to know more about SPARKS? Read on!

4.2 Language Definitions.

First the SPARKS language must be precisely defined. A distinction must be made between strictly FORTRAN statements and SPARKS statements. The latter are recognized by certain keywords and/or delimiters. Statements not containing SPARKS keywords are regarded as strictly FORTRAN and passed without alteration. SPARKS statements cause the translator to produce ANSI FORTRAN statements which accomplish the equivalent computation. Thus, SPARKS is compatible with FORTRAN and is, for all intents and purposes, a FORTRAN program. Hence, the local compiler ultimately defines the semantics of all SPARKS statements.

The method of inputing a SPARKS program is identical to FORTRAN; columns 7 to 72 are used for statements. Labels, except for format statements, are not normally used in SPARKS. However, they are input as are FORTRAN labels, in columns 1 to 5.

The SPARKS preprocessor outputs two temporary files. One file is the processed SPARKS code which is nicely indented. The second file is the FORTRAN translation of the SPARKS program. The FORTRAN is passed to the compiler for further translation. The SPARKS output is automatically provided to the user. The FORTRAN will normally be echoed by the local compiler depending on the installation.

SPARKS has a total of 20 keywords and special symbols. These words and symbols are listed below.

BY	CASE	CYCLE	DO	ELSE
ENDCASE	ENDIF	EOJ	EXIT	FOR
IF	LOOP	REPEAT	UNTIL	WHILE
TO	THEN	:	;	//

* * * * * * N O T E * * * * * *

The above keywords/symbols MUST be surrounded by blanks to be recognized by the SPARKS preprocessor. Otherwise, they will be passed as FORTRAN variables. The keywords are not reserved in that they can be used as variables when not surrounded by blanks. However, to avoid confusion and possible errors, use of the keywords as variables is not recommended.

* * * * * *

SPARKS statements can now be defined by giving their FORTRAN equivalents. The term statement, used below, and indicated by S, S1, S2, etc is meant to include any arbitrary number of both SPARKS and FORTRAN statements. There are six basic SPARKS constructions, two of which improve the testing of cases and four which improve the description of looping.

4.3 SPARKS IF Statement.

SPARKS

IF cond THEN

```
        S1
    ELSE
        S2
    ENDIF
```

FORTRAN TRANSLATION

```
    IF(.NOT.(cond)) GO TO 99999
        S1
        GO TO 99998
99999 CONTINUE
        S2
99998 CONTINUE
```

The ELSE clause is optional, but the terminating ENDIF is mandatory. An IF statement without the ELSE clause is included below.

SPARKS

```
    IF cond THEN
        S1
        GO TO 200
    ENDIF
    S3
200 CONTINUE
```

FORTRAN TRANSLATION

```
    IF(.NOT.(cond)) GO TO 99999
        S1
        GO TO 200
99999 CONTINUE
```

```
        S3

    200 CONTINUE
```

The GO TC in the SPARKS is included for those FORTRAN programmers who refuse to let go. It does serve to illustrate; however, that any legal FORTRAN statement can be mixed with SPARKS code. The 'cond' of the above IF statements must be a legal FORTRAN conditional.

4.4 SPARKS CASE Statement.

The second construction dealing with testing is the SPARKS CASE statement. This construct is most useful in consolidating nested test statements or strings of related conditional tests.

SPARKS

```
CASE
   : cond1 : S1
   : cond2 : S2
   : condn : Sn
   : ELSE  : Sn+1
ENDCASE
```

FORTRAN TRANSLATION

```
      IF(.NOT.(cond1)) GO TC 99999
        S1
        GO TO 99998
99999 IF(.NOT.(cond2)) GO TC 99997
        S2
```

```
                GO TO 99998
99997 IF(.NOT.(condn)) GO TC 99996
                Sn
                GO TO 99998
99996 CONTINUE
                Sn+1
99998 CONTINUE
```

The cond1, cond2, etc, are any arbitrary number of legal FORTRAN conditionals. THe EISE case designates that the statements representated by Sn+1 will always be executed if all previous conditicns are false. The ELSE clause is cptional. The ENICASE is mandatory.

4.5 SPARKS LOOP Statements.

The four LOOP structures of SPARKS are the WHILE, LOOP/UNTIL, LOOP/REPEAT, and the FOR loop.

* * * * * * N O T E * * * * * *

The prcgrammer must provide fcr the incrementation or change in condition for proper termination of the WHILE loop and the LCOP/UNTIL. Also, the ICCP/REPEAT is a loop forever ccnstruction. The programmer must prcvide the means to get cut of the LOOP/REPEAT.

* * * * * *

The construction of the WHILE loop is shown below.

SPARKS

```
WHILE cond DO
   S1
REPEAT
```

FORTRAN TRANSLATION

```
99999 IF(.NOT.(cond)) GO TO 99998
        S1
        GO TO 99999
99998 CONTINUE
```

The S1 indicates an arbitrary number of statements and must include a means for meeting the cond for termination. The cond must be a legal FORTRAN conditional. The REPEAT is mandatory.

The LOOP/UNTIL is similar to the WHILE loop and is used when the test to exit is desired at the bottom of the loop.

SPARKS

```
LOOP
   S1
UNTIL cond REPEAT
```

FORTRAN TRANSLATION

```
99999 CONTINUE
        S1
        IF(.NOT.(cond)) GO TO 99999
```

The S1 and cond imply the same meaning as in the WHILE loop above. Again, inclusion of the REPEAT is mandatory.

The LOOP/REPEAT is convenient to provide a loop

structure where a exit condition can be encountered at several locations within the loop body. The programmer should be aware that this loop structure will not terminate unless some exit or stop condition is included in the loop body. The REPEAT is the loop delimiter and is mandatory.

SPARKS

```
LOOP
   S1
REPEAT
```

FORTRAN TRANSLATION

```
99999 CONTINUE
   S1
   GO TO 99999
```

The SPARKS FOR loop is similar in its action to the FORTRAN DO loop but is much more powerful. It should be noted; however, that the translation of the FOR loop in no way resembles the DO loop.

SPARKS

```
FOR vble=exp1 TO exp2 BY exp3 DO
   S1
REPEAT
```

FORTRAN TRANSLATION

```
vble=exp1
GO TO 99999
99998 vble=vble+exp3
```

```
99999 IF(.NOT.(vble-(exp2))*(exp3).LE.0)
     1GO TO 99997
        S1
     GO TO 99998
99997 CONTINUE
```

The '1GO TO 99997' is a continuation line. The three expressions (exp1, exp2, exp3) are allowed to be arbitrary FORTRAN arithmetic expressions. Similarily, vble may be any FORTRAN variable. Negative incrementation of the FOR loop is acceptable as shown in the example below.

```
FOR I=10 TO 2 BY -2 DO
   J=I+10
REPEAT
```

The phrase BY exp3 is optional and if not included the incrementation defaults to +1. Since exp2 and exp3 are re-evaluated each time through the loop, care must be taken not to modify the value of these expressions within the loop body. As with all SPARKS loops, the REPEAT is mandatory.

4.6 SPARKS EXIT and CYCLE Statements.

The EXIT statement is used in SPARKS to leave a loop structure prematurely. The EXIT causes control to be transferred to the first statement outside of the innermost loop statement that contains it. The EXIT statement can be used with any of the four SPARKS loop structures. An example of its use and the FORTRAN translation is shown

below.

SPARKS

```
      LOOP
         S1
         IF cond THEN EXIT
         ENDIF
         S2
      REPEAT
```

FORTRAN TRANSLATION

```
99999 CONTINUE
         S1
         IF(.NOT.(cond)) GO TO 99998
            GO TO 99997
99998 CONTINUE
         S2
         GO TO 99999
99997 CONTINUE
```

The CYCLE statement may also be used within any SPARKS loop construction. The CYCLE causes control to be transferred directly to the end of the innermost loop that contains it. Normal loop iteration is then continued. The following is example of both the EXIT and CYCLE statements. Note that the CYCLE does not transfer control outside of the loop that contains it.

SPARKS

```
LCOP
   S1
   CASE
      : cond1 : EXIT
      : cond2 : CYCLE
   ENDCASE
   S2
REPEAT
FORTRAN TRANSLATION
99999 CONTINUE
         S1
         IF(.NOT.(cond1)) GO TO 99996
            GO TO 99994
            GO TO 99995
99996 IF(.NOT.(cond2)) GO TO 99995
            GO TO 99998
99995 CONTINUE
      S2
99998 GO TO 99999
99994 CONTINUE
```

The sharp FORTRAN programmer will immediately notice that this translation produces unreachable GO TO statements. The EXIT, CYCLE, and RETURN statements when used in a CASE or an ELSE of the IF/THEN/ELSE statement nested in a loop will generate unreachable GO TO statements. Many FORTRAN compilers will give non-fatal compiler warnings concerning

the unreachable statements; however, the warnings should not effect the execution of the program.

4.7 Labelling in SPARKS.

The use of five digit labels in the examples has probably been noticed. SPARKS generates labels beginning with 99999 and decreases by one for each new label required. The numbering begins again for each new subroutine. Programmers who use labelled statements should avoid using large numbers to prevent label conflicts when translation occurs. User defined labels are passed without modification to the FORTRAN output.

4.8 The SPARKS EOJ Statement.

The SPARKS EOJ signals the translator that is the last card of the program. It must appear somewhere in columns 7 to 72 and be surrounded by blanks. The EOJ is used one per program.

4.9 SPARKS Comments.

SPARKS comments are delimited by the double slash. The following is an example of a proper SPARKS comment.

```
// THIS IS A SPARKS COMMENT //
```

All characters within the double slashes will be ignored. The comments are restricted to one line in columns 7 to 72. The double slashes must be surrounded by blanks. The double

slash is taken as a single symbol and may not contain blanks between the two single slash marks. SPARKS comments appearing on the same line as other statements must be separated from those statements with a semicolon (see section 4.9). SPARKS comments will appear in the FORTRAN output as follows:

```
c       // THIS IS A SPARKS COMMENT
```

The translator will automatically put the C in column one. Note that the trailing double slash is not passed to the FORTRAN output. However, failure to include the trailing double slash in the SPARKS will cause the translator to output an error message. FORTRAN comments are allowed and are output without modification.

4.10 General Information and Other Features

* * * * * * N O T E * * * * * *

The indention feature in SPARKS will sometimes cause a long line to exceed column 72. Normally this is not a problem as the translator automatically introduces a proper continuation card. However, if the long line contains a Hollerith field and the break occurs in that field, a fatal FORTRAN error may be introduced. The programmer should use caution when coding long statements such as FORMAT and DATA statements containing Hollerith fields.

* * * * * *

If such statements cannot be avoided, the preprocessor operating mode can be altered to pass the statements without processing. A F in column one signals the preprocessor that subsequent lines contain only FORTRAN statements. Those statements are then passed without any checking or indentation. The next line with a F in column one returns the preprocessor to its normal mode. The cards containing the signal F must not contain other statements. The SPARKS output will have the lines surrounded by the F signal blocked out by the message:

```
++++++++++BEGINNING OF FORTRAN
Arbitrary number of FORTRAN Statements
++++++++++END OF FORTRAN
```

The FORTRAN output will not contain any messages and it will not be indented.

The semicolon is used in SPARKS as a statement delimiter. Using the semicolon, more than one statement can be put on a single line. For example, the following line beginning in column 7 is a legal SPARKS input.

```
C=D+E ; A=B+C ; X=A ; // SPARKS COMMENT //
```

The preprocessor will translate the line into four lines of FORTRAN as follows:

```
      C=D+E
      A=B+C
      X=A
C     // SPARKS COMMENT
```

As a SPARKS key symbol, the semicolon must be surrounded by blanks.

4.11 Conclusions.

This user's guide is an adaptation of the user's guide developed by Dr. Ellis Horowitz and the SPARKS User Group, University of Southern California. Dr. Horowitz designed the SPARKS language originally for describing algorithms in the book FUNDAMENTALS OF DATA STRUCTURES by Ellis Horowitz and Sartaj Sahni, published by Computer Science Press, Woodland Hills, California. The original SPARKS preprocessor was developed from this prototype language.

This user's guide is an accurate discription of the language features as currently supported by the revised version of the translator. The new version of the translator was rewritten in SPARKS by Richard M. Stroud and Dr. Myron A. Calhoun, Kansas State University, Manhattan, Kansas.

APPENDIX A

SPARKS PREPROCESSOR

WRITTEN IN

SPARKS

```
NEST  BLK  STNO                                  THIS IS A SPARKS PROGRAM

                          SPARKS TRANSLATION BY KSU SPARKS PREPROCESSOR AS
                          MODIFIED BY R. STROUD, M. CALHOUN, AND J. MARTIN.

   0    0     0          // MAIN PROGRAM //
   0    0     1          //              //
   0    0     2          IMPLICIT INTEGER(A-Z)
   0    0     3          LOGICAL LBFLG,ECONC,ECONC1,BAD
   0    0     4          COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
   0    0     5          COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX
   0    0     6          COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)
   0    0     7          COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STERM(6),
   0    0     8        1 TPITR,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB
   0    0     9          COMMON/OTPAT/OGOTO(9),OCNTI(8),EQL(1),GOTO(6),IFNOT(9),LEO(5),
   0    0    10        1 MINUS(1),ONE(1),PLU(1),RSTAR(2)
   0    0    11          COMMON/INPAT/OOLN(1),ELZ(4),OBSLH(2),DOO(2),BYE(2),THENN(4),TOO(2)
   0    0    12          COMMON/STTYP1/OCASE,OOLON,DBLSL,EELSE,EEND,EOJ,EEXIT,FFOR,IIF,
   0    0    13        1 LLOOP,RREPT,UNTL,WHLE,SUBFN,RINLO,NOYCLE,ENDCAS,EENDIF
   0    0    14          COMMON/STTYP2/OONT,OONT,FOTRN,BLANK,SEMI
   0    0    15          COMMON/NMNIC/OEE,PRTR,FFILE,ORDR
   0    0    16          COMMON/KEYWD/EEEND(3),EEEXT(4),OOYCLE(5),UUNTL(5),OASS(4),IIIF(2),
   0    0    17        1 ITERAT(9),ENOASS(7),ENIF(5),REPT(6),OOOLON(5)
   0    0    18          COMMON/TABLES/TRES(23),RESRD(200)
   0    0    19          COMMON/TYP/TYPE
   0    0    20          //              //
   0    0    21          // WRITE TITLES ETC. //
   0    0    22          WRITE(PRTR,3)(SPIN(I),I=1,80)
   0    0    23        3 FORMAT(4X,4HNEST,2X,3HBLK,2X,4HSTNO,4X,30X,80A1)
   0    0    24          WRITE(PRTR,4)
   0    0    25        4 FORMAT(/32X,'SPARKS TRANSLATION BY KSU SPARKS PREPROCESSOR ',
   0    0    26        1 'AS'/32X,'MODIFIED BY R. STROUD, M. CALHOUN, AND J. MARTIN.'/)
   0    0    27          //              //
   0    0    28          // SET UP OPTIONS //
   0    0    29          //              //
   0    0    30          IMAX=72
   0    0    31          SMAX=100
```

```
0    0    32              FMAX=72
0    0    33              INCR=3
0    0    34              FSTRT=7+INCR
0    0    35              SSTRT=7+INCR
0    0    36              // EXTRACT OPTIONS GIVEN ON THE CONTROL CARD AND CHANGE ABOVE //
0    0    37              // DEFAULTS ACCORDINGLY---THIS COULD USE DETNX //
0    0    38              // INITIALISE THE PROGRAM RELATED VARIABLES. //
0    0    39              STACK(1)=SUBFN
0    0    40              TOP=1
0    0    41              TPITR=0
0    0    42              LPPTR=6
0    0    43              TOEND=IMAX+1
0    0    44              BLKNO=0
0    0    45              STMNO=0
0    0    46              NESTNO=0
0    0    47              // START MAIN LOOP //
0    0    48              LOOP
1    1    49                 FPTR=FSTRT
1    1    50                 CALL DETNX(TYPE)
1    1    51                 // PROCESS STATEMENTS ACCORDING TO TYPE //
1    1    52                 // 1. TRANSLATE AND PRINT FORTRAN. //
1    1    53                 // 2. SET INDENTATION FLAG---INDFLG. //
1    1    54                 // 3. SET CURFLG NEW-OLD--CURRENT STATEMENT TO BE PRINTED IN //
1    1    55                 // OLD OR NEW INDENTATION. //
1    1    56                 //              //
1    1    57                 IF TYPE.EQ.RINLC THEN
2    2    58                    A=TOBEG
2    2    59                    B=TOEND
2    2    60                    CALL DETNX(TYPE)
2    2    61                    // IF THE SECOND TOKEN IS NOT FUNCTION THEN PASS AS FORTRAN. //
2    2    62                    IF TYPE.NE.SUBFN THEN
3    3    63                       TYPE=FOTRN
3    3    64                    ENDIF
2    4    65                    TOEND=B
2    4    66                    TOBEG=A
2    4    67                 ENDIF
1    5    68                 STBEG=TOBEG
1    5    69                 // THE MAIN CASE STATEMENT //
```

```
1     5     70                CASE
2     6     71                : TYPE.EQ.FCTRN :
2     7     72                   CALL FHAND
2     7     73                   STEND=TCEND
2     7     74                   CURFLG=CLD
2     7     75                   INDFLG=NEUTR
2     7     76                : TYPE.EQ.DBLSL :
2     8     77                   // LOCK FOR A DOUBLE SLASH--SCAN IS NOT USED SINCE IT STOPS //
2     8     78                   // ON A SEMICOLON. //
2     8     79                   LOOP
3     9     80                      IF TOEND.GE.IMAX-1 THEN
4    10     81                         CALL ERROR(2,DBSLH,2,1,0,0)
4    10     82                         // ERROR--MISSING DOUBLE SLASH SUPPLIED AT END OF STATEMENT. //
4    10     83                         EXIT
4    10     84                      ENDIF
3    11     85                      IF SPIN(TCEND+1).EQ.DBSLH(1).AND.SPIN(TOEND+2).EQ.DBSLH(1) THEN
4    12     86                         EXIT
4    12     87                      ENDIF
3    13     88                      TCEND=TCEND+1
3    13     89                   REPEAT
2    14     90                   FBUF(1)=CEE
2    14     91                   // THE TYPE IS CHANGED SO THAT FOUT CAN CHECK IT AND PUT A C IN //
2    14     92                   // COLUMN ONE FOR LONG COMMENTS. //
2    14     93                   TYPE=COMT
2    14     94                   CALL FOUT(SPIN,TOBEG,TCEND)
2    14     95                   CALL FPRNT
2    14     96                   INDFLG=NEUTR
2    14     97                   CURFLG=CLD
2    14     98                   TCEND=TCEND+2
2    14     99                   STEND=TCEND
2    14    100                : TYPE.EQ.IIF :
2    15    101                   LAB=GENLAB(1)
2    15    102                   CALL SCAN(THENN,4,ECOND)
2    15    103                   IF ECOND THEN
3    16    104                      // ABSENSE OF THEN ASSUMED TO INDICATE FORTRAN IF. //
3    16    105                      TOBEG=TOBEG-2
3    16    106                      CALL FHAND
3    16    107                   ELSE
```

```
3   17   108                  // OUTPUT: IF(.NOT.COND) GO TO LAB //
3   17   109                  CALL FOUT(IFNOT,1,9)
3   17   110                  CALL FOUT(SPIN,TOBEG,TOEND)
3   17   111                  CALL FOUT(CGOTO,1,9)
3   17   112                  CALL FOUT(LABEL,LAB,LAB+4)
3   17   113                  CALL FPRNT
3   17   114                  STACK(TOP+1)=LAB
3   17   115                  STACK(TOP+2)=IIF
3   17   116                  TOP=TOP+2
3   17   117                  IF TOP.GT.STMAX THEN
4   18   118                     CALL ERROR(6,0,0,5,0,0)
4   18   119                  ENDIF
3   19   120                  //  STACK OVERFLOWED, PROGRAM TERMINATED. //
3   19   121                  TOEND=TOEND+4
3   19   122                  IF FSTRT+INCR.LT.IMAX THEN
4   20   123                     FSTRT=FSTRT+INCR
4   20   124                  ENDIF
3   21   125               ENDIF
2   22   126               STEND=TOEND
2   22   127               CURFLG=OLD
2   22   128               INDFLG=FRWRD
2   22   129            : TYPE.EQ.EENDIF :
2   23   130               CALL CHECK(EENDIF,BAD)
2   23   131               IF BAD THEN
3   24   132                  CYCLE
3   24   133               ENDIF
2   25   134               FSTRT=FSTRT-INCR
2   25   135               FPTR=FSTRT
2   25   136               IF LBFLG THEN
3   26   137                  // LABELLED STATEMENT--GENERATE 'LAB CONTINUE' //
3   26   138                  CALL FOUT(CONTI,1,8)
3   26   139                  CALL FPRNT
3   26   140               ENDIF
2   27   141               LAB=STACK(TOP-1)
2   27   142               TOP=TOP-2
2   27   143               LBPTR=LBPTR-5
2   27   144               FOR I=1 TO 6 DO
3   28   145                  L=LAB+I-1
```

```
3   28   146                        FBUF(I)=LABEL(L)
3   28   147                     REPEAT
2   29   148                     FBUF(6)=BLANK
2   29   149                     CALL FOUT(CONTI,1,8)
2   29   150                     CALL FPRNT
2   29   151                     //   GENERATE A LAB CONTINUE.   //
2   29   152                     STEND=TCEND
2   29   153                     INDFLG=BKWRD
2   29   154                     CURFLG=NEW
2   29   155                  : TYPE.EQ.EELSE :
2   30   156                     CALL CHECK(EELSE,BAD)
2   30   157                     IF BAD THEN
3   31   158                        CYCLE
3   31   159                     ENDIF
2   32   160                     FSTRT=FSTRT-INCR
2   32   161                     IF LBFLG THEN
3   33   162                        CALL ERROR(1,ELZ,4,2,0,0)
3   33   163                        // ERROR--LABELLING THE ELSE HAMPERS PROGRAM CLEANLINESS-- //
3   33   164                        // DELETED FROM THE FORTRAN OUTPUT. //
3   33   165                     ENDIF
2   34   166                     LAB2=GENLAB(1)
2   34   167                     LAB1=STACK(TOP-1)
2   34   168                     FOR I=1 TO 6 DO
3   35   169                        FBUF(I)=BLANK
3   35   170                     REPEAT
2   36   171                     CALL FOUT(GOTO,1,6)
2   36   172                     CALL FOUT(LABEL,LAB2,LAB2+4)
2   36   173                     CALL FPRNT
2   36   174                     // GO TO LAB2 //
2   36   175                     FOR I=1 TO 5 DO
3   37   176                        L=LAB1+I-1
3   37   177                        FBUF(I)=LABEL(L)
3   37   178                     REPEAT
2   38   179                     FBUF(6)=BLANK
2   38   180                     CALL FOUT(CONTI,1,8)
2   38   181                     CALL FPRNT
2   38   182                     // LAB1 CONTINUE //
2   38   183                     STACK(TOP-1)=LAB2
```

```
2   38   184                  STEND=TCEND
2   38   185                  INDFLG=NEUTR
2   38   186                  CURFLG=PULCUT
2   38   187                  IF FSTRT+INCR.LT.IMAX THEN
3   39   188                     FSTRT=FSTRT+INCR
3   39   189                  ENDIF
2   40   190               : TYPE.EQ.FFOR :
2   41   191                  IF LBFLG THEN
3   42   192                     //  LAB WILL HAVE IDENTIFYING LABEL IF ANY. //
3   42   193                     LAB=GENLAB(0)
3   42   194                     CALL FOUT(CONTI,1,8)
3   42   195                     CALL FPRNT
3   42   196                  ENDIF
2   43   197                  TOBEG=TCEND+1
2   43   198                  CALL SCAN1(EQL,ECOND)
2   43   199                  IF ECOND THEN
3   44   200                     CALL ERROR(2,EQL,1,3,0,0)
3   44   201                     //  ERROR--MISSING EQUAL-STATEMENT SKIPPED.  //
3   44   202                     CYCLE
3   44   203                  ENDIF
2   45   204                  VBLEA=TCBEG
2   45   205                  VBLEB=TCEND
2   45   206                  TCEND=TCEND+1
2   45   207                  CALL SCAN(TOC,2,ECOND)
2   45   208                  IF ECOND THEN
3   46   209                     CALL ERROR(2,TOC,2,3,0,0)
3   46   210                     // ERROR--MISSING 'TO' - STATEMENT SKIPPED. //
3   46   211                     TCEND=TCEND+1
3   46   212                     CYCLE
3   46   213                  ENDIF
2   47   214                  TCEND=TCEND+2
2   47   215                  CALL FOUT(SPIN,VBLEA,TCEND-2)
2   47   216                  CALL FPRNT
2   47   217                  //  VARIABLE=EXPRESSION1.  //
2   47   218                  ITMP=TCEND
2   47   219                  CALL SCAN(BYE,2,ECOND1)
2   47   220                  EX2A=TOBEG
2   47   221                  EX2B=TCEND
```

```
2   47   222                TCEND=TCEND+2
2   47   223                IF ECOND1 THEN
3   48   224                   TCEND=ITMP
3   48   225                ENDIF
2   49   226                CALL SCAN(CCC,2,ECOND)
2   49   227                IF ECOND1 THEN
3   50   228                   EX2B=TCEND
3   50   229                ENDIF
2   51   230                IF ECOND THEN
3   52   231                   CALL ERROR(2,DOC,2,1,0,0)
3   52   232                   // ERROR--MISSING 'DO'- SUPPLIED AT END OF STATEMENT. //
3   52   233                   TCEND=TCEND+1
3   52   234                ELSE
3   53   235                   TCEND=TCEND+2
3   53   236                ENDIF
2   54   237                LAB1=GENLAB(1)
2   54   238                LAB2= GENLAB(1)
2   54   239                LAB3=GENLAB(1)
2   54   240                LAB4=GENLAB(1)
2   54   241                // LAB IS THE IDENTIFYING LABEL, LAB4 IS THE TARGET OF EXIT, //
2   54   242                // AND LAB3 IS THE TARGET OF CYCLES. //
2   54   243                STACK(TCP+1)=LAB2
2   54   244                STACK(TCP+3)=LAB
2   54   245                STACK(TOP+4)=LAB3
2   54   246                STACK(TCP+5)=LAB4
2   54   247                STACK(TOP+6)=FFOR
2   54   248                STACK(TCP+2)=TPITR
2   54   249                TOP=TCP+6
2   54   250                IF TOP.GT.STMAX THEN
3   55   251                   CALL ERROR(6,0,0,5,0,0)
3   55   252                ENDIF
2   56   253                //  STACK OVERFLOWED----PROGRAM TERMINATED.  //
2   56   254                TPITR=TCP
2   56   255                FOR I=1 TO 6 DO
3   57   256                   FBUF(I)=BLANK
3   57   257                REPEAT
2   58   258                CALL FOUT(GOTO,1,6)
2   58   259                CALL FOUT(LABEL,LAB1,LAB1+4)
```

```
2   58   260                  CALL FPRNT
2   58   261                  // GO TO LAB1. //
2   58   262                  FOR I=1 TO 5 DO
3   59   263                     L=LAB2+I-1
3   59   264                     FBUF(I)=LABEL(L)
3   59   265                  REPEAT
2   60   266                  FBUF(6)=BLANK
2   60   267                  CALL FOUT(SPIN,VBLEA,VBLEB)
2   60   268                  CALL FOUT(EQL,1,1)
2   60   269                  CALL FOUT(SPIN,VBLEA,VBLEB)
2   60   270                  CALL FOUT(ELL,1,1)
2   60   271                  IF ECOND1 THEN
3   61   272                     // THE 'BY' IS MISSING--DEFAULT INCREMENT OF EXPRESSION TO ONE. /
3   61   273                     CALL FOUT(ONE,1,1)
3   61   274                  ELSE
3   62   275                     CALL FOUT(IFNOT,9,9)
3   62   276                     CALL FOUT(SPIN,TOBEG,TCEND-2)
3   62   277                     CALL FOUT (RSTAR,1,1)
3   62   278                  ENDIF
2   63   279                  CALL FPRNT
2   63   280                  //  LAB2--INDICATES VARIABLE=VARIABLE+EXPRESSION3. //
2   63   281                  FOR I=1 TO 5 DO
3   64   282                     L=LAB1+I-1
3   64   283                     FBUF(I)=LABEL(L)
3   64   284                  REPEAT
2   65   285                  FBUF(6)=BLANK
2   65   286                  CALL FOUT(IFNOT,1,9)
2   65   287                  CALL FOUT(IFNOT,9,9)
2   65   288                  CALL FOUT(SPIN,VBLEA,VBLEB)
2   65   289                  CALL FOUT(MINUS,1,1)
2   65   290                  CALL FOUT(IFNOT,9,9)
2   65   291                  CALL FOUT(SPIN,EX2A,EX2B)
2   65   292                  CALL FOUT(RSTAR,1,1)
2   65   293                  CALL FOUT(RSTAR,1,1)
2   65   294                  IF .NOT.ECOND1 THEN
3   66   295                     // IF 'BY' MISSING THEN GENERATE...(EXP2)).LE.0... //
3   66   296                     //  IN PLACE OF...(EXP2)*(EXP3).LE.0..... //
3   66   297                     CALL FOUT(RSTAR,2,2)
```

```
3   66   298                  CALL FOUT(IFNOT,9,9)
3   66   299                  CALL FOUT(SPIN,TCBEG,TCEND-2)
3   66   300                  CALL FOUT(RSTAR,1,1)
3   66   301               ENDIF
2   67   302               CALL FOUT(LEC,1,5)
2   67   303               CALL FOUT(CGCTC,1,9)
2   67   304               CALL FOUT(LABEL,LAB4,LAB4+4)
2   67   305               CALL FPRNT
2   67   306               // LAB1...IF(.NOT.((VBLE-(EXP3).LE.0) GO TO LAB3... //
2   67   307               STEND=TCEND
2   67   308               INDFLG=FRWRD
2   67   309               CURFLG=CLO
2   67   310               IF FSTRT+INCR.LT.IMAX THEN
3   68   311                  FSTRT=FSTRT+INCR
3   68   312               ENDIF
2   69   313            : TYPE.EQ.RREPT :
2   70   314               CALL CHECK(RREPT,BAD)
2   70   315               IF BAD THEN
3   71   316                  CYCLE
3   71   317               ENDIF
2   72   318               IF LBFLG THEN
3   73   319                  //  LABELLED STMT, GENERATED 'LAB CONTINUE' //
3   73   320                  CALL FOUT(CONTI,1,8)
3   73   321                  CALL FPRNT
3   73   322               ENDIF
2   74   323               STTOP=STACK(TOP)
2   74   324               // LAB3=EXIT TARGET,LAB2=CYCLE TARGET,LAB1=LOOP BACK LABEL //
2   74   325               IF STTOP.EQ.FFOR THEN
3   75   326                  LAB1=STACK(TOP-5)
3   75   327                  LAB2=STACK(TOP-2)
3   75   328                  LAB3=STACK(TOP-1)
3   75   329                  TOP=TOP-6
3   75   330                  LBPTR=LBPTR-20
3   75   331               ELSE
3   76   332                  LAB1=STACK(TOP-3)
3   76   333                  LAB2=STACK(TOP-2)
3   76   334                  LAB3=STACK(TOP-1)
3   76   335                  TOP=TOP-5
```

```
3   76   336                  LBPTR=LBPTR-15
3   76   337               ENDIF
2   77   338               TPITR=STACK(TPITR-4)
2   77   339               FSTRT=FSTRT-INCR
2   77   340               FPTR=FSTRT
2   77   341               FOR I=1 TO 5 DO
3   78   342                  L=LAB2+I-1
3   78   343                  FBUF(I)=LABEL(L)
3   78   344               REPEAT
2   79   345               FBUF(6)=BLANK
2   79   346               CALL FOUT(GOTO,1,6)
2   79   347               CALL FOUT(LABEL,LAB1,LAB1+4)
2   79   348               CALL FPRNT
2   79   349               //    GO TO LABEL 1 //
2   79   350               FOR I=1 TO 5 DO
3   80   351                  L=LAB3+I-1
3   80   352                  FBUF(I)=LABEL(L)
3   80   353               REPEAT
2   81   354               FBUF(6)=BLANK
2   81   355               CALL FOUT(CONTI,1,8)
2   81   356               CALL FPRNT
2   81   357               //    LAB3 CONTINUE    //
2   81   358               STEND=TOEND
2   81   359               CURFLG=NEW
2   81   360               INDFLG=BKWRD
2   81   361            : TYPE.EQ.CCASE :
2   82   362               LAB=GENLAB(1)
2   82   363               //  LABEL(1:5)=BLANKS. //
2   82   364               STACK(TOP+1)=1
2   82   365               STACK(TOP+2)=LAB
2   82   366               STACK(TOP+3)=CCASE
2   82   367               TOP=TOP+3
2   82   368               IF TOP.GT.STMAX THEN
3   83   369                  CALL ERROR(6,0,0,5,0,0)
3   83   370               ENDIF
2   84   371               //  STACK OVERFLOWED----PROGRAM TERMINATED. //
2   84   372               STEND=TOEND
2   84   373               CURFLG=OLD
```

```
2   84   374                  INDFLG=FRWRD
2   84   375                  IF LBFLG THEN
3   85   376                     //  LABELED STATEMENT--GENERATE 'LAB CONTINUE'. //
3   85   377                     CALL FOUT(CONTI,1,8)
3   85   378                     CALL FPRNT
3   85   379                  ENDIF
2   86   380                  //  SET THE INDENTION  //
2   86   381                  IF FSTRT+INCR.LT.IMAX THEN
3   87   382                     FSTRT=FSTRT+INCR
3   87   383                  ENDIF
2   88   384               : TYPE.EQ.CCLCN :
2   89   385                  CALL CHECK(CCLCN,BAD)
2   89   386                  IF BAD THEN
3   90   387                     CYCLE
3   90   388                  ENDIF
2   91   389                  IF LBFLG THEN
3   92   390                     CALL ERROR(1,CASS,4,2,0,0)
3   92   391                     // ERROR--LABELLING THE CASE HAMPERS PROGRAM CLEANLINESS //
3   92   392                     // DELETED FROM THE FORTRAN OUTPUT. //
3   92   393                     FOR I=1 TO 6 DO
4   93   394                        FBUF(I)=BLANK
4   93   395                     REPEAT
3   94   396                  ENDIF
2   95   397                  LAB1=STACK(TOP-1)
2   95   398                  LAB2=STACK(TOP-2)
2   95   399                  FSTRT=FSTRT-INCR
2   95   400                  FPTR=FSTRT
2   95   401                  // CASE MATERIAL PRINTED ONE IDENT TO THE RIGHT. //
2   95   402                  IF LAB2.NE.1 THEN
3   96   403                     CALL FOUT(GOTO,1,6)
3   96   404                     CALL    FOUT(LABEL,LAB1,LAB1+4)
3   96   405                     CALL    FPRNT
3   96   406                     //  GO TO LAB1.  //
3   96   407                  ENDIF
2   97   408                  CALL SCAN1(CCLN,ECOND)
2   97   409                  IF ECOND THEN
3   98   410                     CALL ERROR(2,CCLN,1,1,0,0)
3   98   411                  ENDIF
```

```
2   99   412                    // ERROR--MISSING COLON-SUPPLIED AT END OF STATEMENT. //
2   99   413                    K=TOEND
2   99   414                    TCEND=TOBEG
2   99   415                    // SPIN(K)=IMAX OR SPIN(K+1)=SEMICOLON OR COLON. //
2   99   416                    TMAX=IMAX
2   99   417                    IMAX=K
2   99   418                    CALL SCAN(ELZ,4,ECOND)
2   99   419                    IMAX=TMAX
2   99   420                    IF ECOND THEN
3  100   421                       LAB3=GENLAB(1)
3  100   422                       TCEND=K
3  100   423                       // OUTPUT-LAB2-IF NOT CONDITION THE GO TO LAB3. //
3  100   424                       FOR I=1 TO 5 DO
4  101   425                          L=LAB2+I-1
4  101   426                          FBUF(I)=LABEL(L)
4  101   427                       REPEAT
3  102   428                       FBUF(6)=BLANK
3  102   429                       CALL    FOUT(IFNOT,1,9)
3  102   430                       CALL FOUT(SPIN,TOBEG,TCEND)
3  102   431                       CALL    FOUT(CGOTO,1,9)
3  102   432                       CALL FOUT(LABEL,LAB3,LAB3+4)
3  102   433                       CALL    FPRNT
3  102   434                       TCEND=TCEND+1
3  102   435                    ELSE
3  103   436                       //  ELSE CLAUSE FOUND  //
3  103   437                       FOR I=1 TO 5 DO
4  104   438                          L=LAB2+I-1
4  104   439                          FBUF(I)=LABEL(L)
4  104   440                       REPEAT
3  105   441                       FBUF(6)=BLANK
3  105   442                       CALL FOUT(CONTI,1,8)
3  105   443                       CALL FPRNT
3  105   444                       //  LAB2 CONTINUE  //
3  105   445                       TCEND=K+1
3  105   446                       LAB3=1
3  105   447                       //  LABEL(1....5) ARE BLANKS.  //
3  105   448                    ENDIF
2  106   449                    STACK(TOP-2)=LAB3
```

```
2  106   450                  INDFLG=NEUTR
2  106   451                  CURFLG=PULOUT
2  106   452                  STEND=TOEND
2  106   453                  //  RESTORE THE INDENTATION.  //
2  106   454                  IF FSTRT+INCR.LT.IMAX THEN
3  107   455                     FSTRT=FSTRT+INCR
3  107   456                  ENDIF
2  108   457               : TYPE.EQ.ENDCAS :
2  109   458                  CALL CHECK(ENDCAS,BAD)
2  109   459                  IF BAD THEN
3  110   460                     CYCLE
3  110   461                  ENDIF
2  111   462                  FSTRT=FSTRT-INCR
2  111   463                  FPTR=FSTRT
2  111   464                  IF LBFLG THEN
3  112   465                     // LABELLED STATEMENT--GENERATE 'LAB CONTINUE' //
3  112   466                     CALL FOUT(CONTI,1,8)
3  112   467                     CALL FPRNT
3  112   468                  ENDIF
2  113   469                  LAB1=STACK(TOP-1)
2  113   470                  LAB2=STACK(TOP-2)
2  113   471                  TOP=TOP-3
2  113   472                  LBPTR=LAB1
2  113   473                  FOR I=1 TO 5 DO
3  114   474                     L=LAB2+I-1
3  114   475                     FBUF(I)=LABEL(L)
3  114   476                  REPEAT
2  115   477                  FBUF(6)=BLANK
2  115   478                  CALL FOUT(CONTI,1,8)
2  115   479                  CALL  FPRNT
2  115   480                  //  LAB2 CONTINUE  //
2  115   481                  FOR I=1 TO 5 DO
3  116   482                     L=LAB1+I-1
3  116   483                     FBUF(I)=LABEL(L)
3  116   484                  REPEAT
2  117   485                  FBUF(6)=BLANK
2  117   486                  CALL FOUT(CONTI,1,8)
2  117   487                  CALL FPRNT
```

```
2  117   488                  //  LAB1 CONTINUE  //
2  117   489                  STEND=TCENC
2  117   490                  INDFLG=BKWRD
2  117   491                  CURFLG=NEW
2  117   492               : TYPE.EQ.LLOOP :
2  118   493                  //  LAB2 IS THE TARGET OF CYCLE AND LAB3 OF EXITS //
2  118   494                  //  LAB1 IS THE IDENTIFYING LABEL.  //
2  118   495                  LAB1=GENLAB(0)
2  118   496                  LAB2=GENLAB(1)
2  118   497                  LAB3= GENLAB(1)
2  118   498                  STACK(TOP+2)=LAB1
2  118   499                  STACK(TOP+3)=LAB2
2  118   500                  STACK(TOP+4)=LAB3
2  118   501                  STACK(TOP+5)=LLOOP
2  118   502                  STACK(TOP+1)=TPITR
2  118   503                  TOP=TOP+5
2  118   504                  IF TOP.GT.STMAX THEN
3  119   505                     CALL ERROR(6,0,0,5,0,0)
3  119   506                  ENDIF
2  120   507                  //  STACK OVERFLOWED, PROGRAM TERMINATED //
2  120   508                  FOR I=1 TO 5 DO
3  121   509                     L=LAB1+I-1
3  121   510                     FBUF(I)=LABEL(L)
3  121   511                  REPEAT
2  122   512                  FBUF(6)=BLANK
2  122   513                  CALL FOUT(CONTI,1,8)
2  122   514                  CALL FPRNT
2  122   515                  TPITR=TOP
2  122   516                  CURFLG=OLD
2  122   517                  INDFLG=FRWRD
2  122   518                  STEND=TCENC
2  122   519                  IF FSTRT+INCR.LT.IMAX THEN
3  123   520                     FSTRT=FSTRT+INCR
3  123   521                  ENDIF
2  124   522               : TYPE.EQ.WHLE :
2  125   523                  LAB1=GENLAB(0)
2  125   524                  LAB2=GENLAB(1)
2  125   525                  LAB3=GENLAB(1)
```

```
2  125   526                    //  LAB2 IS THE TARGET OF EXITS, LAB3 FOR CYCLES.  //
2  125   527                    CALL SCAN(COC,2,ECCND)
2  125   528                    IF ECCND THEN
3  126   529                       CALL ERROR(2,DCC,2,1,0,0)
3  126   530                       //  CALL ERROR---MISSING 'DO'-SUPPLIED AT THE END OF STMT.  //
3  126   531                       TOEND=TOEND+1
3  126   532                    ELSE
3  127   533                       TOEND=TOEND+2
3  127   534                    ENDIF
2  128   535                    IF .NOT.LBFLG THEN
3  129   536                       FOR I=1 TO 5 DO
4  130   537                          L=LAB1+I-1
4  130   538                          FBUF(I)=LABEL(L)
4  130   539                       REPEAT
3  131   540                       FBUF(6)=BLANK
3  131   541                    ENDIF
2  132   542                    CALL FOUT(IFNOT,1,9)
2  132   543                    CALL FOUT(SPIN,TOBEG,TOEND-2)
2  132   544                    CALL FOUT(CGOTO,1,9)
2  132   545                    CALL FOUT(LABEL,LAB2,LAB2+4)
2  132   546                    CALL FPRNT
2  132   547                    //  GENERATE LAB1 AND IF TRUE COND GO TO LAB2.  //
2  132   548                    STACK(TOP+2)=LAB1
2  132   549                    STACK(TOP+3)=LAB3
2  132   550                    STACK(TOP+4)=LAB2
2  132   551                    STACK(TOP+5)=WHLE
2  132   552                    STACK(TOP+1)=TPITR
2  132   553                    TOP=TOP+5
2  132   554                    IF TOP.GT.STMAX THEN
3  133   555                       CALL ERROR(6,0,0,5,0,0)
3  133   556                    ENDIF
2  134   557                    //  STACK OVERFLOWED, PROGRAM TERMINATED.  //
2  134   558                    TPITR=TOP
2  134   559                    CURFLG=OLD
2  134   560                    INDFLG=FRWRD
2  134   561                    STEND=TOEND
2  134   562                    IF FSTRT+INCR.LT.IMAX THEN
3  135   563                       FSTRT=FSTRT+INCR
```

```
3  135  564                ENDIF
2  136  565              : TYPE.EQ.CONT :
2  137  566                FOR I=1 TO 6 DO
3  138  567                   FBUF(I)=SPIN(I)
3  138  568                REPEAT
2  139  569                TOBEG=7
2  139  570                WHILE SPIN(TOBEG).EQ.BLANK DO
3  140  571                   TOBEG=TOBEG+1
3  140  572                   IF TOBEG.GT.IMAX THEN
4  141  573                      TOEND=IMAX+1
4  141  574                      // BLANK CONTINUATION CARD, DELETED.  //
4  141  575                      EXIT
4  141  576                   ENDIF
3  142  577                REPEAT
2  143  578                IF TOEND.GT.IMAX THEN
3  144  579                   CYCLE
3  144  580                ENDIF
2  145  581                // SPIN(TOBEG)=NONBLANK; REST OF CARD CAN BE INDENTED.  //
2  145  582                CALL FHAND
2  145  583                FBUF(6)=BLANK
2  145  584                STBEG=TOBEG
2  145  585                STEND=TOEND
2  145  586                CURFLG=OLD
2  145  587                INDFLG=NEUTR
2  145  588              : TYPE.EQ.EEXIT :
2  146  589                CALL CHECK(EEXIT,BAD)
2  146  590                IF BAD THEN
3  147  591                   CYCLE
3  147  592                ENDIF
2  148  593                CALL TYHAND(TYPE,EEEXT,CCYCLE)
2  148  594              : TYPE.EQ.NCYCLE :
2  149  595                CALL CHECK(EEXIT,BAD)
2  149  596                IF BAD THEN
3  150  597                   CYCLE
3  150  598                ENDIF
2  151  599                CALL TYHAND(TYPE,EEEXT,CCYCLE)
2  151  600              : TYPE.EQ.UNTL :
2  152  601                CALL CHECK(UNTL,BAD)
```

```
2  152  602                IF BAD THEN
3  153  603                   CYCLE
3  153  604                ENDIF
2  154  605                FSTRT=FSTRT-INCR
2  154  606                FPTR=FSTRT
2  154  607                LAB1=STACK(TOP-3)
2  154  608                LAB2=STACK(TOP-2)
2  154  609                LAB3=STACK(TOP-1)
2  154  610                TOP=TOP-5
2  154  611                LBPTR=LBPTR-15
2  154  612                TPITR=STACK(TPITR-4)
2  154  613                IF LBFLG THEN
3  155  614                   CALL ERROR(1,UUNTL,5,2,0,0)
3  155  615                   //  ERROR---LABELLING OF THE UNTIL HAMPERS PROGRAM  //
3  155  616                   //  CLARITY, DELETED FROM FORTRAN OUTPUT.  //
3  155  617                ENDIF
2  156  618                FOR I=1,TO 5 DO
3  157  619                   L=LAB2+I-1
3  157  620                   FBUF(I)=LABEL(L)
3  157  621                REPEAT
2  158  622                FBUF(6)=BLANK
2  158  623                CALL FOUT(IFNOT,1,9)
2  158  624                CALL SCAN(REPT,6,ECOND)
2  158  625                CALL FOUT(SPIN,TOBEG,TCEND)
2  158  626                IF ECOND THEN
3  159  627                   CALL ERROR(2,REPT,6,1,0,0)
3  159  628                   // ERROR---MISSING REPEAT-SUPPLIED AT THE END OF THE STATEMENT. /
3  159  629                   TCEND=TCEND+1
3  159  630                ELSE
3  160  631                   TCEND=TCEND+6
3  160  632                ENDIF
2  161  633                CALL FOUT( GOTO,1,9)
2  161  634                CALL FOUT (LABEL,LAB1,LAB1+4)
2  161  635                CALL FPRNT
2  161  636                //  GENERATE IF(.NOT.(COND)) GO TO LAB1  //
2  161  637                //  GENERATE THE EXIT TARGET---LAB3 CONTINUE.  //
2  161  638                FOR I=1 TO 5 DO
3  162  639                   L=LAB3+I-1
```

```
3  162   640                        FBUF(I)=LABEL(L)
3  162   641                     REPEAT
2  163   642                     FBUF(6)=BLANK
2  163   643                     CALL FOUT(CONTI,1,8)
2  163   644                     CALL FPRNT
2  163   645                     STEND=TOEND
2  163   646                     INDFLG=BKWRD
2  163   647                     CURFLG=NEW
2  163   648                  : TYPE.EQ.SUBFN :
2  164   649                     WRITE(PRTR,5711)
2  164   650     5711            FORMAT(1H1)
2  164   651                     NUMB=100000
2  164   652                     STACK(TOP+1)=SUBFN
2  164   653                     TOP=TOP+1
2  164   654                     IF TOP.GT.STMAX THEN
3  165   655                        CALL ERROR(6,0,0,5,0,0)
3  165   656                     ENDIF
2  166   657                     // STACK OVERFLOWED, PROGRAM TERMINATED.  //
2  166   658                     CALL FHAND
2  166   659                     CURFLG=OLD
2  166   660                     INDFLG=FRWRD
2  166   661                     STEND=TOEND
2  166   662                     IF FSTRT+INCR.LT.IMAX THEN
3  167   663                        FSTRT=FSTRT+INCR
3  167   664                     ENDIF
2  168   665                  : TYPE.EQ.EEND :
2  169   666                     CALL CHECK(FEND,BAD)
2  169   667                     IF BAD THEN
3  170   668                        CYCLE
3  170   669                     ENDIF
2  171   670                     FSTRT=FSTRT-INCR
2  171   671                     FPTR=FSTRT
2  171   672                     IF LBFLG THEN
3  172   673                        // LABELLED STATEMENT--GENERATE 'LAB CONTINUE' //
3  172   674                        CALL FOUT(CONTI,1,8)
3  172   675                        CALL FPRNT
3  172   676                     ENDIF
2  173   677                     TOP=TOP-1
```

```
2  173   678                      CALL FCUT(SPIN,TOBEG,TCEND)
2  173   679                      CALL FPRNT
2  173   680                      // GIVE ANY END OF SUBROUTINE MESSAGES AND RESET ANY VARIBLES //
2  173   681                      // IF NECESSARY. //
2  173   682                      STEND=TCEND
2  173   683                      INDFLG=BKWRD
2  173   684                      CURFLG=NEW
2  173   685                    : TYPE.EQ.CCMT :
2  174   686                      FBUF(1)=CEE
2  174   687                      TOBEG=2
2  174   688                      TCEND=IMAX
2  174   689                      LOOP
3  175   690                         IF SPIN(TOBEG).NE.BLANK THEN
4  176   691                            FOR I=2 TO 6 DO
5  177   692                               FBUF(I)=BLANK
5  177   693                            REPEAT
4  178   694                            CALL FCUT(SPIN,TOBEG,IMAX)
4  178   695                            CALL FPRNT
4  178   696                            STBEG=TOBEG
4  178   697                            EXIT
4  178   698                         ENDIF
3  179   699                         TOBEG=TOBEG+1
3  179   700                         IF TOBEG.GT.IMAX THEN
4  180   701                            // A BLANK COMMENT, OUTPUT A BLANK LINE.  //
4  180   702                            CALL FCUT(SPIN,FPTR,FMAX)
4  180   703                            CALL FPRNT
4  180   704                            STBEG=SSTRT
4  180   705                            EXIT
4  180   706                         ENDIF
3  181   707                      REPEAT
2  182   708                      STEND=IMAX
2  182   709                      CURFLG=CLD
2  182   710                      INDFLG=NEUTR
2  182   711                    : TYPE.EQ.EOJ :
2  183   712                      //  CLOSE THE FILE.  //
2  183   713                      WRITE(PRTR,1300)
2  183   714    1300              FORMAT(/35H          END OF SPARKS PROGRAM.....)
2  183   715                      STOP
```

```
2  183   716                ENDCASE
1  184   717                IF CURFLG.EQ.PULOUT THEN
2  185   718                   SSTRT=SSTRT-INCR
2  185   719                   CALL SPRNT
2  185   720                   SSTRT=SSTRT+INCR
2  185   721                   IF SSTRT.GT.SMAX THEN
3  186   722                      SSTRT=SSTRT-INCR
3  186   723                   ENDIF
2  187   724                   BLKNO=BLKNO+1
2  187   725                ENDIF
1  188   726                IF CURFLG.EQ.CLO THEN
2  189   727                   CALL SPRNT
2  189   728                ENDIF
1  190   729                IF INDFLG.EQ.FRWRD THEN
2  191   730                   SSTRT=SSTRT+INCR
2  191   731                   IF SSTRT.GT.SMAX THEN
3  192   732                      SSTRT=SSTRT-INCR
3  192   733                   ENDIF
2  193   734                   IF CURFLG.EQ.NEW THEN
3  194   735                      CALL SPRNT
3  194   736                   ENDIF
2  195   737                   BLKNO=BLKNO+1
2  195   738                   NESTNO=NESTNO+1
2  195   739                ENDIF
1  196   740                IF INDFLG.EQ.BKWRD THEN
2  197   741                   SSTRT=SSTRT-INCR
2  197   742                   IF CURFLG.EQ.NEW THEN
3  198   743                      CALL SPRNT
3  198   744                   ENDIF
2  199   745                   NESTNO=NESTNO-1
2  199   746                   BLKNO=BLKNO+1
2  199   747                ENDIF
1  200   748                STMNO=STMNO+1
1  200   749             REPEAT
0  201   750          END
```

```
-1  202   751              BLOCK DATA
 0  203   752                 IMPLICIT INTEGER(A-Z)
 0  203   753                 LOGICAL LBFLG
 0  203   754                 COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  203   755                 COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX
 0  203   756                 COMMON/FRTINF/FBLF(80),FPTR,FMAX,FSTRT,BLNKS(131)
 0  203   757                 COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),
 0  203   758           1     TPITR,NEUTR,BKWRD,FRWRD,NEW,CLD,PULOUT,STMAX,NUMB
 0  203   759                 COMMON/OTPAT/CGOTO(9),CONTI(8),EQL(1),GOTO(6),IFNOT(9),LEQ(5),
 0  203   760           1     MINUS(1),ONE(1),PLU(1),RSTAR(2)
 0  203   761                 COMMON/INPAT/CCLN(1),ELZ(4),DBSLH(2),DOC(2),BYE(2),THENN(4),TOO(2)
 0  203   762                 COMMON/STTYP1/CCASE,COLON,DBLSL,EELSE,EEND,EOJ,EEXIT,FFOR,IIF,
 0  203   763           1     LLOOP,RREPT,UNTL,WHLE,SUBFN,RINLC,NCYCLE,ENDCAS,EENDIF
 0  203   764                 COMMON/STTYP2/COMT,CONT,FOTRN,BLANK,SEMI
 0  203   765                 COMMON/TABLES/TRES(23),RESRD(200)
 0  203   766                 COMMON/NMNIC/CEE,PRTR,FFILE,CRDR
 0  203   767                 COMMON/KEYWD/EEEND(3),EEEXT(4),CCYCLE(5),UUNTL(5),CASS(4),IIIF(2),
 0  203   768           1     ITERAT(9),ENCASS(7),ENIF(5),REPT(6),CCCLCN(5)
 0  203   769                 //              //
 0  203   770                 //  CRITICAL CODES--STFRM IS ACCESSED USING THE FOLLOWING: //
 0  203   771                 //  CCASE=1, FFOR=2, IIF=3, LLOOP=4, WHLE=5, SUBFN=6.  //
 0  203   772                 //              //
 0  203   773                 DATA SPIN/1HT,1HH,1HI,1HS,1H ,1HI,1HS,1H ,1HA,1H ,1HS,1HP,1HA,
 0  203   774           X     1HR,1HK,1HS,1H ,1HP,1HR,1HO,1HG,1HR,1HA,1HM,
 0  203   775           1     1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,  1H ,  1H ,1H ,1H ,1H ,
 0  203   776           2     1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,  1H ,  1H ,1H ,1H ,1H ,
 0  203   777           3     1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,  1H ,  1H ,1H ,1H ,1H ,
 0  203   778           4     1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,  1H ,  1H ,1H ,1H ,1H /
 0  203   779                 DATA BLNKS/131*1H /
 0  203   780                 DATA STMAX,LBMAX,NUMB/200,200,100000/
 0  203   781                 DATA NEUTR,BKWRD,FRWRD,NEW,CLD,PULOUT/1,2,3,1,2,3/
 0  203   782                 //   STACK FRAME SIZES: //
 0  203   783                 //              CASE    FOR       IF     LOOP    SUBFN     WHILE  //
 0  203   784                 DATA STFRM /  3   ,   6   ,   2   ,   5   ,    1  ,    5    /
 0  203   785                 DATA    CGOTO/1H),1H),1H ,1HG ,1HO,1H ,1HT,1HO,1H /
 0  203   786                 DATA    CONTI/1HC,1HC,1HN,1HT,1HI,1HN,1HU,1HE/
 0  203   787                 DATA    EQL/1H=/
```

```
0  203   788              DATA    GOTO /1HG,1HO,1H ,1HT,1HO,1H /
0  203   789              DATA    IFNOT/1HI,1HF,1H(,1H.,1HN,1HO,1HT,1H.,1H(/
0  203   790              DATA    LABEL/1H ,1H ,1H ,1H ,1H ,195*1H /
0  203   791              DATA    LEQ/1H.,1HL,1HE,1H.,1HQ/
0  203   792              DATA    MINUS/1H-/
0  203   793              DATA    ONE/1H1/
0  203   794              DATA    PLU/1H+/
0  203   795              DATA    RSTAR/1H),1H*/
0  203   796              DATA DBSLH,ELZ/1H/,1H/,1HE,1HL,1HS,1HE/
0  203   797              DATA COLN,COLON/1H:,1HC,1HO,1HL,1HO,1HN/
0  203   798              DATA DOO,BYE,THEN,REPT,TOO/1HD,1HO,1HB,1HY,1HT,1HH,1HE,1HN,1HR,
0  203   799         1    1HE,1HP,1HE,1HA,1HT,1HT,1HO/
0  203   800              DATA     OCASE,OCLON,DBLSL,EELSE,EEND,EOJ,EEXIT,FFOR,COMT,
0  203   801         1    CONT,FOTRN,IIF,LLOOP,RREPT,UNTL,WHLE,SUBFN,RINLO
0  203   802         1    ,NCYCLE,ENDCAS,EENDIF
0  203   803         1    /1,7,8,9,10,11,12,2,13,14,15,3,4,16,17,6,5,18,19,20,21/
0  203   804              DATA RESRD/4,1HC,1HA,1HS,1HE,1,170,
0  203   805         1    3,1HE,1HN,1HD,10,14,4,1HE,1HL,1HS,1HE,9,21,
0  203   806         2    3,1HE,1HO,1HJ,11,27,4,1HE,1HX,1HI,1HT,12,152,
0  203   807         3    3,1HF,1HO,1HR,2,133,
0  203   808         4    2,1HI,1HF,3,45,7,1HI,1HN,1HT,1HE,1HG,1HE,1HR,18,0,
0  203   809         5    7,1HL,1HO,1HG,1HI,1HC,1HA,1HL,18,65,4,1HL,1HC,1HO,1HP,4
0  203   810         6    ,0,4,1HR,1HE,1HA,1HL,18,79,6,1HR,1HE,1HP,1HE,1HA,1HT,
0  203   811         7    16,0,5,1HU,1HN,1HT,1HI,1HL,17,0,5,1HW,1HH,1HI,1HL,1HE,6
0  203   812         8    ,0,4,1HN,1HE,1HX,1HT,0,0,10,1HS,1HU,1HB,1HR,1HO,1HU,1HT
0  203   813         9    ,1HI,1HN,1HE,5,0,2,1H/,1H/,8,0
0  203   814         A    ,1,1H:,7,0,8,1HF,1HL,1HN,1HC,1HT,1HI,1HO,1HN,5,0,
0  203   815         B    5,1HB,1HL,1HO,1HC,1HK,5,0,7,1HE,1HN,1HD,1HC,1HA,1HS,1HE
0  203   816         C    ,20,162,5,1HE,1HN,1HC,1HI,1HF,21,0,5,1HC,1HY,1HC,1HL,1HE
0  203   817         D    ,19,0,23*0/
0  203   818              //   178TH PLACE AVAILABLE FOR INSERTION. //
0  203   819              DATA TRES/144,1,170,14,8,152,162,21,27,34,133,40,45,55,65,
0  203   820         1    104,72,79,111,88,96,124,129/
0  203   821              // TRES CONTAINS POINTS INTO RESRD WHICH CONTAINS ALL THE //
0  203   822              // RESERVE WORDS IN SPARKS.  *****  NOTE  *****  THIS ORDER IS  //
0  203   823              // PRESENTLY SET UP FOR EBCDIC. //
0  203   824              DATA PRTR,CRDR,FFILE/ 6, 5, 4/
0  203   825              DATA BLANK,CEE,SEMI/1H ,1HC,1H;/
```

```
0  203    826              DATA EEEND,ENCASS/1HE,1HN,1HC,1HE,1HN,1HD,1HC,1HA,1HS,1HE/
0  203    827              DATA EEEXT,ENIF/1HE,1HX,1HI,1HT,1HE,1HN,1HD,1HI,1HF/
0  203    828              DATA CCYCLE,CASS/1HC,1HY,1HC,1HL,1HE,1HC,1HA,1HS,1HE/
0  203    829              DATA UUNTL,IIIF/1HU,1HN,1HT,1HI,1HL,1HI,1HF/
0  203    830              DATA ITERAT/1HI,1HT,1HE,1HR,1HA,1HT,1HI,1HV,1HE/
0  203    831          END
```

```
-1  204   832          SUBROUTINE DETNX(TYPE)
 0  205   833             //           //
 0  205   834             // THE SUBROUTINE DETERMINES THE TYPE OF CYCLE STATEMENT AND //
 0  205   835             // SETS TOBEG AND TCEND ON TOKEN BOUNDARIES. //
 0  205   836             //           //
 0  205   837             IMPLICIT INTEGER(A-Z)
 0  205   838             LOGICAL LBFLG,ECOND
 0  205   839             COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  205   840             COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)
 0  205   841             COMMON/STTYP2/COMT,CONT,FCTRN,BLANK,SEMI
 0  205   842             COMMON/NMNIC/CEE,PRTR,FFILE,CRDR
 0  205   843             DATA EFF/1HF/
 0  205   844             //           //
 0  205   845             LBFLG=.FALSE.
 0  205   846             LOOP
 1  206   847                LOOP
 2  207   848                   TOEND=TOEND+1
 2  207   849                   IF TCEND.GT.IMAX THEN
 3  208   850                      EXIT
 3  208   851                   ENDIF
 2  209   852                   IF SPIN(TOEND).NE.BLANK THEN
 3  210   853                      IF SPIN(TCEND).EQ.SEMI THEN
 4  211   854                         CYCLE
 4  211   855                      ENDIF
 3  212   856                      IF SPIN(1).EQ.CEE THEN
 4  213   857                         TOBEG=1
 4  213   858                         TCEND=1
 4  213   859                         TYPE=COMT
 4  213   860                         RETURN
 4  213   861                      ENDIF
 3  214   862                      TOBEG=TCEND
 3  214   863                      CALL SCAN1(BLANK,ECOND)
 3  214   864                      CASE
 4  215   865                      : ECOND :
 4  216   866                         TYPE=FCTRN
 4  216   867                         RETURN
 4  216   868                      : SPIN(6).NE.BLANK :
```

```
4  217   869                        TYPE=CCNT
4  217   870                        RETURN
4  217   871                     : TCEND.GT.6 :
4  218   872                        CALL STTYP(TYPE)
4  218   873                        IF .NOT.LBFLG THEN
5  219   874                           FOR I=1 TO 6 DO
6  220   875                              FBUF(I)=BLNKS(I)
6  220   876                           REPEAT
5  221   877                        ENDIF
4  222   878                        RETURN
4  222   879                     : TCEND.LE.6 :
4  223   880                        FOR I=1 TO 5 DO
5  224   881                           BLNKS(I)=SPIN(I)
5  224   882                           FBUF(I)=SPIN(I)
5  224   883                        REPEAT
4  225   884                        LBFLG=.TRUE.
4  225   885                        FBUF(6)=BLANK
4  225   886                     ENDCASE
3  226   887                  ENDIF
2  227   888               REPEAT
1  228   889               READ(CRDR,20) SPIN
1  228   890         20    FORMAT(80A1)
1  228   891               IF SPIN(1).EQ.EFF THEN
2  229   892                  WRITE(PRTR,31)(SPIN(I),I=1,80)
2  229   893         31       FORMAT(34H  -----BEGINNING OF FORTRAN-----  ,80A1)
2  229   894                  READ(CRDR,20)SPIN
2  229   895                  WHILE SPIN(1).NE.EFF DO
3  230   896                     WRITE(PRTR,33)(SPIN(I),I=1,80)
3  230   897         33          FORMAT(20X,80A1)
3  230   898                     WRITE(FFILE,32)(SPIN(I),I=1,80)
3  230   899         32          FORMAT(80A1)
3  230   900                     READ(CRDR,20) SPIN
3  230   901                  REPEAT
2  231   902                  WRITE(PRTR,34) (SPIN(I),I=1,80)
2  231   903         34       FORMAT(28H  -----END OF FORTRAN-----  ,80A1)
2  231   904                  READ(CRDR,20)SPIN
2  231   905               ENDIF
1  232   906               TOEND=0
```

```
1  232    907                REPEAT
0  233    908              END
```

```
-1  234   909          SUBRCUTINE STTYP(TYPE)
 0  235   910             //  THE TOKEN AT SPIN(TCBEG) TO SPIN(TCEND) IS SEARCHED //
 0  235   911             //    FCR IN THE SYMECL TABLE. //
 0  235   912             //            //
 0  235   913             IMPLICIT INTEGER(A-Z)
 0  235   914             CCMMCN/INPUT/SPIN(80),TCBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  235   915             COMMON/STTYP2/COMT,CCNT,FOTRN,BLANK,SEMI
 0  235   916             CCMMCN/TABLES/TRES(23),RESRD(200)
 0  235   917             //            //
 0  235   918             TCP=1
 0  235   919             BCTTOM=23
 0  235   920             LENGTH=TCEND-TCBEG+1
 0  235   921             WHILE TOP.LE.BCTTOM DO
 1  236   922                MID=(TOP+BCTTCM)/2
 1  236   923                I=TRES(MID)
 1  236   924                CASE
 2  237   925                : SPIN(TCBEG).LT.RESRD(I+1) :
 2  238   926                   BCTTOM=MID-1
 2  238   927                : SPIN(TCBEG).GT.RESRD(I+1) :
 2  239   928                   TCP=MID+1
 2  239   929                : SPIN(TCBEG).EC.RESRD(I+1) :
 2  240   930                   IF LENGTH.EQ.1 THEN
 3  241   931                      TYPE=RESRD(I+2)
 3  241   932                      RETURN
 3  241   933                   ENDIF
 2  242   934                   PNT=2
 2  242   935                   WHILE PNT.LE.RESRD(I).AND.PNT.LE.LENGTH DO
 3  243   936                      ITEMP=I+PNT
 3  243   937                      JTEMP=TCBEG-1+PNT
 3  243   938                      CASE
 4  244   939                      : SPIN(JTEMP).LT.RESRD(ITEMP) :
 4  245   940                         BCTTCM=BCTTOM-1
 4  245   941                         PNT=9999
 4  245   942                      : SPIN(JTEMP).GT.RESRD(ITEMP) :
 4  246   943                         TOP=TOP+1
 4  246   944                         PNT=9999
 4  246   945                      : SPIN(JTEMP).EQ.RESRD(ITEMP) :
```

```
4  247  946                        PNT=PNT+1
4  247  947                     ENDCASE
3  248  948                  REPEAT
2  249  949                  IF PNT.NE.9999 THEN
3  250  950                     CASE
4  251  951                     : LENGTH.EQ.RESRC(I) :
4  252  952                        JTEMP=LENGTH+1+I
4  252  953                        TYPE=RESRC(JTEMP)
4  252  954                        RETURN
4  252  955                     : LENGTH.GT.RESRC(I) :
4  253  956                        TOP=TOP+1
4  253  957                     : LENGTH.LT.RESRC(I) :
4  254  958                        BOTTOM=BOTTOM-1
4  254  959                     ENDCASE
3  255  960                  ENDIF
2  256  961               ENDCASE
1  257  962            REPEAT
0  258  963            TYPE=FCTRN
0  258  964            RETURN
0  258  965         END
```

```
-1  259   966          SUBROUTINE SCAN1(CHAR,ECOND)
 0  260   967             //  LOOKS FOR CHAR AND IF FOUND, SPIN(TCEND+1)=CHAR //
 0  260   968             //  AND ECOND=FALSE. IF NOT FOUND UPTC IMAX OR SEMICOLON //
 0  260   969             //  ECOND=TRUE. //
 0  260   970             //            //
 0  260   971             IMPLICIT INTEGER(A-Z)
 0  260   972             LOGICAL ECOND
 0  260   973             COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  260   974             COMMON/STTYP2/COMT,CONT,FOTRN,BLANK,SEMI
 0  260   975             //            //
 0  260   976             ECOND =.FALSE.
 0  260   977             WHILE TCEND.LT.IMAX DO
 1  261   978                ITEMP=SPIN(TOEND+1)
 1  261   979                KTEMP=SPIN(TOEND)
 1  261   980                IF ITEMP.EQ.SEMI.AND.KTEMP.EQ.BLANK THEN
 2  262   981                   ECOND=.TRUE.
 2  262   982                   RETURN
 2  262   983                ENDIF
 1  263   984                IF CHAR.NE.SEMI THEN
 2  264   985                   IF ITEMP.EQ.CHAR THEN
 3  265   986                      RETURN
 3  265   987                   ENDIF
 2  266   988                ENDIF
 1  267   989                TCEND=TCEND+1
 1  267   990             REPEAT
 0  268   991             ECOND = .TRUE.
 0  268   992             RETURN
 0  268   993          END
```

```
-1  269   994          SUBROUTINE SCAN(PAT,LEN,ECOND)
 0  270   995             // THE SUBROUTINE COMPARES THE PATERN PAT WITH ALL TOKENS //
 0  270   996             // SURSUROUNDED BY BLANKS FROM SPIN(TOEND+1) ONWARDS AND //
 0  270   997             // RETURNS WITH TOEND AT THE OLD POSITION OF TOEND; WITH THE //
 0  270   998             // PATTERN STARTING AT TOEND+1.  IF NOT FOUND ECOND IS SET TRUE //
 0  270   999             //           //
 0  270  1000             IMPLICIT INTEGER(A-Z)
 0  270  1001             LOGICAL ECOND
 0  270  1002             COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  270  1003             COMMON/STTYP2/COMT,CONT,FOTRN,BLANK,SEMI
 0  270  1004             DIMENSION PAT(6)
 0  270  1005             //           //
 0  270  1006             TOBEG=TOEND+1
 0  270  1007             LOOP
 1  271  1008                K=TOEND+1
 1  271  1009                WHILE K.LE.IMAX.AND.SPIN(K).EQ.BLANK DO
 2  272  1010                   K=K+1
 2  272  1011                REPEAT
 1  273  1012                TOEND=K
 1  273  1013                CALL SCAN1(BLANK,ECOND)
 1  273  1014                IF ECOND THEN
 2  274  1015                   RETURN
 2  274  1016                ENDIF
 1  275  1017                IF LEN.NE.TOEND-K+1 THEN
 2  276  1018                   CYCLE
 2  276  1019                ENDIF
 1  277  1020                // COMPARE THE PATTERN WITH A TOKEN OF THE SAME LENGTH. //
 1  277  1021                FOR I=1 TO LEN DO
 2  278  1022                   J=K+I-1
 2  278  1023                   IF PAT(I).NE.SPIN(J) THEN
 3  279  1024                      EXIT
 3  279  1025                   ENDIF
 2  280  1026                   IF I.EQ.LEN THEN
 3  281  1027                      // PATTERN FOUND, RETURN WITH ECOND FALSE. //
 3  281  1028                      TOEND=TOEND-LEN
 3  281  1029                      RETURN
 3  281  1030                   ENDIF
```

```
2  282  1031                  REPEAT
1  283 - 1032               REPEAT
0  284  1033            END
```

```
-1  285  1034            SUBROUTINE SPRNT
 0  286  1035               IMPLICIT INTEGER(A-Z)
 0  286  1036               LOGICAL LBFLG,OVFL
 0  286  1037               COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  286  1038               COMMON/STTYP2/CONT,CONT,FCTRN,BLANK,SEMI
 0  286  1039               COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX
 0  286  1040               COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)
 0  286  1041               COMMON/NUMIO/CEE,PRTR,FFILE,CRDR
 0  286  1042               COMMON/TYP/TYPE
 0  286  1043               //            //
 0  286  1044               IF TYPE.EQ.CONT THEN
 1  287  1045                  BLNKS(6)=SPIN(6)
 1  287  1046               ENDIF
 0  288  1047               LOOP
 1  288  1048                  OVFL = .FALSE.
 1  289  1049                  LEN1=STEND-STBEG+1
 1  289  1050                  LEN2=SMAX-SSTRT+1
 1  289  1051                  SEND=STEND
 1  289  1052                  IF LEN1.GT.LEN2 THEN
 2  290  1053                     OVFL=.TRUE.
 2  290  1054                     SEND=STBEG+LEN2-1
 2  290  1055                  ENDIF
 1  291  1056                  FIL=LEN2-LEN1+7
 1  291  1057                  WRITE(PRTR,10) NESTNO,BLKNO,STMNO,(BLNKS(I),I=1,SSTRT),
 1  291  1058           1      (SPIN(I),I=STBEG,SEND),(BLNKS(I),I=7,FIL),(SPIN(I),I=73
 1  291  1059           2      ,80)
 1  291  1060     10            FORMAT(1H ,I5,I5,I6,4X,111A1)
 1  291  1061                  FOR I=1 TO 6 DO
 2  292  1062                     BLNKS(I)=BLANK
 2  292  1063                  REPEAT
 1  293  1064                  STBEG=SEND+1
 1  293  1065               UNTIL .NOT.OVFL REPEAT
 0  294  1066               RETURN
 0  294  1067            END
```

```
-1  295  1068          SUBROUTINE FHAND
 0  296  1069             //   COMPLETES FORTRAN STATEMENTS AND OUTPUTS THE FORTRAN. //
 0  296  1070             //            //
 0  296  1071             IMPLICIT INTEGER(A-Z)
 0  296  1072             LOGICAL LBFLG,ECOND
 0  296  1073             COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  296  1074             COMMON/STTYP2/COMT,CONT,FOTRN,BLANK,SEMI
 0  296  1075             COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)
 0  296  1076             COMMON/TYP/TYPE
 0  296  1077             //            //
 0  296  1078             LOOP
 1  297  1079                CALL SCAN1 (SEMI,ECOND)
 1  297  1080                IF TYPE.EQ.CONT THEN
 2  298  1081                   FBUF(6)=SPIN(6)
 2  298  1082                ENDIF
 1  299  1083                CALL FOUT(SPIN,TOBEG,TOEND)
 1  299  1084                IF ECOND THEN
 2  300  1085                   EXIT
 2  300  1086                ENDIF
 1  301  1087                TOEND=TOEND+1
 1  301  1088                IF SPIN(TOEND+1).NE.SEMI THEN
 2  302  1089                   EXIT
 2  302  1090                ENDIF
 1  303  1091                TOBEG=TOEND+1
 1  303  1092                TOEND=TOBEG
 1  303  1093             REPEAT
 0  304  1094             CALL FPRNT
 0  304  1095             RETURN
 0  304  1096          END
```

```
-1  305  1097            SUBRCUTINE FOUT(PAT,STRT,END)
 0  306  1098               // THE VALUE OF FSTRT WILL USUALLY BE 7 AND FMAX WILL BE 72. //
 0  306  1099               //  FPTR SHCULD BE AT FSTRT.  BLNKLN IS A FLAG TO INDICATE A //
 0  306  1100               // TO INDICATE A SC FAR BLANK LINE. //
 0  306  1101               //          //
 0  306  1102               IMPLICIT INTEGER(A-Z)
 0  306  1103               LCGICAL LBFLG,BLNKLN
 0  306  1104               CCMMCN/CENTRL/STACK(200),LABEL(200),TCP,LBMAX,LBPTR,STFRM(6),
 0  306  1105         1     TPITR,NEUTR,BKWRC,FRWRD,NEW,CLD,PULCUT,STMAX,NUMB
 0  306  1106               CCMMCN/FRTINF/FBLF(80),FPTR,FMAX,FSTRT,BLNKS(131)
 0  306  1107               CCMMCN/STTYP2/COMT,CCNT,FCTRN,BLANK,SEMI
 0  306  1108               COMMCN/NMNIC/CEE,PRTR,FFILE,CRDR
 0  306  1109               CCMMCN/TYP/TYPE
 0  306  1110               CCMMCN/FORPRT/BLNKLN
 0  306  1111               DIMENSION PAT(1)
 0  306  1112               DATA NINE/1H9/
 0  306  1113               //          //
 0  306  1114               BLNKLN = .FALSE.
 0  306  1115               FCR J=STRT TC END DO
 1  307  1116                  IF FPTR.GT.FMAX THEN
 2  308  1117                     CALL FPRNT
 2  308  1118                     FCR I=1 TO 5 DC
 3  309  1119                        FBUF(I)=LABEL(I)
 3  309  1120                     REPEAT
 2  310  1121                     IF TYPE.EQ.CCMT THEN
 3  311  1122                        FBUF(1)=CEE
 3  311  1123                     ENDIF
 2  312  1124                     JJ=J
 2  312  1125                     WHILE PAT(JJ).EQ.BLANK DO
 3  313  1126                        IF JJ.EC.END THEN
 4  314  1127                           BLNKLN=.TRUE.
 4  314  1128                           RETURN
 4  314  1129                        ENDIF
 3  315  1130                        JJ=JJ+1
 3  315  1131                     REPEAT
 2  316  1132                     FBUF(6)=NINE
 2  316  1133                  ENDIF
```

```
1  317  1134                FBUF(FPTR)=PAT(J)
1  317  1135                FPTR=FPTR+1
1  317  1136             REPEAT
0  318  1137             RETURN
0  318  1138          END
```

```
-1  319  1139            SUBROUTINE FPRNT
 0  320  1140               // THE SUBROUTINE PRINTS THE TRANSLATED FORTRAN //
 0  320  1141               IMPLICIT INTEGER(A-Z)
 0  320  1142               LOGICAL BLNKLN
 0  320  1143               COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  320  1144               COMMON/NMNIC/CEE,PRTR,FFILE,CRDR
 0  320  1145               COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX
 0  320  1146               COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)
 0  320  1147               COMMON/FORPRT/BLNKLN
 0  320  1148               //              //
 0  320  1149               IF FPTR.EQ.FSTRT THEN
 1  321  1150                  RETURN
 1  321  1151               ENDIF
 0  322  1152               IFF=FSTRT-1
 0  322  1153               IF IFF.GE.7 THEN
 1  323  1154                  FOR I=7 TO IFF DO
 2  324  1155                     FBUF(I)=BLNKS(I)
 2  324  1156                  REPEAT
 1  325  1157               ENDIF
 0  326  1158               IF FPTR.LE.FMAX THEN
 1  327  1159                  FOR I=FPTR TO FMAX DO
 2  328  1160                     FBUF(I)=BLNKS(I)
 2  328  1161                  REPEAT
 1  329  1162               ENDIF
 0  330  1163               FOR I=73 TO 75 DO
 1  331  1164                  FBUF(I)=SPIN(I)
 1  331  1165               REPEAT
 0  332  1166               IF .NOT.BLNKLN THEN
 1  333  1167                  WRITE(FFILE,10)(FBUF(I),I=1,75),STMNO
 1  333  1168      10          FORMAT(75A1,I5)
 1  333  1169               ENDIF
 0  334  1170               FPTR=FSTRT
 0  334  1171               RETURN
 0  334  1172            END
```

```
-1  336  1173          INTEGER FUNCTION ISNUM(Q)
 0  337  1174             // THE FUNCTION CHECKS TO SEE IF Q IS THE ALPHANUMERIC          //
 0  337  1175             // REPRESENTATION OF A NUMBER AND RETURNS 1 IF IT IS AND 0 IF   //
 0  337  1176             // IT IS NOT. //
 0  337  1177             //           //
 0  337  1178             IMPLICIT INTEGER(A-Z)
 0  337  1179             DIMENSION NUMBER(10)
 0  337  1180             DATA NUMBER/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
 0  337  1181             //           //
 0  337  1182             FOR I=1 TO 10 DO
 1  338  1183                IF Q.EQ.NUMBER(I) THEN
 2  339  1184                   ISNUM=1
 2  339  1185                   RETURN
 2  339  1186                ENDIF
 1  340  1187             REPEAT
 0  341  1188             ISNUM=0
 0  341  1189             RETURN
 0  341  1190          END
```

```
-1  342  1191          INTEGER FUNCTION CONV(DIGIT)
 0  343  1192             // CONVERTS THE DIGIT INTO ITS CHARACTER REPRESENTATION. //
 0  343  1193             //                                                      //
 0  343  1194             IMPLICIT INTEGER(A-Z)
 0  343  1195             DIMENSION CVNUM(10)
 0  343  1196             DATA CVNUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
 0  343  1197             CONV=CVNUM(DIGIT+1)
 0  343  1198             RETURN
 0  343  1199          END
```

```
-1  344  1200            INTEGER FUNCTION GENLAB(GEN)
 0  345  1201               //  GENERATE A LABEL UNCONDITIONALLY IF GEN IS 1 ELSE //
 0  345  1202               //  GENERATE THE LABEL IF LBFLG=0, IF LABEL FLAG //
 0  345  1203               //  EQUAL 1 AND GEN EQUAL 0 THEN USE THE EXISTING //
 0  345  1204               //  LABEL. NUMB IS INITIALIZED TO 100000. THIS IS AN //
 0  345  1205               //  INTEGER FUNCTION RETURNING A POINTER TO THE LABEL. //
 0  345  1206               //           //
 0  345  1207               IMPLICIT INTEGER(A-Z)
 0  345  1208               LOGICAL LBFLG
 0  345  1209               COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  345  1210               COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),
 0  345  1211         1     TPITR,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB
 0  345  1212               //           //
 0  345  1213               IF .NOT.LBFLG.OR.GEN.EQ.1 THEN
 1  346  1214                  NUMB=NUMB-1
 1  346  1215                  N=NUMB
 1  346  1216                  FOR I=1 TO 5 DO
 2  347  1217                     M=N/10
 2  347  1218                     //  SHIFT RIGHT  //
 2  347  1219                     NR=N-(10*M)
 2  347  1220                     //  REMAINDER  //
 2  347  1221                     HOLD=5-I+LBPTR
 2  347  1222                     LABEL(HOLD)=CONV(NR)
 2  347  1223                     //  LBPTR IS LEFT AT FIRST BLANK SPACE.  //
 2  347  1224                     N=M
 2  347  1225                  REPEAT
 1  348  1226                  GENLAB=LBPTR
 1  348  1227                  //  NEW LABEL PTR.  //
 1  348  1228                  LBPTR=LBPTR+5
 1  348  1229                  //  POINT TO NEW BLANK LOCATION.  //
 1  348  1230                  IF LBPTR.GT.LBMAX THEN
 2  349  1231                     CALL ERROR(8,0,0,5,0,0)
 2  349  1232                     //  NESTING TOO DEEP, TOO MANY LABELS PROGRAM TERMINATED. //
 2  349  1233                  ENDIF
 1  350  1234               ELSE
 1  351  1235                  FOR I=1 TO 5 DO
 2  352  1236                     HOLD=LBPTR+I-1
```

```
2  352  1237                    LABEL(HOLD)=SPIN(I)
2  352  1238                 REPEAT
1  353  1239                 GENLAB=LBPTR
1  353  1240                 LBPTR=LBPTR+5
1  353  1241                 IF LBPTR.GT.LBMAX THEN
2  354  1242                    CALL ERROR(8,0,0,5,0,0)
2  354  1243                 ENDIF
1  355  1244                 //  NESTING TOO DEEP, TOO MANY LABELS PROGRAM TERMINATED. //
1  355  1245              ENDIF
0  356  1246              RETURN
0  356  1247           END
```

```
-1  357  1248            SUBROUTINE CHECK(STYPE,ECOND)
 0  358  1249               IMPLICIT INTEGER(A-Z)
 0  358  1250               LOGICAL ECOND,DUMM,LBFLG
 0  358  1251               COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  358  1252               COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)
 0  358  1253               COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX
 0  358  1254               COMMON/OTPAT/CGOTO(9),CONTI(8),EQL(1),GOTO(6),IFNOT(9),LEO(5),
 0  358  1255         1     MINUS(1),ONE(1),PLU(1),RSTAR(2)
 0  358  1256               COMMON/STTYP1/CCASE,COLON,DBLSL,EELSE,EEND,EOJ,EEXIT,FFOR,IIF,
 0  358  1257         1     LLOOP,RREPT,UNTL,WHLE,SUBFN,RINLO,NCYCLE,ENDCAS,EENDIF
 0  358  1258               COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),
 0  358  1259         1     TPITR,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB
 0  358  1260               COMMON/INPAT/CCLN(1),ELZ(4),DBSLH(2),DO0(2),BYE(2),THENN(4),TOO(2)
 0  358  1261               COMMON/TABLES/TRES(23),RESRD(200)
 0  358  1262               COMMON/KEYWD/EEEND(3),EEEXT(4),CCYCLE(5),UUNTL(5),CASS(4),IIIF(2),
 0  358  1263         1     ITERAT(9),ENCASS(7),ENIF(5),REPT(6),CCCLCN(5)
 0  358  1264               //                //
 0  358  1265               ECOND=.FALSE.
 0  358  1266               IF TOP.EQ.0 THEN
 1  359  1267                  CALL ERROR(4,0,0,3,0,0)
 1  359  1268                  // ERROR---STACK EMPTY, STATEMENT IGNORED. //
 1  359  1269                  ECOND=.TRUE.
 1  359  1270                  RETURN
 1  359  1271               ENDIF
 0  360  1272               STTOP=STACK(TOP)
 0  360  1273               CASE
 1  361  1274               : STYPE.EQ.CCLCN :
 1  362  1275                  IF STTOP.NE.CCASE THEN
 2  363  1276                     CALL ERROR(2,CASS,4,4,CCOLCN,5)
 2  363  1277                     // ERROR---MISSING CASE STATEMENT, COLON IGNORED. //
 2  363  1278                     TOEND=TOEND+1
 2  363  1279                     ECOND=.TRUE.
 2  363  1280                  ENDIF
 1  364  1281               : STYPE.EQ.EELSE :
 1  365  1282                  IF STTOP.NE.IIF THEN
 2  366  1283                     CALL ERROR(2,IIIF,2,4,ELZ,4)
 2  366  1284                     // ERROR---MISSING IF STATEMENT, ELSE IGNORED. //
```

```
2  366  1285                  ECOND=.TRUE.
2  366  1286               ENDIF
1  367  1287            : STYPE.EQ.UNTL :
1  368  1288               IF STTOP.NE.LLOOP THEN
2  369  1289                  CALL ERROR(5,ITERAT,9,4,UUNTL,5)
2  369  1290                  // ERROR---MISSING LOOP STATEMENT, UNTIL IGNORED. //
2  369  1291                  CALL SCAN(REPT,3,DUMM)
2  369  1292                  IF DUMM THEN
3  370  1293                     TOEND=TOEND+1
3  370  1294                  ELSE
3  371  1295                     TOEND=TOEND+6
3  371  1296                  ENDIF
2  372  1297                  ECOND=.TRUE.
2  372  1298               ENDIF
1  373  1299            : STYPE.EQ.EEND :
1  374  1300               IF STTOP.NE.SUBFN THEN
2  375  1301                  // UNSTACK UNTIL A SUBFN IS FOUND ON THE STACK. //
2  375  1302                  TTOP=TOP
2  375  1303                  LOOP
3  376  1304                     TTOP=TTOP-STFRM(STTOP)
3  376  1305                     IF TTOP.LE.0 THEN
4  377  1306                        CALL ERROR(3,EEEND,3,4,EEEND,3)
4  377  1307                        // ERROR---TOO MANY ENDS, END IGNORED. //
4  377  1308                        ECOND=.TRUE.
4  377  1309                        RETURN
4  377  1310                     ENDIF
3  378  1311                     STTOP=STACK(TTOP)
3  378  1312                  UNTIL STTOP.EQ.SUBFN REPEAT
2  379  1313                  TOP=TTOP
2  379  1314                  CALL ERROR(7,0,0,6,0,0)
2  379  1315                  // PREMATURE END SUBPROGRAM CLOSED. //
2  379  1316                  FSTRT=7
2  379  1317                  SSTRT=7
2  379  1318               ENDIF
1  380  1319            : STYPE.EQ.ENDCAS :
1  381  1320               IF STTOP.NE.CCASE THEN
2  382  1321                  CALL ERROR(3,ENCASS,7,4,ENCASS,7)
2  382  1322                  // ERROR---TOO MANY ENDCASES, ENDCASE IGNORED. //
```

```
2  382  1323                  ECOND=.TRUE.
2  382  1324               ENDIF
1  383  1325            : STYPE.EQ.EENDIF :
1  384  1326               IF STTOP.NE.IIF THEN
2  385  1327                  CALL ERROR(3,ENIF,5,4,ENIF,5)
2  385  1328                  // ERROR---TOO MANY ENDIFS, ENDIF IGNORED. //
2  385  1329                  ECOND=.TRUE.
2  385  1330               ENDIF
1  386  1331            : STYPE.EQ.RREPT :
1  387  1332               IF STTOP.EQ.FFOR.OR.STTOP.EQ.LLOOP.OR.STTOP.EQ.WHLE THEN
2  388  1333                  RETURN
2  388  1334               ELSE
2  389  1335                  CALL ERROR(5,ITERAT,9,4,REPT,6)
2  389  1336                  // ERROR---NO ITERATIVE STATEMENT FOUND, REPEAT IGNORED. //
2  389  1337                  ECOND=.TRUE.
2  389  1338               ENDIF
1  390  1339            : STYPE.EQ.EEXIT :
1  391  1340               IF TPITR.EQ.0 THEN
2  392  1341                  CALL ERROR(5,ITERAT,9,4,EEXT,4)
2  392  1342                  // ERROR---NO ITERATIVE STATEMENT, EXIT IGNORED. //
2  392  1343                  ECOND=.TRUE.
2  392  1344               ENDIF
1  393  1345            : STYPE.EQ.NCYCLE :
1  394  1346               IF TPITR.EQ.0 THEN
2  395  1347                  CALL ERROR(5,ITERAT,9,4,CCYCLE,5)
2  395  1348                  // ERROR---NO ITERATIVE STATEMENT, CYCLE IGNORED. //
2  395  1349                  ECOND=.TRUE.
2  395  1350               ENDIF
1  396  1351            ENDCASE
0  397  1352            RETURN
0  397  1353         END
```

```
 -1  398  1354            SUBROUTINE ERROR(CODE1,PARM1,LEN1,CODE2,PARM2,LEN2)
  0  399  1355               //  CODE1 IS ERROR MESSAGE CODE //
  0  399  1356               //  CODE2 IS THE ACTION TAKEN //
  0  399  1357               //  PARM1 AND PARM2 MODIFY THE TWO //
  0  399  1358               //  LEN1 AND LEN2 ARE THE LENGTH OF MODIFYING PATTERN //
  0  399  1359               //           //
  0  399  1360               IMPLICIT INTEGER(A-Z)
  0  399  1361               COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
  0  399  1362               COMMON/STTYP2/COMT,CONT,FOTRN,BLANK,SEMI
  0  399  1363               COMMON/NMNIC/ZERO,PRTR,FFILE,CRDR
  0  399  1364               DIMENSION PARM1(1),PARM2(1),ERPAT(45,8),POSN(8),LEN(8),
  0  399  1365         1     ACPAT(35,6),ALEN(6),APSN(6)
-----BEGINNING OF FORTRAN-----  F
                           DATA ERPAT/1HL,1HA,1HB,1HE,1HL,1HL,1HI,1HN,1HG,1H ,1HT,1HH,1HE,
                          1  1H ,1H ,1H ,1H ,1H ,1F ,1H ,1HH,1HA,1HM,1HP,1HE,1HR,1HS,
                          1  1HP,1HR,1HC,1HG,1HR,1HA,1HM,1H ,1HC,1HL,1HA,1HR,1HI,1HT,1HY,1H.,
                          1  1H ,1H ,
                          2  1HM,1HI,1HS,1HS,1HI,1HN,1HG,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,
                          2  1H ,1H ,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                          2  1H ,1H ,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                          3  1HD,1HA,1HN,1HG,1HL,1HI,1HN,1HG,1H ,1H ,1F ,1H ,1H ,1H ,1H ,
                          3  1H ,1H ,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                          3  1H ,1H ,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                          4  1HS,1HT,1HA,1HC,1HK,1F ,1HE,1HM,1HP,1HT,1HY,1H.,1H ,1H ,1H ,
                          4  1H ,1F ,1H ,1H ,1H ,1F ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,1H ,1H ,
                          4  1H ,1H ,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,
                          5  1HN,1HO,1H ,1HC,1HO,1HR,1HR,1HE,1HS,1HP,1HO,1HN,1HD,1HI,1HN,1HG,
                          5  1H ,1H ,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,1H ,1HS,1HT,1HM,1HT,1H.,
                          5  1H ,1H ,1H ,1H ,1H ,1F ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                          6  1HS,1HT,1HA,1HC,1HK,1F ,1HF,1HU,1HL,1HL,
                          7  35*1H ,1HP,1HR,1HE,1HM,1HA,1HT,1HU,1HR,1HE,1H ,1HE,1HN,1HD,
                          8  32*1H ,1HT,1HC,1HC,1H ,1HM,1HA,1HN,1HY,1F ,1HL,1HA,1HB,1HE,1HL,
                          8  1HS,30*1H /
                             DATA LEN/5,6,6,0,9,0,0,0/
                             DATA POSN/15,9,10,0,18,0,0,0/
                             DATA ACPAT/1HS,1HU,1HP,1HP,1HL,1HI,1HE,1HD,1H ,1HA,1HT,
                          1  1HE,1HN,1HD,1H ,1HO,1HF,1H ,1HS,1HT,1HM,1HT,1H.,
```

```
                                1  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                                2  1HL,1HA,1HB,1HE,1HL,1H ,1HD,1HE,1HL,1HE,1HT,1HE,1HD,
                                2  1HF,1HR,1HC,1HM,1H ,1HF,1HC,1HR,1HT,1HR,1HA,1HN,1H ,
                                2  1HO,1HU,1HT,1HP,1HU,1HT,1H.,1H ,1H ,
                                3  1HS,1HT,1HA,1HT,1HE,1HM,1HE,1HN,1HT,1H ,
                                3  1HS,1HK,1HI,1HP,1HP,1HE,1HD,1H.,1H ,1H ,1H ,1H ,1H ,
                                3  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                                4  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1HI,1HG,1HN,1HO,1HR,1HE,1HD,
                                4  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                                4  1H ,1H ,1H ,1H ,1H ,1H ,
                                5  1HP,1HR,1HC,1HG,1HR,1HA,1HM,1H ,1HT,1HE,1HR,1HM,1HI,1HN,1HA,
                                5  1HT,1HE,1HD,17*1H ,
                                6  1HS,1HU,1HB,1HP,1HR,1HO,1HG,1HR,1HA,1HM,1H ,1HC,1HL,1HO,1HS,
                                6  1HE,1HD,18*1H /
                                   DATA ALEN/0,0,0,6,0,0/
                                   DATA APSN/0,0,0,1,0,0/
-----END OF FORTRAN-----           F
A-45
  0  399  1366                        //              //
  0  399  1367                        WRITE(PRTR,2222) SPIN
  0  399  1368      2222              FORMAT(26H    ++++++OFFENDING CARD: ,80A1)
  0  399  1369                        LENTH=LEN(CODE1)
  0  399  1370                        ALNTH=ALEN(CODE2)
  0  399  1371                        POS=POSN(CODE1)
  0  399  1372                        APOS=APSN(CODE2)
  0  399  1373                        FOR I=1 TO LEN1 DO
  1  400  1374                           IT=POS+I-1
  1  400  1375                           ERPAT(IT,CODE1)=PARM1(I)
  1  400  1376                        REPEAT
  0  401  1377                        IF LENTH.GT.LEN1 THEN
  1  402  1378                           PO=POS+LEN1
  1  402  1379                           PE=POS+LENTH-1
  1  402  1380                           FOR I=PO TO PE DO
  2  403  1381                              ERPAT(I,CODE1)=BLANK
  2  403  1382                           REPEAT
  1  404  1383                        ENDIF
  0  405  1384                        FOR I=1 TO LEN2 DO
  1  406  1385                           IT=APOS+I-1
  1  406  1386                           ACPAT(IT,CODE2)=PARM2(I)
```

```
1  406  1387                REPEAT
0  407  1388                IF ALENTH.GT.LEN2 THEN
1  408  1389                   PO=APOS+LEN2
1  408  1390                   PE=APOS+ALNTH-1
1  408  1391                   FOR I=PO TO PE DO
2  409  1392                      ACPAT(I,CODE2)=BLANK
2  409  1393                   REPEAT
1  410  1394                ENDIF
0  411  1395                WRITE(PRTR,10)   (ERPAT(I,CODE1),I=1,45)
0  411  1396                WRITE (PRTR,10)     (ACPAT(I,CODE2),I=1,35)
0  411  1397          10    FORMAT(12H     +++++++,80A1)
0  411  1398                RETURN
0  411  1399             END
```

```
-1  412  1400            SUBROUTINE TYHAND(TYPE,EEEXT,CCYCLE,EEXIT)
 0  413  1401               IMPLICIT INTEGER(A-Z)
 0  413  1402               LOGICAL LBFLG
 0  413  1403               COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG
 0  413  1404               COMMON/SPRINF/SSTRT,INCR,STMND,BLKND,NESTND,INDFLG,CURFLG,SMAX
 0  413  1405               COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)
 0  413  1406               COMMON/OTPAT/GOTO(9),CONTI(9),EQL(1),GOTO(6),IFNOT(9),LEQ(5),
 0  413  1407         1     MINUS(1),ONE(1),PLL(1),RSTAR(2)
 0  413  1408               COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),
 0  413  1409         1     TPITR,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB
 0  413  1410               COMMON/STTYP2/COMT,CONT,FOTRN,BLANK,SEMI
 0  413  1411               //            //
 0  413  1412               IF LBFLG THEN
 1  414  1413                  IF TYPE.EQ.EEXIT THEN
 2  415  1414                     CALL ERROR(1,EEEXT,4,2,0,0)
 2  415  1415                     // ERROR--LABELLING THE EXIT HAMPERS PROGRAM CLEANLINESS-- //
 2  415  1416                     // --DELETED FROM THE FORTRAN OUTPUT. //
 2  415  1417                  ELSE
 2  416  1418                     CALL ERROR(1,CCYCLE,5,2,0,0)
 2  416  1419                     // ERROR---LABELLING THE CYCLE HAMPERS PROGRAM  //
 2  416  1420                     //  CLEANLINESS, DELETED FROM FORTRAN OUTPUT.  //
 2  416  1421                  ENDIF
 1  417  1422               ENDIF
 0  418  1423               //  CHECK IF A NUMERIC LABEL FOLLOWS.  //
 0  418  1424               K=TOEND+1
 0  418  1425               LOOP
 1  419  1426                  IF K.GT.IMAX THEN
 2  420  1427                     I=0
 2  420  1428                     EXIT
 2  420  1429                  ENDIF
 1  421  1430                  IF SPIN(K).NE.BLANK THEN
 2  422  1431                     I=ISNUM(SPIN(K))
 2  422  1432                     EXIT
 2  422  1433                  ENDIF
 1  423  1434                  K=K+1
 1  423  1435               REPEAT
 0  424  1436               IF I.EQ.0 THEN
```

```
1  425  1437                  FOR I=1 TO 6 DO
2  426  1438                     FBUF(I)=BLANK
2  426  1439                  REPEAT
1  427  1440                  IF TYPE.EQ.'EXIT THEN
2  428  1441                     K=STACK(TPITR-1)
2  428  1442                  ELSE
2  429  1443                     K=STACK(TPITR-2)
2  429  1444                  ENDIF
1  430  1445                  CALL FOUT(GOTO,1,6)
1  430  1446                  CALL FOUT(LABEL,K,K+4)
1  430  1447                  CALL FPRNT
1  430  1448               ELSE
1  431  1449                  TOEND=K
1  431  1450                  CALL SCAN1(BLANK,ECOND)
1  431  1451                  WRITE(PRTR,4999)
1  431  1452     4999         FORMAT( '++++++++++LABELLED EXIT-CYCLE NOT IMPLEMENTED')
1  431  1453               ENDIF
0  432  1454               STEND=TOEND
0  432  1455               INDFLG=NEUTR
0  432  1456               CURFLG=OLD
0  432  1457               RETURN
0  432  1458            END

     END OF SPARKS PROGRAM.....
```

APPENDIX B

FORTRAN TRANSLATION

OF THE

SPARKS PREPROCESSOR

```
      C        // MAIN PROGRAM                                                  0
      C        //                                                               1
0001           IMPLICIT INTEGER(A-Z)                                           2
0002           LOGICAL LBFLG,ECEND,ECCNC1;BAC                                  3
0003           COMMON/INPUT/SPIN(80),TCBEG,TCEND,STBEG,STEND,IMAX,LBFLG        4
0004           COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX  5
0005           COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)               6
0006           COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),   7
     1         TPITP,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB               8
0007           COMMON/CTPAT/CGOTO(9),CONTI(8),EQL(1),GOTO(6),IFNOT(9),LEO(5),  9
     1         MINUS(1),ONE(1),PLU(1),RSTAR(2)                                10
0008           COMMON/INPAT/CCLN(1),ELZ(4),OBSLH(2),DCO(2),BYE(2),THENN(4),TOO 11
     9         (2)                                                            11
0009           COMMON/STTYP1/CCASE,COLON,CBLSL,EELSE,EEND,EOJ,EEXIT,FFOR,IIF,  12
     1         LLOOP,RREPT,UNTL,WHLE,SUBFN,RINLC,NCYCLE,ENDCAS,EENDIF          13
0010           COMMON/STTYP2/COMT,CONT,FCTRN,BLANK,SEMI                        14
0011           COMMON/NMNIC/CEE,PRTR,FFILE,CRDR                                15
0012           COMMON/KEYWD/EEEND(3),EEXT(4),CCYCLE(5),UUNTL(5),CASS(4),IIIF(  16
     9         2),                                                            16
     1         ITERAT(9),ENCASS(7),ENIF(5),REPT(6),CCOLON(5)                  17
0013           COMMON/TABLES/TRES(23),RESRD(200)                               18
0014           COMMON/TYP/TYPE                                                 19
      C        //                                                              20
      C        // WRITE TITLES ETC.                                            21
0015           WRITE(PRTR,3)(SPIN(I),I=1,80)                                   22
0016         3 FORMAT(4X,4HNEST,2X,3HBLK,2X,4HSTNO,4X,30X,80A1)                23
0017           WRITE(PRTR,4)                                                   24
0018         4 FORMAT(/32X,'SPARKS TRANSLATION BY KSU SPARKS PREPROCESSOR ',   25
     1         'AS'/32X,'MODIFIED BY R. STROUD, M. CALHOUN, AND J. MARTIN.'/)  26
      C        //                                                              27
      C        // SET UP OPTIONS                                               28
      C        //                                                              29
0019           IMAX=72                                                         30
0020           SMAX=100                                                        31
0021           FMAX=72                                                         32
0022           INCR=3                                                          33
```

```
0023                FSTRT=7+INCR                                                34
0024                SSTRT=7+INCR                                                35
          C         // EXTRACT OPTIONS GIVEN ON THE CONTROL CARD AND CHANGE ABOVE 36
          C         // DEFAULTS ACCORDINGLY---THIS COULD USE DETNX               37
          C         // INITIALISE THE PROGRAM RELATED VARIABLES.                 38
0025                STACK(1)=SUBFN                                              39
0026                TOP=1                                                       40
0027                TPITR=0                                                     41
0028                LBPTR=6                                                     42
0029                TOEND=IMAX+1                                                43
0030                BLKNO=0                                                     44
0031                STMNO=0                                                     45
0032                NESTNO=0                                                    46
          C         // START MAIN LOOP                                          47
0033      99999     CONTINUE                                                    48
0034                   FPTR=FSTRT                                               49
0035                   CALL DETNX(TYPE)                                         50
          C            // PROCESS STATEMENTS ACCORDING TO TYPE                  51
          C            // 1. TRANSLATE AND PRINT FORTRAN.                       52
          C            // 2. SET INDENTATION FLAG---INDFLG.                     53
          C            // 3. SET CURFLG NEW-OLD--CURRENT STATEMENT TO BE PRINTED IN 54
          C            // OLD OR NEW INDENTATION.                               55
```

```
      C            //                                                          56
0036                IF(.NOT.( TYPE.EQ.RINLC )) GO TO 99996                    57
0037                   A=TOBEG                                                58
0038                   B=TOEND                                                59
0039                   CALL DETNX(TYPE)                                       60
      C                // IF THE SECOND TOKEN IS NOT FUNCTION THEN PASS AS FORTR  61
      C     $          AN.                                                    61
0040                   IF(.NOT.( TYPE.NE.SUBFN )) GO TO 99995                 62
0041                      TYPE=FOTRN                                          63
0042          99995    CONTINUE                                               64
0043                   TOEND=B                                                65
0044                   TOBEG=A                                                66
0045          99996    CONTINUE                                               67
0046                   STBEG=TOBEG                                            68
      C                // THE MAIN CASE STATEMENT                             69
0047                   IF(.NOT.( TYPE.EQ.FOTRN )) GO TO 99993                 71
0048                      CALL FHAND                                          72
0049                      STEND=TOEND                                         73
0050                      CURFLG=OLD                                          74
0051                      INDFLG=NEUTR                                        75
0052                   GO TO 99994                                            76
0053          99993    IF(.NOT.( TYPE.EQ.DBLSL )) GO TO 99992                 76
      C                   // LOOK FOR A DOUBLE SLASH--SCAN IS NOT USED SINCE IT STO  77
      C     $             PS                                                  77
      C                   // ON A SEMICOLON.                                  78
0054          99991       CONTINUE                                           79
0055                         IF(.NOT.( TOEND.GE.IMAX-1 )) GO TO 99988         80
0056                            CALL ERROR(2,DBSLH,2,1,0,0)                   81
      C                         // ERROR--MISSING DOUBLE SLASH SUPPLIED AT END OF S  82
      C     9                   TATEMENT.                                     82
0057                            GO TO 99989                                   83
0058          99988          CONTINUE                                         84
0059                         IF(.NOT.( SPIN(TOEND+1).EQ.DBSLH(1).AND.SPIN(TOEND+2).  85
                   $         EQ.DBSLH(1) )) GO TO 99987                      85
0060                            GO TO 99989                                   86
0061          99987          CONTINUE                                         87
```

```
0062                        TOEND=TOEND+1                                    88
0063          99990       GO TO 99991                                      89
0064          99989       CONTINUE                                         89
0065                      FBUF(1)=CEE                                      90
              C           // THE TYPE IS CHANGED SO THAT FOUT CAN CHECK IT AND PUT   91
              C     9     A C IN                                           91
              C           // COLUMN ONE FOR LONG COMMENTS.                 92
0066                      TYPE=COMT                                        93
0067                      CALL FOUT(SPIN,TOBEG,TOEND)                      94
0068                      CALL FPRNT                                       95
0069                      INDFLG=NEUTR                                     96
0070                      CLRFLG=CLD                                       97
0071                      TOEND=TOEND+2                                    98
0072                      STEND=TOEND                                      99
0073                    GO TO 99994                                       100
0074          99992     IF(.NOT.( TYPE.EQ.IIF )) GO TO 99986              100
0075                      LAB=GENLAB(1)                                    101
0076                      CALL SCAN(THENN,4,ECOND)                         102
0077                      IF(.NOT.( ECOND )) GO TO 99985                   103
              C             // ABSENSE OF THEN ASSUMED TO INDICATE FORTRAN IF.   104
0078                        TOBEG=TOBEG-2                                  105
0079                        CALL FHANG                                     106
```

```
0080                      GO TO 99984                                          107
0081          99985     CONTINUE                                             107
              C           // OUTPUT: IF(.NOT.COND) GO TO LAB                  108
0082                      CALL FOUT(IFNCT,1,9)                                 109
0083                      CALL FOUT(SPIN,TOBEG,TCEND)                          110
0084                      CALL FOUT(CGOTO,1,9)                                 111
0085                      CALL FOUT(LABEL,LAB,LAB+4)                           112
0086                      CALL FPRNT                                           113
0087                      STACK(TOP+1)=LAB                                     114
0088                      STACK(TOP+2)=IIF                                     115
0089                      TOP=TOP+2                                            116
0090                      IF(.NOT.( TOP.GT.STMAX )) GO TO 99983                117
0091                         CALL ERROR(6,0,0,5,0,0)                           118
0092          99983       CONTINUE                                             119
              C           //  STACK OVERFLOWED, PROGRAM TERMINATED.            120
0093                      TCEND=TCEND+4                                        121
0094                      IF(.NOT.( FSTRT+INCR.LT.IMAX )) GO TO 99982          122
0095                         FSTRT=FSTRT+INCR                                  123
0096          99982       CONTINUE                                             124
0097          99984     CONTINUE                                             125
0098                    STEND=TCEND                                            126
0099                    CLRFLG=OLD                                             127
0100                    INDFLG=FRWRD                                           128
0101                  GO TO 99994                                              129
0102          99986   IF(.NOT.( TYPE.EQ.EENDIF )) GO TO 99981                 129
0103                    CALL CHECK(EENDIF,BAD)                                 130
0104                    IF(.NOT.( BAD )) GO TO 99980                           131
0105                       GO TO 99998                                         132
0106          99980     CONTINUE                                             133
0107                    FSTRT=FSTRT-INCR                                       134
0108                    FPTR=FSTRT                                             135
0109                    IF(.NOT.( LBFLG )) GO TO 99979                         136
              C           // LABELLED STATEMENT--GENERATE 'LAB CONTINUE'       137
0110                      CALL FOUT(CONTI,1,8)                                 138
0111                      CALL FPRNT                                           139
0112          99979     CONTINUE                                             140
```

```
0113                      LAB=STACK(TOP-1)                                    141
0114                      TOP=TOP-2                                           142
0115                      LBPTR=LBPTR-5                                       143
0116                       I=1                                                144
0117                      GO TO 99978                                         144
0118          99977        I= I+1                                             144
0119          99978       IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99975              144
0120                         L=LAB+I-1                                        145
0121                         FBUF(I)=LABEL(L)                                 146
0122          99976       GO TO 99977                                         147
0123          99975       CONTINUE                                            147
0124                      FBUF(6)=BLANK                                       148
0125                      CALL FOUT(CONTI,1,8)                                149
0126                      CALL FPRNT                                          150
              C           //   GENERATE A LAB CONTINUE.                       151
0127                      STEND=TOEND                                         152
0128                      INDFLG=BKWRD                                        153
0129                     CURFLG=NEW                                           154
0130                   GO TO 99994                                           155
0131          99981    IF(.NOT.( TYPE.EQ.EELSE )) GO TO 99974                155
0132                      CALL CHECK(EELSE,BAD)                               156
0133                      IF(.NOT.( BAD )) GO TO 99973                        157
```

```
0134                        GO TO 99998                                          158
0135        99973         CONTINUE                                             159
0136                      FSTRT=FSTRT-INCR                                     160
0137                      IF(.NOT.( LBFLG )) GO TO 99972                       161
0138                        CALL ERROR(1,ELZ,4,2,0,0)                          162
            C               // ERROR--LABELLING THE ELSE HAMPERS PROGRAM CLEANLINE 163
            C     9         SS--                                               163
            C               // DELETED FROM THE FORTRAN OUTPUT.                164
0139        99972         CONTINUE                                             165
0140                      LAB2=GENLAB(1)                                       166
0141                      LAB1=STACK(TOP-1)                                    167
0142                       I=1                                                 168
0143                      GO TO 99971                                          168
0144        99970          I= I+1                                              168
0145        99971         IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99968               168
0146                        FBUF(I)=BLANK                                      169
0147        99969         GO TO 99970                                          170
0148        99968         CONTINUE                                             170
0149                      CALL FOUT(GOTO,1,6)                                  171
0150                      CALL FOUT(LABEL,LAB2,LAB2+4)                         172
0151                      CALL FPRNT                                           173
            C             // GO TO LAB2                                        174
0152                       I=1                                                 175
0153                      GO TO 99967                                          175
0154        99966          I= I+1                                              175
0155        99967         IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99964               175
0156                        L=LAB1+I-1                                         176
0157                        FBUF(I)=LABEL(L)                                   177
0158        99965         GO TO 99966                                          178
0159        99964         CONTINUE                                             178
0160                      FBUF(6)=BLANK                                        179
0161                      CALL FOUT(CONTI,1,8)                                 180
0162                      CALL FPRNT                                           181
            C             // LAB1 CONTINUE                                     182
0163                      STACK(TOP-1)=LAB2                                    183
0164                      STEND=TOEND                                          184
```

```
0165                      INDFLG=NEUTR                                          185
0166                      CURFLG=PULOUT                                         186
0167                      IF(.NOT.( FSTRT+INCR.LT.IMAX )) GO TO 99963           187
0168                         FSTRT=FSTRT+INCR                                   188
0169           99963      CONTINUE                                              189
0170                    GO TO 99994                                             190
0171           99974    IF(.NOT.( TYPE.EQ.FFOR )) GO TO 99962                   190
0172                      IF(.NOT.( LBFLG )) GO TO 99961                        191
               C             //  LAB WILL HAVE IDENTIFYING LABEL IF ANY.       192
0173                         LAB=GENLAB(0)                                      193
0174                         CALL FOUT(CONTI,1,8)                               194
0175                         CALL FPRNT                                         195
0176           99961      CONTINUE                                              196
0177                      TCBEG=TCEND+1                                         197
0178                      CALL SCAN1(EQL,ECOND)                                 198
0179                      IF(.NOT.( ECOND )) GO TO 99960                        199
0180                         CALL ERROR(2,EQL,1,3,0,0)                          200
               C             //  ERROR--MISSING EQUAL-STATEMENT SKIPPED.       201
0181                         GO TO 99998                                        202
0182           99960      CONTINUE                                              203
0183                      VBLEA=TCBEG                                           204
0184                      VBLEB=TCEND                                           205
```

```
0185                    TCEND=TOEND+1                                                  206
0186                    CALL SCAN(TCC,2,ECCND)                                         207
0187                    IF(.NOT.( ECCND )) GC TO 99959                                 208
0188                       CALL ERRCR(2,TOC,2,3,0,0)                                   209
        C                  // ERROR--MISSING 'TC' - STATEMENT SKIPPED.                 210
0189                       TOEND=TCEND+1                                               211
0190                       GC TC 99998                                                 212
0191            99959   CCNTINUE                                                       213
0192                    TOEND=TOEND+2                                                  214
0193                    CALL FOUT(SPIN,VBLEA,TCEND-2)                                  215
0194                    CALL FPRNT                                                     216
        C               //  VARIABLE=EXPRESSION1.                                     217
0195                    ITMP=TCEND                                                     218
0196                    CALL SCAN(BYE,2,ECCND1)                                        219
0197                    EX2A=TCBEG                                                     220
0198                    EX2B=TCEND                                                     221
0199                    TCEND=TOEND+2                                                  222
0200                    IF(.NOT.( ECCND1 )) GO TO 99958                                223
0201                       TCEND=ITMP                                                  224
0202            99958   CCNTINUE                                                       225
0203                    CALL SCAN(DCC,2,ECCND)                                         226
0204                    IF(.NOT.( ECCND1 )) GO TO 99957                                227
0205                       EX2B=TCEND                                                  228
0206            99957   CCNTINUE                                                       229
0207                    IF(.NOT.( ECCND )) GC TO 99956                                 230
0208                       CALL ERROR(2,DOC,2,1,0,0)                                   231
        C                  // ERROR--MISSING 'DO'- SUPPLIED AT END OF STATEMENT.       232
0209                       TOEND=TCEND+1                                               233
0210                       GC TO 99955                                                 234
0211            99956   CCNTINUE                                                       234
0212                       TCEND=TCEND+2                                               235
0213            99955   CCNTINUE                                                       236
0214                    LAB1=GENLAB(1)                                                 237
0215                    LAB2= GENLAB(1)                                                238
0216                    LAB3=GENLAB(1)                                                 239
0217                    LAB4=GENLAB(1)                                                 240
```

```
                C                   // LAB IS THE IDENTIFYING LABEL, LAB4 IS THE TARGET OF EX      241
                C     S             IT,                                                         241
                C                   // AND LAB3 IS THE TARGET OF CYCLES.                          242
0218                                STACK(TOP+1)=LAB2                                           243
0219                                STACK(TOP+3)=LAB                                            244
0220                                STACK(TOP+4)=LAB3                                           245
0221                                STACK(TOP+5)=LAB4                                           246
0222                                STACK(TOP+6)=FFOR                                           247
0223                                STACK(TOP+2)=TPITR                                          248
0224                                TOP=TOP+6                                                   249
0225                                IF(.NOT.( TOP.GT.STMAX )) GO TO 99954                       250
0226                                   CALL ERROR(6,0,0,5,0,0)                                  251
0227            99954               CONTINUE                                                    252
                C                   //  STACK OVERFLOWED----PROGRAM TERMINATED.                 253
0228                                TPITR=TOP                                                   254
0229                                 I=1                                                        255
0230                                GO TO 99953                                                 255
0231            99952                I= I+1                                                     255
0232            99953               IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99950                      255
0233                                   FBUF(I)=BLANK                                            256
0234            99951               GO TO 99952                                                 257
0235            99950               CONTINUE                                                    257
```

```
0236                    CALL FCUT(GCTC,1,6)                                     258
0237                    CALL FCUT(LABEL,LAB1,LAB1+4)                            259
0238                    CALL FPRNT                                              260
        C               // GC TC LAB1.                                          261
0239                     I=1                                                    262
0240                    GC TO 99949                                             262
0241    99948           I= I+1                                                  262
0242    99949           IF(.NCT.(( I-( 5 )).LE.0)) GO TC 99946                  262
0243                       L=LAB2+I-1                                           263
0244                       FBUF(I)=LABEL(L)                                     264
0245    99947           GO TO 99948                                             265
0246    99946           CONTINUE                                                265
0247                    FBUF(6)=BLANK                                           266
0248                    CALL FOUT(SPIN,VBLEA,VBLEB)                             267
0249                    CALL FCUT(EQL,1,1)                                      268
0250                    CALL FOUT(SPIN,VBLEA,VBLEB)                             269
0251                    CALL FOUT(PLU,1,1)                                      270
0252                    IF(.NCT.( ECCND1 )) GC TC 99945                         271
        C                  // THE 'BY' IS MISSING--DEFAULT INCREMENT OF EXPRESSIO 272
        C    9             N TO CNE.                                            272
0253                       CALL FOUT(ONE,1,1)                                   273
0254                       GO TO 99944                                          274
0255    99945           CONTINUE                                                274
0256                       CALL FOUT(IFNCT,9,9)                                 275
0257                       CALL FOUT(SPIN,TOBEG,TCEND-2)                        276
0258                       CALL FOUT (RSTAR,1,1)                                277
0259    99944           CONTINUE                                                278
0260                    CALL FPRNT                                              279
        C               //  LAB2--INDICATES VARIABLE=VARIABLE+EXPRESSION3.      280
0261                     I=1                                                    281
0262                    GC TO 99943                                             281
0263    99942           I= I+1                                                  281
0264    99943           IF(.NOT.(( I-( 5 )).LE.0)) GC TO 99940                  281
0265                       L=LAB1+I-1                                           282
0266                       FBUF(I)=LABEL(L)                                     283
0267    99941           GC TO 99942                                             284
```

```
0268        99940       CONTINUE                                                  284
0269                    FBUF(6)=BLANK                                             285
0270                    CALL FOUT(IFNOT,1,9)                                      286
0271                    CALL FOUT(IFNOT,9,9)                                      287
0272                    CALL FOUT(SPIN,VBLEA,VBLEB)                               288
0273                    CALL FOUT(MINUS,1,1)                                      289
0274                    CALL FOUT(IFNOT,9,9)                                      290
0275                    CALL FOUT(SPIN,EX2A,EX2B)                                 291
0276                    CALL FOUT(RSTAR,1,1)                                      292
0277                    CALL FOUT(RSTAR,1,1)                                      293
0278                    IF(.NOT.( .NOT.ECOND1 )) GO TO 99939                      294
            C              // IF 'BY' MISSING THEN GENERATE...(EXP2)).LE.0...     295
            C              //  IN PLACE OF...(EXP2)*(EXP3).LE.0.....              296
0279                       CALL FOUT(RSTAR,2,2)                                   297
0280                       CALL FOUT(IFNOT,9,9)                                   298
0281                       CALL FOUT(SPIN,TCBEG,TCEND-2)                          299
0282                       CALL FOUT(RSTAR,1,1)                                   300
0283        99939       CONTINUE                                                  301
0284                    CALL FOUT(LEC,1,5)                                        302
0285                    CALL FOUT(CGOTO,1,9)                                      303
0286                    CALL FOUT(LABEL,LAB4,LAB4+4)                              304
0287                    CALL FPRNT                                                305
```

```
      C              // LAB1...IF(.NOT.((VBLE-(EXP3).LE.0) GO TO LAB3...                    306
0288                 STEND=TOEND                                                          307
0289                 INDFLG=FRWRD                                                         308
0290                 CURFLG=CLD                                                           309
0291                 IF(.NOT.( FSTRT+INCR.LT.IMAX )) GO TO 99938                          310
0292                    FSTRT=FSTRT+INCR                                                  311
0293      99938      CONTINUE                                                             312
0294                 GO TO 99994                                                          313
0295      99962      IF(.NOT.( TYPE.EQ.RREPT )) GO TO 99937                               313
0296                 CALL CHECK(RREPT,BAD)                                                314
0297                 IF(.NOT.( BAD )) GO TO 99936                                         315
0298                    GO TO 99998                                                       316
0299      99936      CONTINUE                                                             317
0300                 IF(.NOT.( LBFLG )) GO TO 99935                                       318
      C                 //  LABELLED STMT, GENERATED 'LAB CONTINUE'                       319
0301                    CALL FOUT(CONTI,1,8)                                              320
0302                    CALL FPRNT                                                        321
0303      99935      CONTINUE                                                             322
0304                 STTOP=STACK(TOP)                                                     323
      C              // LAB3=EXIT TARGET,LAB2=CYCLE TARGET,LAB1=LOOP BACK LABE            324
      C    S         L                                                                    324
0305                 IF(.NOT.( STTOP.EQ.FFOR )) GO TO 99934                               325
0306                    LAB1=STACK(TOP-5)                                                 326
0307                    LAB2=STACK(TOP-2)                                                 327
0308                    LAB3=STACK(TOP-1)                                                 328
0309                    TOP=TOP-6                                                         329
0310                    LBPTR=LBPTR-20                                                    330
0311                    GO TO 99933                                                       331
0312      99934      CONTINUE                                                             331
0313                    LAB1=STACK(TOP-3)                                                 332
0314                    LAB2=STACK(TOP-2)                                                 333
0315                    LAB3=STACK(TOP-1)                                                 334
0316                    TOP=TOP-5                                                         335
0317                    LBPTR=LBPTR-15                                                    336
0318      99933      CONTINUE                                                             337
0319                 TPITR=STACK(TPITR-4)                                                 338
```

```
0320                    FSTRT=FSTRT-INCR                                    339
0321                    FPTR=FSTRT                                          340
0322                     I=1                                                341
0323                    GC TC 99932                                         341
0324          99931      I= I+1                                             341
0325          99932     IF(.NCT.(( I-( 5 )).LE.0)) GO TC 99929              341
0326                       L=LAB2+I-1                                       342
0327                       FBUF(I)=LABEL(L)                                 343
0328          99930     GO TO 99931                                         344
0329          99929     CONTINUE                                            344
0330                    FBUF(6)=BLANK                                       345
0331                    CALL FCUT(GCTC,1,6)                                 346
0332                    CALL FCUT(LABEL,LAB1,LAB1+4)                        347
0333                    CALL FPRNT                                          348
              C         //    GC TC LABEL 1                                 349
0334                     I=1                                                350
0335                    GC TO 99928                                         350
0336          99927      I= I+1                                             350
0337          99928     IF(.NCT.(( I-( 5 )).LE.0)) GO TO 99925              350
0338                       L=LAB3+I-1                                       351
0339                       FBUF(I)=LABEL(L)                                 352
0340          99926     GO TO 99927                                         353
```

```
0341        99925          CONTINUE                                                    353
0342                       FBUF(6)=BLANK                                               354
0343                       CALL FOUT(CONTI,1,8)                                        355
0344                       CALL FPRNT                                                  356
            C              //    LAB3 CONTINUE                                         357
0345                       STEND=TOEND                                                 358
0346                       CURFLG=NEW                                                  359
0347                       INDFLG=BKWRD                                                360
0348                    GO TO 99994                                                    361
0349        99937       IF(.NOT.( TYPE.EQ.CCASE )) GO TO 99924                         361
0350                       LAB=GENLAB(1)                                               362
            C              //  LABEL(1:5)=BLANKS.                                      363
0351                       STACK(TOP+1)=1                                              364
0352                       STACK(TOP+2)=LAB                                            365
0353                       STACK(TOP+3)=CCASE                                          366
0354                       TOP=TOP+3                                                   367
0355                       IF(.NOT.( TOP.GT.STMAX )) GO TO 99923                       368
0356                          CALL ERROR(6,C,C,5,0,0)                                  369
0357        99923          CONTINUE                                                    370
            C              //  STACK OVERFLOWED----PROGRAM TERMINATED.                 371
0358                       STEND=TOEND                                                 372
0359                       CURFLG=OLD                                                  373
0360                       INDFLG=FRWRD                                                374
0361                       IF(.NOT.( LBFLG )) GO TO 99922                              375
            C                 //  LABELED STATEMENT--GENERATE 'LAB CONTINUE'.          376
0362                          CALL FOUT(CONTI,1,8)                                     377
0363                          CALL FPRNT                                               378
0364        99922          CONTINUE                                                    379
            C              //  SET THE INDENTION                                       380
0365                       IF(.NOT.( FSTRT+INCR.LT.IMAX )) GO TO 99921                 381
0366                          FSTRT=FSTRT+INCR                                         382
0367        99921          CONTINUE                                                    383
0368                    GO TO 99994                                                    384
0369        99924       IF(.NOT.( TYPE.EQ.COLON )) GO TO 99920                         384
0370                       CALL CHECK(COLON,BAD)                                       385
0371                       IF(.NOT.( BAD )) GO TO 99919                                386
```

```
0372                  GO TO 99998                                          387
0373    99919       CONTINUE                                               388
0374                IF(.NOT.( LBFLG )) GO TO 99918                         389
0375                  CALL ERROR(1,CASS,4,2,0,0)                           390
        C             // ERROR--LABELLING THE CASE HAMPERS PROGRAM CLEANLINE  391
        C     9       SS                                                   391
        C             // DELETED FROM THE FORTRAN OUTPUT.                  392
0376                   I=1                                                 393
0377                  GO TO 99917                                          393
0378    99916          I= I+1                                              393
0379    99917         IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99914               393
0380                     FBUF(I)=BLANK                                     394
0381    99915         GO TO 99916                                          395
0382    99914         CONTINUE                                             395
0383    99918       CONTINUE                                               396
0384                LAB1=STACK(TOP-1)                                      397
0385                LAB2=STACK(TOP-2)                                      398
0386                FSTRT=FSTRT-INCR                                       399
0387                FPTR=FSTRT                                             400
        C           // CASE MATERIAL PRINTED ONE IDENT TO THE RIGHT.       401
0388                IF(.NOT.( LAB2.NE.1 )) GO TO 99913                     402
0389                  CALL FOUT(GOTC,1,6)                                  403
```

```
0390                      CALL    FOUT(LABEL,LAB1,LAB1+4)                          404
0391                      CALL    FPRNT                                            405
          C               // GO TO LAB1.                                           406
0392      99913       CONTINUE                                                     407
0393                  CALL SCAN1(COLN,ECOND)                                       408
0394                  IF(.NOT.( ECOND )) GO TO 99912                               409
0395                     CALL ERROR(2,COLN,1,1,0,0)                                410
0396      99912       CONTINUE                                                     411
          C           // ERROR--MISSING COLON-SUPPLIED AT END OF STATEMENT.       412
0397                  K=TCEND                                                      413
0398                  TCEND=TOBEG                                                  414
          C           // SPIN(K)=IMAX OR SPIN(K+1)=SEMICOLON OR COLON.            415
0399                  TMAX=IMAX                                                    416
0400                  IMAX=K                                                       417
0401                  CALL SCAN(ELZ,4,ECOND)                                       418
0402                  IMAX=TMAX                                                    419
0403                  IF(.NOT.( ECOND )) GO TO 99911                               420
0404                     LAB3=GENLAB(1)                                            421
0405                     TCEND=K                                                   422
          C              // OUTPUT-LAB2-IF NOT CONDITION THE GO TO LAB3.          423
0406                      I=1                                                      424
0407                     GO TO 99910                                               424
0408      99909           I= I+1                                                   424
0409      99910           IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99907                   424
0410                        L=LAB2+I-1                                             425
0411                        FBUF(I)=LABEL(L)                                       426
0412      99908           GO TO 99909                                              427
0413      99907           CONTINUE                                                 427
0414                      FBUF(6)=BLANK                                            428
0415                      CALL    FOUT(IFNOT,1,9)                                  429
0416                      CALL FOUT(SPIN,TCBEG,TCEND)                              430
0417                      CALL    FOUT(CGOTO,1,9)                                  431
0418                      CALL FOUT(LABEL,LAB3,LAB3+4)                             432
0419                      CALL    FPRNT                                            433
0420                      TCEND=TCEND+1                                            434
0421                      GO TO 99906                                              435
```

```
0422        99911       CONTINUE                                                435
            C              //  ELSE CLAUSE FOUND                                436
0423                        I=1                                                 437
0424                       GO TO 99905                                          437
0425        99904           I= I+1                                              437
0426        99905          IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99902               437
0427                          L=LAB2+I-1                                        438
0428                          FBUF(I)=LABEL(L)                                  439
0429        99903          GO TO 99904                                          440
0430        99902          CONTINUE                                             440
0431                       FBUF(6)=BLANK                                        441
0432                       CALL FOUT(CONTI,1,8)                                 442
0433                       CALL FPRNT                                           443
            C              //  LAB2 CONTINUE                                    444
0434                       TOEND=K+1                                            445
0435                       LAB3=1                                               446
            C              //  LABEL(1....5) ARE BLANKS.                        447
0436        99906       CONTINUE                                                448
0437                    STACK(TOP-2)=LAB3                                       449
0438                    INDFLG=NEUTR                                            450
0439                    CURFLG=PULOUT                                           451
0440                    STEND=TOEND                                             452
```

```
            C               // RESTORE THE INCENTATION.                                  453
 0441                       IF(.NOT.( FSTRT+INCR.LT.IMAX )) GO TO 99901                454
 0442                          FSTRT=FSTRT+INCR                                        455
 0443           99901       CONTINUE                                                   456
 0444                     GO TO 99994                                                  457
 0445           99920     IF(.NOT.( TYPE.EQ.ENDCAS )) GO TO 99900                      457
 0446                       CALL CHECK(ENDCAS,BAD)                                     458
 0447                       IF(.NOT.( BAD )) GO TO 99899                               459
 0448                          GO TO 99998                                             460
 0449           99899       CONTINUE                                                   461
 0450                       FSTRT=FSTRT-INCR                                           462
 0451                       FPTR=FSTRT                                                 463
 0452                       IF(.NOT.( LBFLG )) GO TO 99898                             464
            C                  // LABELLED STATEMENT--GENERATE 'LAB CONTINUE'          465
 0453                          CALL FOUT(CONTI,1,8)                                    466
 0454                          CALL FPRNT                                              467
 0455           99898       CONTINUE                                                   468
 0456                       LAB1=STACK(TOP-1)                                          469
 0457                       LAB2=STACK(TOP-2)                                          470
 0458                       TOP=TOP-3                                                  471
 0459                       LBPTR=LAB1                                                 472
 0460                        I=1                                                       473
 0461                       GO TO 99897                                                473
 0462           99896        I= I+1                                                   473
 0463           99897       IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99894                     473
 0464                          L=LAB2+I-1                                              474
 0465                          FBUF(I)=LABEL(L)                                        475
 0466           99895       GO TO 99896                                                476
 0467           99894       CONTINUE                                                   476
 0468                       FBUF(6)=BLANK                                              477
 0469                       CALL FOUT(CONTI,1,8)                                       478
 0470                       CALL  FPRNT                                                479
            C               //  LAB2 CONTINUE                                         480
 0471                        I=1                                                       481
 0472                       GO TO 99893                                                481
 0473           99892        I= I+1                                                   481
```

```
0474        99893       IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99890                    481
0475                       L=LAB1+I-1                                              482
0476                       FBUF(I)=LABEL(L)                                        483
0477        99891       GO TO 99892                                                484
0478        99890       CONTINUE                                                   484
0479                    FBUF(6)=BLANK                                              485
0480                    CALL FCLT(CONTI,1,8)                                       486
0481                    CALL FPRNT                                                 487
            C           //  LAB1 CONTINUE                                          488
0482                    STEND=TOEND                                                489
0483                    INDFLG=BKWRD                                               490
0484                    CURFLG=NEW                                                 491
0485                 GO TO 99994                                                   492
0486        99900    IF(.NOT.( TYPE.EQ.LLOCP )) GO TO 99889                        492
            C           //  LAB2 IS THE TARGET OF CYCLE AND LAB3 OF EXITS          493
            C           //  LAB1 IS THE IDENTIFYING LABEL.                         494
0487                    LAB1=GENLAB(0)                                             495
0488                    LAB2=GENLAB(1)                                             496
0489                    LAB3= GENLAB(1)                                            497
0490                    STACK(TOP+2)=LAB1                                          498
0491                    STACK(TOP+3)=LAB2                                          499
0492                    STACK(TOP+4)=LAB3                                          500
```

```
0493                    STACK(TOP+5)=LLOCP                                      501
0494                    STACK(TOP+1)=TPITR                                      502
0495                    TOP=TOP+5                                               503
0496                    IF(.NOT.( TOP.GT.STMAX )) GO TO 99888                   504
0497                       CALL ERROR(6,0,0,5,0,0)                              505
0498          99888     CONTINUE                                                506
              C         //  STACK OVERFLOWED, PROGRAM TERMINATED                507
0499                     I=1                                                     508
0500                    GO TO 99887                                             508
0501          99886      I= I+1                                                 508
0502          99887     IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99884                  508
0503                       L=LAB1+I-1                                           509
0504                       FBUF(I)=LABEL(L)                                     510
0505          99885     GO TO 99886                                             511
0506          99884     CONTINUE                                                511
0507                    FBUF(6)=BLANK                                           512
0508                    CALL FOUT(CONTI,1,8)                                    513
0509                    CALL FPRNT                                              514
0510                    TPITR=TOP                                               515
0511                    CLRFLG=OLD                                              516
0512                    INDFLG=FRWRD                                            517
0513                    STEND=TOEND                                             518
0514                    IF(.NOT.( FSTRT+INCR.LT.IMAX )) GO TO 99883             519
0515                       FSTRT=FSTRT+INCR                                     520
0516          99883     CONTINUE                                                521
0517                  GO TO 99994                                               522
0518          99889   IF(.NOT.( TYPE.EQ.WHLE )) GO TO 99882                     522
0519                     LAB1=GENLAB(0)                                         523
0520                     LAB2=GENLAB(1)                                         524
0521                     LAB3=GENLAB(1)                                         525
              C          //  LAB2 IS THE TARGET OF EXITS, LAB3 FOR CYCLES.      526
0522                     CALL SCAN(DOO,2,ECOND)                                 527
0523                     IF(.NOT.( ECOND )) GO TO 99881                         528
0524                        CALL ERROR(2,DOO,2,1,0,0)                           529
              C             //  CALL ERROR---MISSING 'DO'-SUPPLIED AT THE END OF S  530
              C     S       TMT.                                                530
```

```
0525                      TCEND=TCEND+1                                    531
0526                      GO TO 99880                                      532
0527          99881     CONTINUE                                         532
0528                      TCEND=TCEND+2                                    533
0529          99880     CONTINUE                                         534
0530                    IF(.NOT.( .NOT.LBFLG )) GO TO 99879                535
0531                       I=1                                             536
0532                      GO TO 99878                                      536
0533          99877        I= I+1                                        536
0534          99878       IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99875          536
0535                          L=LAB1+I-1                                   537
0536                          FBUF(I)=LABEL(L)                             538
0537          99876       GO TO 99877                                      539
0538          99875       CONTINUE                                         539
0539                      FBUF(6)=BLANK                                    540
0540          99879     CONTINUE                                         541
0541                    CALL FOUT(IFNOT,1,9)                               542
0542                    CALL FOUT(SPIN,TCBEG,TCEND-2)                      543
0543                    CALL FOUT(CGOTO,1,9)                               544
0544                    CALL FOUT(LABEL,LAB2,LAB2+4)                       545
0545                    CALL FPRNT                                         546
              C         //  GENERATE LAB1 AND IF TRUE COND GO TO LAB2.     547
```

```
0546                    STACK(TOP+2)=LAB1                                    548
0547                    STACK(TOP+3)=LAB3                                    549
0548                    STACK(TOP+4)=LAB2                                    550
0549                    STACK(TOP+5)=WHLE                                    551
0550                    STACK(TOP+1)=TPITR                                   552
0551                    TOP=TOP+5                                            553
0552                    IF(.NOT.( TOP.GT.STMAX )) GO TO 99874                554
0553                       CALL ERROR(6,0,0,5,0,0)                           555
0554          99874     CONTINUE                                             556
              C         //  STACK OVERFLOWED, PROGRAM TERMINATED.            557
0555                    TPITR=TOP                                            558
0556                    CURFLG=OLD                                           559
0557                    INCFLG=FRWRD                                         560
0558                    STEND=TOEND                                          561
0559                    IF(.NOT.( FSTRT+INCR.LT.IMAX )) GO TO 99873          562
0560                       FSTRT=FSTRT+INCR                                  563
0561          99873     CONTINUE                                             564
0562                 GO TO 99994                                             565
0563          99882  IF(.NOT.( TYPE.EQ.CONT )) GO TO 99872                   565
0564                    I=1                                                  566
0565                    GO TO 99871                                          566
0566          99870     I= I+1                                               566
0567          99871     IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99868               566
0568                       FBUF(I)=SPIN(I)                                   567
0569          99869     GO TO 99870                                          568
0570          99868     CONTINUE                                             568
0571                    TOBEG=7                                              569
0572          99867     IF(.NOT.( SPIN(TOBEG).EQ.BLANK )) GO TO 99866        570
0573                       TOBEG=TOBEG+1                                     571
0574                       IF(.NOT.( TOBEG.GT.IMAX )) GO TO 99864            572
0575                          TOEND=IMAX+1                                   573
              C               // BLANK CONTINUATION CARD, DELETED.           574
0576                          GO TO 99866                                    575
0577          99864        CONTINUE                                          576
0578          99865     GO TO 99867                                          577
0579          99866     CONTINUE                                             577
```

```
0580                     IF(.NCT.( TOEND.GT.IMAX )) GC TC 99863                    578
0581                        GO TC 99998                                            579
0582           99863     CCNTINUE                                                  580
               C         // SPIN(TOBEG)=NCNBLANK; REST OF CARD CAN BE INDENTED.    581
0583                     CALL FHAND                                                582
0584                     FBUF(6)=BLANK                                             583
0585                     STREG=TOBEG                                               584
0586                     STEND=TOEND                                               585
0587                     CURFLG=CLD                                                586
0588                     INDFLG=NEUTR                                              587
0589                  GO TC 99994                                                  588
0590           99872  IF(.NCT.( TYPE.EQ.EEXIT )) GO TO 99862                      588
0591                     CALL CHECK(EEXIT,BAD)                                     589
0592                     IF(.NOT.( BAD )) GC TO 99861                              590
0593                        GC TO 99998                                            591
0594           99861     CCNTINUE                                                  592
0595                     CALL TYHAND(TYPE,EEEXT,CCYCLE)                            593
0596                  GO TC 99994                                                  594
0597           99862  IF(.NOT.( TYPE.EQ.NCYCLE )) GO TO 99860                     594
0598                     CALL CHECK(EEXIT,BAD)                                     595
0599                     IF(.NCT.( BAD )) GC TO 99859                              596
0600                        GO TO 99998                                            597
```

```
0601        99859       CONTINUE                                                  598
0602                    CALL TYHAND(TYPE,EEEXT,CCYCLE)                            599
0603                GO TO 99994                                                   600
0604        99860       IF(.NOT.( TYPE.EQ.UNTL )) GO TO 99858                     600
0605                    CALL CHECK(UNTL,BAD)                                      601
0606                    IF(.NOT.( BAD )) GO TO 99857                              602
0607                      GO TO 99998                                             603
0608        99857       CONTINUE                                                  604
0609                    FSTRT=FSTRT-INCR                                          605
0610                    FPTR=FSTRT                                                606
0611                    LAB1=STACK(TOP-3)                                         607
0612                    LAB2=STACK(TOP-2)                                         608
0613                    LAB3=STACK(TOP-1)                                         609
0614                    TOP=TOP-5                                                 610
0615                    LBPTR=LBPTR-15                                            611
0616                    TPTR=STACK(TPTR-4)                                        612
0617                    IF(.NOT.( LBFLG )) GO TO 99856                            613
0618                      CALL ERROR(1,LUNTL,5,2,0,0)                             614
            C             //  ERROR---LABELLING OF THE UNTIL HAMPERS PROGRAM      615
            C             //  CLARITY, DELETED FROM FORTRAN OUTPUT.               616
0619        99856       CONTINUE                                                  617
0620                     I=1                                                      618
0621                    GO TO 99855                                               618
0622        99854        I= I+1                                                   618
0623        99855       IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99852                    618
0624                      L=LAB2+I-1                                              619
0625                      FBUF(I)=LABEL(L)                                        620
0626        99853       GO TO 99854                                               621
0627        99852       CONTINUE                                                  621
0628                    FBUF(6)=BLANK                                             622
0629                    CALL FOUT(IFNOT,1,9)                                      623
0630                    CALL SCAN(REPT,6,ECCND)                                   624
0631                    CALL FOUT(SPIN,TOBEG,TOEND)                               625
0632                    IF(.NOT.( ECCND )) GO TO 99851                            626
0633                      CALL ERROR(2,REPT,6,1,0,0)                              627
            C             // ERROR---MISSING REPEAT-SUPPLIED AT THE END OF THE S  628
```

```
      C     9           TATEMENT.                                          628
0634                    TCEND=TCEND+1                                      629
0635                    GO TO 99850                                        630
0636        99851     CONTINUE                                             630
0637                    TCEND=TCEND+6                                      631
0638        99850     CONTINUE                                             632
0639                  CALL FOUT( CGOTO,1,9)                                633
0640                  CALL FOUT (LABEL,LAB1,LAB1+4)                        634
0641                  CALL FPRNT                                           635
      C               //  GENERATE IF(.NOT.(COND)) GO TO LAB1              636
      C               //  GENERATE THE EXIT TARGET---LAB3 CONTINUE.        637
0642                   I=1                                                 638
0643                  GO TO 99849                                          638
0644        99848      I= I+1                                              638
0645        99849     IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99846               638
0646                    L=LAB3+I-1                                         639
0647                    FBUF(I)=LABEL(L)                                   640
0648        99847     GO TO 99848                                          641
0649        99846     CONTINUE                                             641
0650                  FBUF(6)=BLANK                                        642
0651                  CALL FOUT(CONTI,1,8)                                 643
0652                  CALL FPRNT                                           644
```

```
0653                      STEND=TOEND                                          645
0654                      INDFLG=BKWRD                                         646
0655                      CURFLG=NEW                                           647
0656                    GO TO 99994                                            648
0657          99858     IF(.NOT.( TYPE.EQ.SUBFN )) GO TO 99845                 648
0658                      WRITE(PRTR,5711)                                     649
0659           5711       FORMAT(1H1)                                          650
0660                      NUMB=100000                                          651
0661                      STACK(TOP+1)=SUBFN                                   652
0662                      TOP=TOP+1                                            653
0663                      IF(.NOT.( TOP.GT.STMAX )) GO TO 99844                654
0664                        CALL ERROR(6,C,0,5,0,0)                            655
0665          99844       CONTINUE                                             656
              C           // STACK OVERFLOWED, PROGRAM TERMINATED.             657
0666                      CALL FHAND                                           658
0667                      CURFLG=OLD                                           659
0668                      INDFLG=FRWRD                                         660
0669                      STEND=TOEND                                          661
0670                      IF(.NOT.( FSTRT+INCR.LT.IMAX )) GO TO 99843          662
0671                        FSTRT=FSTRT+INCR                                   663
0672          99843       CONTINUE                                             664
0673                    GO TO 99994                                            665
0674          99845     IF(.NOT.( TYPE.EQ.EEND )) GO TO 99842                  665
0675                      CALL CHECK(EEND,BAD)                                 666
0676                      IF(.NOT.( BAD )) GO TO 99841                         667
0677                        GO TO 99998                                        668
0678          99841       CONTINUE                                             669
0679                      FSTRT=FSTRT-INCR                                     670
0680                      FPTR=FSTRT                                           671
0681                      IF(.NOT.( LBFLG )) GO TO 99840                       672
              C             // LABELLED STATEMENT--GENERATE 'LAB CONTINUE'     673
0682                        CALL FOUT(CONTI,1,8)                               674
0683                        CALL FPRNT                                         675
0684          99840       CONTINUE                                             676
0685                      TOP=TOP-1                                            677
0686                      CALL FOUT(SPIN,TOBEG,TOEND)                          678
```

```
0687                    CALL FPRNT                                                  679
      C                 // GIVE ANY END OF SUBROUTINE MESSAGES AND RESET ANY VARI   680
      C    9            BLES                                                       680
      C                 // IF NECESSARY.                                           681
0688                    STEND=TOEND                                                682
0689                    INDFLG=BKWRD                                               683
0690                    CURFLG=NEW                                                 684
0691                  GO TO 99994                                                  685
0692          99842   IF(.NOT.( TYPE.EQ.COMT )) GO TO 99839                        685
0693                    FBUF(1)=CEE                                                686
0694                    TOBEG=2                                                    687
0695                    TOEND=IMAX                                                 688
0696          99838     CONTINUE                                                   689
0697                      IF(.NOT.( SPIN(TOBEG).NE.BLANK )) GO TO 99835            690
0698                        I=2                                                    691
0699                      GO TO 99834                                              691
0700          99833         I= I+1                                               691
0701          99834         IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99831               691
0702                          FBUF(I)=BLANK                                        692
0703          99832         GO TO 99833                                          693
0704          99831         CONTINUE                                             693
0705                        CALL FOUT(SPIN,TOBEG,IMAX)                             694
```

```
0706                        CALL FPRNT                                          695
0707                        STBEG=TOBEG                                         696
0708                        GO TO 99836                                         697
0709        99835         CONTINUE                                            698
0710                      TOBEG=TOBEG+1                                       699
0711                      IF(.NOT.( TOBEG.GT.IMAX )) GO TO 99830              700
            C                 // A BLANK COMMENT, OUTPUT A BLANK LINE.        701
0712                        CALL FOUT(SPIN,FPTR,FMAX)                         702
0713                        CALL FPRNT                                        703
0714                        STBEG=SSTRT                                       704
0715                        GO TO 99836                                       705
0716        99830         CONTINUE                                            706
0717        99837       GO TO 99838                                           707
0718        99836       CONTINUE                                              707
0719                    STEND=IMAX                                            708
0720                    CURFLG=OLD                                            709
0721                    INCFLG=NEUTR                                          710
0722                  GO TO 99994                                             711
0723        99839     IF(.NOT.( TYPE.EQ.EOJ )) GO TO 99829                    711
            C           //  CLOSE THE FILE.                                   712
0724                    WRITE(PRTR,1300)                                      713
0725        1300        FORMAT(/35H          END OF SPARKS PROGRAM.....)      714
0726                    STOP                                                  715
0727        99829     CONTINUE                                                716
0728        99994     CONTINUE                                                716
0729                  IF(.NOT.( CURFLG.EQ.PULOUT )) GO TO 99828               717
0730                    SSTRT=SSTRT-INCR                                      718
0731                    CALL SPRNT                                            719
0732                    SSTRT=SSTRT+INCR                                      720
0733                    IF(.NOT.( SSTRT.GT.SMAX )) GO TO 99827                721
0734                      SSTRT=SSTRT-INCR                                    722
0735        99827       CONTINUE                                            723
0736                    BLKNO=BLKNO+1                                         724
0737        99828     CONTINUE                                                725
0738                  IF(.NOT.( CURFLG.EQ.OLD )) GO TO 99826                  726
0739                    CALL SPRNT                                            727
```

```
0740        99826       CONTINUE                                                728
0741                    IF(.NOT.( INDFLG.EQ.FRWRD )) GO TO 99825                729
0742                       SSTRT=SSTRT+INCR                                     730
0743                       IF(.NOT.( SSTRT.GT.SMAX )) GO TO 99824               731
0744                          SSTRT=SSTRT-INCR                                  732
0745        99824          CONTINUE                                             733
0746                       IF(.NOT.( CURFLG.EQ.NEW )) GO TO 99823               734
0747                          CALL SPRNT                                        735
0748        99823          CONTINUE                                             736
0749                       BLKNO=BLKNO+1                                        737
0750                       NESTNO=NESTNO+1                                      738
0751        99825       CONTINUE                                                739
0752                    IF(.NOT.( INDFLG.EQ.BKWRD )) GO TO 99822                740
0753                       SSTRT=SSTRT-INCR                                     741
0754                       IF(.NOT.( CURFLG.EQ.NEW )) GO TO 99821               742
0755                          CALL SPRNT                                        743
0756        99821          CONTINUE                                             744
0757                       NESTNO=NESTNO-1                                      745
0758                       BLKNO=BLKNO+1                                        746
0759        99822       CONTINUE                                                747
0760                    STMNO=STMNO+1                                           748
0761        99998    GO TO 99999                                                749
```

```
0762        99997    CONTINUE                                          749
0763                 END                                               750
```

```
0001                BLOCK DATA                                                    751
0002                   IMPLICIT INTEGER(A-Z)                                      752
0003                   LOGICAL LBFLG                                              753
0004                   COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG   754
0005                   COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX  755
0006                   COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)          756
0007                   COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),  757
                    1  TPITR,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB         758
0008                   COMMON/CTPAT/CGOTO(9),CONTI(8),EQL(1),GOTO(6),IFNOT(9),LEO(5),  759
                    1  MINUS(1),ONE(1),PLL(1),RSTAR(2)                           760
0009                   COMMON/INPAT/COLN(1),ELZ(4),DBSLH(2),DOO(2),BYE(2),THENN(4),TOO  761
                    9  (2)                                                        761
0010                   COMMON/STTYP1/CCASE,COLCN,CBLSL,EELSE,EEND,ECJ,EEXIT,FFOR,IIF,  762
                    1  LLOOP,REPT,UNTL,WHLE,SUBFN,RINLO,NCYCLE,ENDCAS,EENDIF     763
0011                   COMMON/STTYP2/COMT,CONT,FCTRN,BLANK,SEMI                   764
0012                   COMMON/TABLES/TRES(23),RESRD(200)                          765
0013                   COMMON/NUMRIC/DEE,PRTR,FFILE,CRDR                          766
0014                   COMMON/KEYWD/EEEND(3),EEEXT(4),CCYCLE(5),UUNTL(5),CASS(4),IIIF(  767
                    9  2),                                                        767
                    1  ITERAT(9),ENCASS(7),ENIF(5),REPT(6),CCCLCN(5)            768
                C      //                                                         769
                C      //  CRITICAL CODES--STFRM IS ACCESSED USING THE FOLLOWING:  770
                C      //  CCASE=1, FFOR=2, IIF=3, LLOOP=4, WHLE=5, SUBFN=6.      771
                C      //                                                         772
0015                   DATA SPIN/1HT,1HH,1HI,1HS,1H ,1HI,1HS,1H ,1HA,1H ,1HS,1HP,1HA,  773
                    X  1HR,1HK,1HS,1H ,1HP,1HR,1HC,1HG,1HR,1HA,1HM,                773
                    1  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H , 1H , 1H ,1H ,1H ,1H ,  775
                    2  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H , 1H , 1H ,1H ,1H ,1H ,  776
                    3  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H , 1H , 1H ,1H ,1H ,1H ,  777
                    4  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H , 1H , 1H ,1H ,1H ,1H /  778
0016                   DATA BLNKS/131*1H /                                        779
0017                   DATA STMAX,LBMAX,NUMB/200,200,100000/                      780
0018                   DATA NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT/1,2,3,1,2,3/         781
                C      //   STACK FRAME SIZES:                                    782
                C      //              CASE    FOR       IF     LOOP    SUBFN     WHILE  783
0019                   DATA STFRM /  3   ,   6   ,   2   ,   5   ,    1 ,    5   /  784
```

```
0020                DATA    CGOTO/1H),1H),1H ,1HG ,1HC,1H ,1HT,1HO,1H /                    785
0021                DATA    CONTI/1HC,1HO,1HN,1HT,1HI,1HN,1HU,1HE/                         786
0022                DATA    EQL/1H=/                                                       787
0023                DATA    GOTO /1HG,1HO,1H ,1HT,1HO,1H /                                 788
0024                DATA    IFNOT/1HI,1HF,1H(,1H.,1HN,1HO,1HT,1H.,1H(/                     789
0025                DATA    LABEL/1H ,1H ,1H ,1H ,1H ,195*1H /                             790
0026                DATA    LEQ/1H.,1HL,1HE,1H.,1HQ/                                       791
0027                DATA    MINUS/1H-/                                                     792
0028                DATA    ONE/1H1/                                                       793
0029                DATA    PLU/1H+/                                                       794
0030                DATA    RSTAR/1H),1H*/                                                 795
0031                DATA DBSLH,ELZ/1H/,1H/,1HE,1HL,1HS,1HE/                                796
0032                DATA COLN,COLON/1H:,1HC,1HO,1HL,1HO,1HN/                               797
0033                DATA DOC,BYE,THENN,REPT,TOO/1HD,1HO,1HB,1HY,1HT,1HH,1HE,1HN,1HR        798
                  9 ,                                                                      798
                  1 1HE,1HP,1HE,1HA,1HT,1HT,1HO/                                           799
0034                DATA     CCASE,CCLON,DBLSL,EELSE,EEND,EOJ,EEXIT,FFOR,COMT,             800
                  1 CONT,FOTRN,IIF,LLOOP,RREPT,UNTL,WHLE,SUBFN,RINLO                       801
                  1 ,NCYCLE,ENDCAS,EENDIF                                                  802
                  1 /1,7,8,9,10,11,12,2,13,14,15,3,4,16,17,6,5,18,19,20,21/                803
0035                DATA RESRD/4,1HO,1HA,1HS,1HE,1,170,                                    804
                  1 3,1HE,1HN,1HD,10,14,4,1HE,1HL,1HS,1HE,9,21,                            805
```

```
     2   3,1HE,1HC,1HJ,11,27,4,1HE,1HX,1HI,1HT,12,152,                          806
     3   3,1HF,1HC,1HR,2,133,                                                   807
     4   2,1HI,1HF,3,45,7,1HI,1HN,1HT,1HE,1HG,1HE,1HR,18,0,                     808
     5   7,1HL,1HO,1HG,1HI,1HC,1HA,1HL,18,65,4,1HL,1HC,1HO,1HP,4                809
     6   ,0,4,1HR,1HE,1HA,1HL,18,79,6,1HR,1HE,1HP,1HE,1HA,1HT,                  810
     7   16,0,5,1HU,1HN,1HT,1HI,1HL,17,0,5,1HW,1HH,1HI,1HL,1HE,6                811
     8   ,0,4,1HN,1HE,1HX,1HT,0,0,10,1HS,1HU,1HB,1HR,1HO,1HU,1HT                812
     9   ,1HI,1HN,1HE,5,0,2,1H/,1H/,8,0                                         813
     A   ,1,1H:,7,0,8,1HF,1HU,1HN,1HC,1HT,1HI,1HO,1HN,5,0,                      814
     B   5,1HB,1HL,1HC,1HC,1HK,5,0,7,1HE,1HN,1HD,1HC,1HA,1HS,1HE                815
     C   ,20,162,5,1HE,1HN,1HD,1HI,1HF,21,0,5,1HC,1HY,1HC,1HL,1HE              816
     D   ,19,0,23*0/                                                            817
C        //   178TH PLACE AVAILABLE FOR INSERTION.                              818
0036     DATA TRES/144,1,170,14,8,152,162,21,27,34,133,40,45,55,65,             819
     1   104,72,79,111,88,96,124,129/                                           820
C        // TRES CONTAINS POINTS INTO RESRD WHICH CONTAINS ALL THE               821
C        // RESERVE WORDS IN SPARKS.  *****  NOTE  *****  THIS ORDER IS         822
C        // PRESENTLY SET UP FOR EBCDIC.                                        823
0037     DATA PRTR,CRDR,FFILE/ 6, 5, 4/                                         824
0038     DATA BLANK,CEE,SEMI/1H ,1HC,1H;/                                       825
0039     DATA EEEND,ENCASS/1HE,1HN,1HD,1HE,1HN,1HD,1HC,1HA,1HS,1HE/             826
0040     DATA EEEXT,ENIF/1HE,1HX,1HI,1HT,1HE,1HN,1HD,1HI,1HF/                   827
0041     DATA CCYCLE,CASS/1HC,1HY,1HC,1HL,1HE,1HC,1HA,1HS,1HE/                  828
0042     DATA UUNTL,IIIF/1HU,1HN,1HT,1HI,1HL,1HI,1HF/                           829
0043     DATA ITERAT/1HI,1HT,1HE,1HR,1HA,1HT,1HI,1HV,1HE/                       830
0044      END                                                                   831
```

```
0001                SUBROUTINE DETNX(TYPE)                                        832
                C     //                                                          833
                C     // THE SUBROUTINE DETERMINES THE TYPE OF CYCLE STATEMENT AND 834
                C     // SETS TOBEG AND TCEND ON TOKEN BOUNDARIES.                 835
                C     //                                                          836
0002                  IMPLICIT INTEGER(A-Z)                                       837
0003                  LOGICAL LBFLG,ECOND                                         838
0004                  COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG    839
0005                  COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)           840
0006                  COMMON/STTYP2/COMT,CONT,FCTRN,BLANK,SEMI                    841
0007                  COMMON/MMNIC/CEE,PRTR,FFILE,CROR                            842
0008                  DATA EFF/1HF/                                               843
                C     //                                                          844
0009                  LBFLG=.FALSE.                                               845
0010            99999 CONTINUE                                                    846
0011            99996    CONTINUE                                                 847
0012                        TCEND=TOEND+1                                         848
0013                        IF(.NOT.( TCEND.GT.IMAX )) GO TO 99993                849
0014                           GO TO 99994                                        850
0015            99993       CONTINUE                                            851
0016                        IF(.NOT.( SPIN(TCEND).NE.BLANK )) GO TO 99992         852
0017                           IF(.NOT.( SPIN(TCEND).EQ.SEMI )) GO TO 99991       853
0018                              GO TO 99995                                     854
0019            99991          CONTINUE                                           855
0020                           IF(.NOT.( SPIN(1).EQ.CEE )) GO TO 99990            856
0021                              TOBEG=1                                         857
0022                              TOEND=1                                         858
0023                              TYPE=COMT                                       859
0024                              RETURN                                          860
0025            99990          CONTINUE                                           861
0026                           TOBEG=TOEND                                        862
0027                           CALL SCAN1(BLANK,ECOND)                            863
0028                           IF(.NOT.( ECOND )) GO TO 99988                     865
0029                              TYPE=FCTRN                                      866
0030                              RETURN                                          867
0031                           GO TO 99989                                        868
```

```
                $
**********01)  IEY002I LABEL*************************************************************
   0032         99988              IF(.NOT.( SPIN(6).NE.BLANK )) GO TO 99987                  868
   0033                               TYPE=CONT                                               869
   0034                               RETURN                                                  870
   0035                            GO TO 99989                                                871
                $
**********01)  IEY002I LABEL*************************************************************
   0036         99987              IF(.NOT.( TCEND.GT.6 )) GO TO 99986                        871
   0037                               CALL STTYP(TYPE)                                        872
   0038                               IF(.NOT.( .NOT.LBFLG )) GO TO 99985                     873
   0039                                  I=1                                                  874
   0040                               GO TO 99984                                             874
   0041         99983                   I= I+1                                              874
   0042         99984                  IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99981               874
   0043                                  FBUF(I)=BLNKS(I)                                     875
   0044         99982                  GO TO 99983                                          876
   0045         99981                  CONTINUE                                             876
   0046         99985               CONTINUE                                                877
   0047                               RETURN                                                  878
   0048                            GO TO 99989                                                879
                $
```

```
**********01)  IEY002I LABEL*****************************************************************
   0049        99986            IF(.NOT.( TOEND.LE.6 )) GO TO 99980                          879
   0050                            I=1                                                       880
   0051                           GO TO 99979                                                880
   0052        99978               I= I+1                                                   880
   0053        99979               IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99976                   880
   0054                              BLNKS(I)=SPIN(I)                                        881
   0055                              FBUF(I)=SPIN(I)                                         882
   0056        99977               GO TO 99978                                             883
   0057        99976               CONTINUE                                                883
   0058                            LBFLG=.TRUE.                                            884
   0059                            FBUF(6)=BLANK                                           885
   0060        99980            CONTINUE                                                   886
   0061        99989            CONTINUE                                                   886
   0062        99992         CONTINUE                                                      887
   0063        99995      GO TO 99996                                                      888
   0064        99994      CONTINUE                                                         888
   0065                   READ(CRDR,20) SPIN                                               889
   0066           20      FORMAT(80A1)                                                     890
   0067                   IF(.NOT.( SPIN(1).EQ.EFF )) GO TO 99975                          891
   0068                      WRITE(PRTR,31)(SPIN(I),I=1,80)                                892
   0069            31        FORMAT(34H  -----BEGINNING OF FORTRAN-----  ,80A1)             893
   0070                      READ(CRDR,20)SPIN                                             894
   0071        99974         IF(.NOT.( SPIN(1).NE.EFF )) GO TO 99973                       895
   0072                         WRITE(PRTR,33)(SPIN(I),I=1,80)                             896
   0073            33           FORMAT(20X,80A1)                                           897
   0074                         WRITE(PFILE,32)(SPIN(I),I=1,80)                            898
   0075            32           FORMAT(80A1)                                               899
   0076                         READ(CRDR,20) SPIN                                         900
   0077        99972         GO TO 99974                                                   901
   0078        99973         CONTINUE                                                      901
   0079                      WRITE(PRTR,34) (SPIN(I),I=1,80)                               902
   0080            34        FORMAT(28H  -----END OF FORTRAN-----  ,80A1)                  903
   0081                      READ(CRDR,20)SPIN                                             904
   0082        99975      CONTINUE                                                         905
   0083                   TOEND=0                                                          906
```

```
0084        99998    GO TO 99999                                          907
0085        99997    CONTINUE                                             907
0086             END                                                      908
```

```
FORTRAN IV G LEVEL  21                    STTYP                DATE = 77326          20/47/11

0001                SUBROUTINE STTYP(TYPE)                                                    909
          C         //  THE TOKEN AT SPIN(TOBEG) TO SPIN(TCEND) IS SEARCHED                   910
          C         //    FOR IN THE SYMBOL TABLE.                                            911
          C         //                                                                        912
0002                IMPLICIT INTEGER(A-Z)                                                     913
0003                COMMON/INPUT/SPIN(80),TOBEG,TCEND,STBEG,STEND,IMAX,LBFLG                  914
0004                COMMON/STTYP2/COMT,CONT,FCTRN,BLANK,SEMI                                  915
0005                COMMON/TABLES/TRES(23),RESRD(200)                                         916
          C         //                                                                        917
0006                TOP=1                                                                     918
0007                BOTTOM=23                                                                 919
0008                LENGTH=TCEND-TOBEG+1                                                      920
0009      99999     IF(.NOT.( TOP.LE.BOTTOM )) GO TO 99998                                    921
0010                   MID=(TOP+BOTTOM)/2                                                     922
0011                   I=TRES(MID)                                                            923
0012                   IF(.NOT.( SPIN(TOBEG).LT.RESRD(I+1) )) GO TO 99995                     925
0013                      BOTTOM=MID-1                                                        926
0014                   GO TO 99996                                                            927
0015      99995        IF(.NOT.( SPIN(TOBEG).GT.RESRD(I+1) )) GO TO 99994                     927
0016                      TOP=MID+1                                                           928
0017                   GO TO 99996                                                            929
0018      99994        IF(.NOT.( SPIN(TOBEG).EQ.RESRD(I+1) )) GO TO 99993                     929
0019                      IF(.NOT.( LENGTH.EQ.1 )) GO TO 99992                                930
0020                         TYPE=RESRD(I+2)                                                  931
0021                         RETURN                                                           932
0022      99992           CONTINUE                                                           933
0023                      PNT=2                                                              934
0024      99991           IF(.NOT.( PNT.LE.RESRD(I).AND.PNT.LE.LENGTH )) GO TO 9999          925
               9          0                                                                  935
0025                         ITEMP=I+PNT                                                     936
0026                         JTEMP=TOBEG-1+PNT                                               937
0027                         IF(.NOT.( SPIN(JTEMP).LT.RESRD(ITEMP) )) GO TO 99987            939
0028                            BOTTOM=BOTTOM-1                                              940
0029                            PNT=9999                                                     941
0030                         GO TO 99988                                                     942
0031      99987              IF(.NOT.( SPIN(JTEMP).GT.RESRD(ITEMP) )) GO TO 99986            942
```

```
0032                              TCP=TCP+1                                          943
0033                              PNT=9999                                           944
0034                           GO TO 99988                                           945
0035        99986              IF(.NOT.( SPIN(JTEMP).EQ.RESRD(ITEMP) )) GO TO 99985  945
0036                              PNT=PNT+1                                          946
0037        99985              CONTINUE                                              947
0038        99988              CONTINUE                                              947
0039        99989           GO TO 99991                                              948
0040        99990           CONTINUE                                                 948
0041                        IF(.NOT.( PNT.NE.9999 )) GO TO 99984                     949
0042                           IF(.NOT.( LENGTH.EQ.RESRD(I) )) GO TO 99982           951
0043                              JTEMP=LENGTH+1+I                                   952
0044                              TYPE=RESRD(JTEMP)                                  953
0045                              RETURN                                             954
0046                           GO TO 99983                                           955
            $
**********01)  IEY002I LABEL*************************************************************
0047        99982              IF(.NOT.( LENGTH.GT.RESRD(I) )) GO TO 99981           955
0048                              TOP=TOP+1                                          956
0049                           GO TO 99983                                           957
0050        99981              IF(.NOT.( LENGTH.LT.RESRD(I) )) GO TO 99980           957
0051                              BOTTOM=BOTTOM-1                                    958
```

```
FORTRAN IV G LEVEL  21                  STTYP           DATE = 77326      20/47/11

0052        99980              CONTINUE                                        959
0053        99983              CONTINUE                                        959
0054        99984            CONTINUE                                          960
0055        99993          CONTINUE                                            961
0056        99996          CONTINUE                                            961
0057        99997       GO TO 99999                                            962
0058        99998       CONTINUE                                               962
0059                    TYPE=FOTRN                                             963
0060                    RETURN                                                 964
0061                END                                                        965
```

```
FORTRAN IV G LEVEL  21                  SCAN1                DATE = 77326          20/47/11

0001                SUBROUTINE SCAN1(CHAR,ECOND)                                          966
          C            //  LOOKS FOR CHAR AND IF FOUND, SPIN(TOEND+1)=CHAR               967
          C            //  AND ECOND=FALSE. IF NOT FOUND UPTO IMAX OR SEMICOLON          968
          C            //  ECOND=TRUE.                                                   969
          C            //                                                                970
0002                   IMPLICIT INTEGER(A-Z)                                             971
0003                   LOGICAL ECOND                                                     972
0004                   COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG          973
0005                   COMMON/STTYP2/COMT,CONT,FCTRN,BLANK,SEMI                          974
          C            //                                                                975
0006                   ECOND =.FALSE.                                                    976
0007          99999    IF(.NOT.( TOEND.LT.IMAX )) GO TO 99998                            977
0008                      ITEMP=SPIN(TOEND+1)                                            978
0009                      KTEMP=SPIN(TOEND)                                              979
0010                      IF(.NOT.( ITEMP.EQ.SEMI.AND.KTEMP.EQ.BLANK )) GO TO 99996      980
0011                         ECOND=.TRUE.                                                981
0012                         RETURN                                                      982
0013          99996          CONTINUE                                                    983
0014                         IF(.NOT.( CHAR.NE.SEMI )) GO TO 99995                       984
0015                            IF(.NOT.( ITEMP.EQ.CHAR )) GO TO 99994                   985
0016                               RETURN                                                986
0017          99994             CONTINUE                                                987
0018          99995          CONTINUE                                                    988
0019                         TOEND=TOEND+1                                               989
0020          99997    GO TO 99999                                                       990
0021          99998    CONTINUE                                                          991
0022                   ECOND = .TRUE.                                                    992
0023                   RETURN                                                            993
0024                END                                                                  994
```

```
FORTRAN IV G LEVEL  21                    SCAN                    DATE = 77326          20/47/11

0001                SUBROUTINE SCAN(PAT,LEN,ECOND)                                          994
                C       // THE SUBROUTINE COMPARES THE PATERN PAT WITH ALL TOKENS          995
                C       // SURSCROUNDED BY BLANKS FROM SPIN(TOEND+1) ONWARDS AND          996
                C       // RETURNS WITH TOEND AT THE OLD POSITION OF TOEND; WITH THE      997
                C       // PATTERN STARTING AT TOEND+1.  IF NOT FOUND ECOND IS SET TRUE   998
                C       //                                                                999
0002                    IMPLICIT INTEGER(A-Z)                                          1000
0003                    LOGICAL ECOND                                                  1001
0004                    COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG       1002
0005                    COMMON/STTYP2/COMT,CONT,FOTRN,BLANK,SEMI                       1003
0006                    DIMENSION PAT(6)                                               1004
                C       //                                                             1005
0007                    TOBEG=TOEND+1                                                  1006
0008            99999   CONTINUE                                                       1007
0009                       K=TOEND+1                                                   1008
0010            99996      IF(.NOT.( K.LE.IMAX.AND.SPIN(K).EQ.BLANK )) GO TO 99995     1009
0011                          K=K+1                                                    1010
0012            99994      GO TO 99996                                                 1011
0013            99995      CONTINUE                                                    1011
0014                       TOEND=K                                                     1012
0015                       CALL SCAN1(BLANK,ECOND)                                     1013
0016                       IF(.NOT.( ECOND )) GO TO 99993                              1014
0017                          RETURN                                                   1015
0018            99993      CONTINUE                                                    1016
0019                       IF(.NOT.( LEN.NE.TOEND-K+1 )) GO TO 99992                   1017
0020                          GO TO 99998                                              1018
0021            99992      CONTINUE                                                    1019
                C          // COMPARE THE PATTERN WITH A TOKEN OF THE SAME LENGTH.     1020
0022                        I=1                                                        1021
0023                       GO TO 99991                                                 1021
0024            99990       I= I+1                                                     1021
0025            99991      IF(.NOT.(( I-( LEN )).LE.0)) GO TO 99988                    1021
0026                          J=K+I-1                                                  1022
0027                          IF(.NOT.( PAT(I).NE.SPIN(J) )) GO TO 99987               1023
0028                             GO TO 99988                                           1024
0029            99987         CONTINUE                                                1025
```

```
0030                    IF(.NOT.( I.EQ.LEN )) GO TO 99986                       1026
     C                     // PATTERN FOUND, RETURN WITH ECOND FALSE.           1027
0031                       TCEND=TCEND-LEN                                      1028
0032                       RETURN                                               1029
0033           99986       CONTINUE                                             1030
0034           99989    GO TO 99990                                             1031
0035           99983    CONTINUE                                                1031
0036           99998 GO TO 99999                                                1032
0037           99997 CONTINUE                                                   1032
0038                 END                                                        1033
```

```
FORTRAN IV G LEVEL  21                    SPRNT                DATE = 77326          20/47/11

0001                SUBROUTINE SPRNT                                                          1034
0002                   IMPLICIT INTEGER(A-Z)                                                  1035
0003                   LOGICAL LBFLG,OVFL                                                     1036
0004                   COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG               1037
0005                   COMMON/STTYP2/CONT,CONT,FCTRN,BLANK,SEMI                               1038
0006                   COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX         1039
0007                   COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)                      1040
0008                   COMMON/NMNIC/CEE,PRTR,FFILE,CRDR                                       1041
0009                   COMMON/TYP/TYPE                                                        1042
               C       //                                                                    1043
0010                   IF(.NOT.( TYPE.EQ.CONT )) GO TO 99999                                  1044
0011                      BLNKS(6)=SPIN(6)                                                    1045
0012           99999   CONTINUE                                                              1046
0013           99998   CONTINUE                                                              1047
0014                      OVFL = .FALSE.                                                      1048
0015                      LEN1=STEND-STBEG+1                                                  1049
0016                      LEN2=SMAX-SSTRT+1                                                   1050
0017                      SEND=STEND                                                          1051
0018                      IF(.NOT.( LEN1.GT.LEN2 )) GO TO 99995                               1052
0019                         OVFL=.TRUE.                                                      1053
0020                         SEND=STBEG+LEN2-1                                                1054
0021           99995         CONTINUE                                                        1055
0022                         FIL=LEN2-LEN1+7                                                 1056
0023                         WRITE(PRTR,10) NESTNO,BLKNO,STMNO,(BLNKS(I),I=1,SSTRT),         1057
                    1        (SPIN(I),I=STBEG,SEND),(BLNKS(I),I=7,FIL),(SPIN(I),I=73       1058
                    2        ,80)                                                           1059
0024            10           FORMAT(1H ,I5,I5,I6,4X,111A1)                                  1060
0025                          I=1                                                            1061
0026                         GO TO 99994                                                     1061
0027           99993          I= I+1                                                         1061
0028           99994         IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99991                          1061
0029                            BLNKS(I)=BLANK                                               1062
0030           99992         GO TO 99993                                                    1063
0031           99991         CONTINUE                                                       1063
0032                         STBEG=SEND+1                                                    1064
0033           99997      IF(.NOT.( .NOT.OVFL )) GO TO 99998                                1065
```

```
0034        99996    CONTINUE                                          1065
0035                 RETURN                                            1066
0036               END                                                 1067
```

```
FORTRAN IV G LEVEL  21                  FHAND                DATE = 77326          20/47/11

0001                SUBROUTINE FHAND                                                        1068
                C      //   COMPLETES FORTRAN STATEMENTS AND OUTPUTS THE FORTRAN.        1069
                C      //                                                               1070
0002                   IMPLICIT INTEGER(A-Z)                                            1071
0003                   LOGICAL LBFLG,ECCND                                              1072
0004                   COMMON/INPUT/SPIN(80),TOBEG,TCEND,STBEG,STEND,IMAX,LBFLG         1073
0005                   COMMON/STTYP2/CCMT,CONT,FOTRN,BLANK,SEMI                         1074
0006                   COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)                1075
0007                   COMMON/TYP/TYPE                                                  1076
                C      //                                                               1077
0008            99999  CONTINUE                                                         1078
0009                      CALL SCAN1 (SEMI,ECCND)                                       1079
0010                      IF(.NOT.( TYPE.EQ.CONT )) GO TO 99996                         1080
0011                         FBUF(6)=SPIN(6)                                            1081
0012            99996     CONTINUE                                                      1082
0013                      CALL FOUT(SPIN,TOBEG,TCEND)                                   1083
0014                      IF(.NOT.( ECCND )) GO TO 99995                                1084
0015                         GO TO 99997                                                1085
0016            99995     CONTINUE                                                      1086
0017                      TOEND=TOEND+1                                                 1087
0018                      IF(.NOT.( SPIN(TCEND+1).NE.SEMI )) GO TO 99994                1088
0019                         GO TO 99997                                                1089
0020            99994     CONTINUE                                                      1090
0021                      TOBEG=TCEND+1                                                 1091
0022                      TOEND=TOBEG                                                   1092
0023            99998  GO TO 99999                                                      1093
0024            99997  CONTINUE                                                         1093
0025                   CALL FPRNT                                                       1094
0026                   RETURN                                                           1095
0027                END                                                                 1096
```

```
FORTRAN IV G LEVEL  21                    FOUT                    DATE = 77326          20/47/11

0001                  SUBROUTINE FOUT(PAT,STRT,END)                                          1097
                C        // THE VALUE OF FSTRT WILL USUALLY BE 7 AND FMAX WILL BE 72.       1098
                C        //  FPTR SHOULD BE AT FSTRT.  BLNKLN IS A FLAG TO INDICATE A      1099
                C        // TO INDICATE A SC FAR BLANK LINE.                               1100
                C        //                                                                1101
0002                     IMPLICIT INTEGER(A-Z)                                             1102
0003                     LOGICAL LBFLG,BLNKLN                                              1103
0004                     COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),     1104
                       1 TPITR,NELTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB                 1105
0005                     COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)                 1106
0006                     COMMON/STTYP2/COMT,CONT,FCTRN,BLANK,SEMI                          1107
0007                     COMMON/MNMIC/CEE,PRTR,FFILE,CRDR                                  1108
0008                     COMMON/TYP/TYPE                                                   1109
0009                     COMMON/FORPRT/BLNKLN                                              1110
0010                     DIMENSION PAT(1)                                                  1111
0011                     DATA NINE/1H9/                                                    1112
                C        //                                                                1113
0012                     BLNKLN = .FALSE.                                                  1114
0013                      J=STRT                                                           1115
0014                     GO TO 99999                                                       1115
0015            99998     J= J+1                                                           1115
0016            99999    IF(.NOT.(( J-( END )).LE.0)) GO TO 99996                          1115
0017                        IF(.NOT.( FPTR.GT.FMAX )) GO TO 99995                          1116
0018                           CALL FPRNT                                                  1117
0019                            I=1                                                        1118
0020                           GO TO 99994                                                 1118
0021            99993           I= I+1                                                     1118
0022            99994           IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99991                     1118
0023                              FBUF(I)=LABEL(I)                                         1119
0024            99992           GO TO 99993                                              1120
0025            99991           CONTINUE                                                 1120
0026                            IF(.NOT.( TYPE.EQ.COMT )) GO TO 99990                    1121
0027                              FBUF(1)=CEE                                            1122
0028            99990           CONTINUE                                                 1123
0029                            JJ=J                                                     1124
0030            99989           IF(.NOT.( PAT(JJ).EQ.BLANK )) GO TO 99988                1125
```

```
0031                        IF(.NOT.( JJ.EQ.END )) GO TO 99986                    1126
0032                           BLNKLN=.TRUE.                                      1127
0033                           RETURN                                             1128
0034        99986              CONTINUE                                           1129
0035                           JJ=JJ+1                                            1130
0036        99987           GO TO 99989                                           1131
0037        99988           CONTINUE                                              1131
0038                        FBUF(6)=NINE                                          1132
0039        99995        CONTINUE                                                 1133
0040                     FBUF(FPTR)=PAT(J)                                        1134
0041                     FPTR=FPTR+1                                              1135
0042        99997     GO TO 99998                                               1136
0043        99996     CONTINUE                                                  1136
0044                  RETURN                                                    1137
0045              END                                                           1138
```

```
FORTRAN IV G LEVEL  21                    FPRNT                    DATE = 77340          12/15/02

0001                SUBROUTINE FPRNT                                                          1139
          C         // THE SUBROUTINE PRINTS THE TRANSLATED FORTRAN                           1140
0002                IMPLICIT INTEGER(A-Z)                                                     1141
0003                LOGICAL BLNKLN                                                            1142
0004                COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG                  1143
0005                COMMON/NMNIC/CEE,PRTR,FFILE,CRDR                                          1144
0006                COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX            1145
0007                COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)                         1146
0008                COMMON/FCRPRT/BLNKLN                                                      1147
          C         //                                                                        1148
0009                IF(.NOT.( FPTR.EQ.FSTRT )) GO TO 99999                                    1149
0010                   RETURN                                                                 1150
0011          99999 CONTINUE                                                                  1151
0012                IFF=FSTRT-1                                                               1152
0013                IF(.NOT.( IFF.GE.7 )) GO TO 99998                                         1153
0014                    I=7                                                                   1154
0015                   GO TO 99997                                                            1154
0016          99996     I= I+1                                                                1154
0017          99997     IF(.NOT.(( I-( IFF )).LE.0)) GO TO 99994                              1154
0018                       FBUF(I)=BLNKS(I)                                                   1155
0019          99995     GO TO 99996                                                           1156
0020          99994     CONTINUE                                                              1156
0021          99998 CONTINUE                                                                  1157
0022                IF(.NOT.( FPTR.LE.FMAX )) GO TO 99993                                     1158
0023                    I=FPTR                                                                1159
0024                   GO TO 99992                                                            1159
0025          99991     I= I+1                                                                1159
0026          99992     IF(.NOT.(( I-( FMAX )).LE.0)) GO TO 99989                             1159
0027                       FBUF(I)=BLNKS(I)                                                   1160
0028          99990     GO TO 99991                                                           1161
0029          99989     CONTINUE                                                              1161
0030          99993 CONTINUE                                                                  1162
0031                 I=73                                                                     1163
0032                GO TO 99988                                                               1163
0033          99987  I= I+1                                                                   1163
0034          99988 IF(.NOT.(( I-( 75 )).LE.0)) GO TO 99985                                   1163
```

```
0035                    FBUF(I)=SPIN(I)                                    1164
0036          99986     GO TO 99987                                        1165
0037          99985     CONTINUE                                           1165
0038                    IF(.NOT.( .NOT.BLANKLN )) GO TO 99984              1166
0039                    WRITE(FFILE,10)(FBUF(I),I=1,75),STMNO              1167
0040             10     FORMAT(75A1,I5)                                    1168
0041          99984     CONTINUE                                           1169
0042                    FPTR=FSTRT                                         1170
0043                    RETURN                                             1171
0044                  END                                                  1172
```

```
FORTRAN IV G LEVEL  21                    ISNUM                DATE = 77326          20/47/11

0001                 INTEGER FUNCTION ISNUM(Q)                                              1173
           C            // THE FUNCTION CHECKS TO SEE IF Q IS THE ALPHANUMERIC             1174
           C            // REPRESENTATION OF A NUMBER AND RETURNS 1 IF IT IS AND 0 IF      1175
           C            // IT IS NOT.                                                      1176
           C            //                                                                 1177
0002                    IMPLICIT INTEGER(A-Z)                                             1178
0003                    DIMENSION NUMBER(10)                                              1179
0004                    DATA NUMBER/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/              1180
           C            //                                                                 1181
0005                     I=1                                                              1182
0006                    GO TO 99999                                                       1182
0007           99998     I= I+1                                                           1182
0008           99999    IF(.NOT.(( I-( 10 )).LE.0)) GO TO 99996                          1182
0009                       IF(.NOT.( Q.EQ.NUMBER(I) )) GO TO 99995                       1183
0010                          ISNUM=1                                                    1184
0011                          RETURN                                                     1185
0012           99995       CONTINUE                                                      1186
0013           99997    GO TO 99998                                                      1187
0014           99996    CONTINUE                                                         1187
0015                    ISNUM=0                                                          1188
0016                    RETURN                                                           1189
0017                 END                                                                 1190
```

```
FORTRAN IV G LEVEL  21                    CONV                DATE = 77326          20/47/11

0001                INTEGER FUNCTION CONV(DIGIT)                                          1191
                C       // CONVERTS THE DIGIT INTO ITS CHARACTER REPRESENTATION.        1192
                C       //                                                             1193
0002                    IMPLICIT INTEGER(A-Z)                                          1194
0003                    DIMENSION CVNUM(10)                                            1195
0004                    DATA CVNUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/            1196
0005                    CONV=CVNUM(DIGIT+1)                                            1197
0006                    RETURN                                                         1198
0007                END                                                                1199
```

```
FORTRAN IV G LEVEL  21                GENLAB          DATE = 77326          20/47/11

0001                INTEGER FUNCTION GENLAB(GEN)                                    1200
          C         //  GENERATE A LABEL UNCONDITIONALLY IF GEN IS 1 ELSE          1201
          C         //  GENERATE THE LABEL IF LBFLG=0, IF LABEL FLAG               1202
          C         //  EQUAL 1 AND GEN EQUAL 0 THEN USE THE EXISTING              1203
          C         //  LABEL. NUMB IS INITIALIZED TO 100000. THIS IS AN           1204
          C         //  INTEGER FUNCTION RETURNING A POINTER TO THE LABEL.         1205
          C         //                                                             1206
0002                IMPLICIT INTEGER(A-Z)                                           1207
0003                LOGICAL LBFLG                                                   1208
0004                COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG        1209
0005                COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),   1210
               1    TPITR,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB               1211
          C         //                                                             1212
0006                IF(.NOT.( .NOT.LBFLG.OR.GEN.EQ.1 )) GO TO 99999                 1213
0007                   NUMB=NUMB-1                                                  1214
0008                   N=NUMB                                                       1215
0009                    I=1                                                         1216
0010                   GO TO 99998                                                  1216
0011          99997     I= I+1                                                      1216
0012          99998     IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99995                      1216
0013                      M=N/10                                                    1217
          C               //  SHIFT RIGHT                                         1218
0014                      NR=N-(10*M)                                               1219
          C               //  REMAINDER                                           1220
0015                      HOLD=5-I+LBPTR                                            1221
0016                      LABEL(HOLD)=CONV(NR)                                      1222
          C               //  LBPTR IS LEFT AT FIRST BLANK SPACE.                 1223
0017                      N=M                                                       1224
0018          99996     GO TO 99997                                                 1225
0019          99995     CONTINUE                                                    1225
0020                    GENLAB=LBPTR                                                1226
          C             //  NEW LABEL PTR.                                        1227
0021                    LBPTR=LBPTR+5                                               1228
          C             //  POINT TO NEW BLANK LOCATION.                          1229
0022                    IF(.NOT.( LBPTR.GT.LBMAX )) GO TO 99994                     1230
0023                      CALL ERROR(8,0,0,5,0,0)                                   1231
```

```
                C                  //  NESTING TOO DEEP, TOC MANY LABELS PROGRAM TERMINATED.    1232
0024            99994           CONTINUE                                                        1233
0025                            GO TO 99993                                                     1234
0026            99999       CONTINUE                                                            1234
0027                             I=1                                                            1235
0028                            GO TO 99992                                                     1235
0029            99991            I= I+1                                                         1235
0030            99992           IF(.NOT.(( I-( 5 )).LE.0)) GO TO 99989                          1235
0031                               HOLD=LBPTR+I-1                                               1236
0032                               LABEL(HOLD)=SPIN(I)                                          1237
0033            99990           GO TO 99991                                                     1238
0034            99989           CONTINUE                                                        1238
0035                            GENLAB=LBPTR                                                    1239
0036                            LBPTR=LBPTR+5                                                   1240
0037                            IF(.NOT.( LBPTR.GT.LBMAX )) GO TO 99988                         1241
0038                               CALL ERROR(8,0,0,5,0,0)                                      1242
0039            99988           CONTINUE                                                        1243
                C                  //  NESTING TOO DEEP, TOC MANY LABELS PROGRAM TERMINATED.    1244
0040            99993       CONTINUE                                                            1245
0041                        RETURN                                                              1246
0042                    END                                                                     1247
```

```
0001              SUBROUTINE CHECK(STYPE,ECCND)                                    1248
0002                 IMPLICIT INTEGER(A-Z)                                         1249
0003                 LOGICAL ECCND,DUMM,LBFLG                                      1250
0004                 COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG      1251
0005                 COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)             1252
0006                 COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX 1253
0007                 COMMON/CTPAT/CGOTO(9),CONTI(8),EQL(1),GOTO(6),IFNOT(9),LEQ(5), 1254
                1    MINUS(1),CNE(1),PLU(1),RSTAR(2)                               1255
0008                 COMMON/STTYP1/CCASE,COLCN,DBLSL,EELSE,EEND,ECJ,EEXIT,FFOR,IIF, 1256
                1    LLOOP,RREPT,UNTL,WHLE,SUBFN,RINLO,NCYCLE,ENDCAS,EENDIF        1257
0009                 COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),  1258
                1    TPITR,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB             1259
0010                 COMMON/INPAT/CCLN(1),ELZ(4),DBSLH(2),DCO(2),BYE(2),THENN(4),TOO 1260
                9    (2)                                                           1260
0011                 COMMON/TABLES/TRES(23),RESRD(200)                             1261
0012                 COMMON/KEYWD/EEEND(3),EEEXT(4),CCYCLE(5),UUNTL(5),CASS(4),IIIF( 1262
                9    2),                                                           1262
                1    ITERAT(9),ENCASS(7),ENIF(5),REPT(6),CCOLON(5)                 1263
          C          //                                                            1264
0013                 ECCND=.FALSE.                                                 1265
0014                 IF(.NOT.( TOP.EQ.0 )) GO TO 99999                             1266
0015                    CALL ERROR(4,0,0,3,0,0)                                    1267
          C             // ERROR---STACK EMPTY, STATEMENT IGNORED.                 1268
0016                    ECCND=.TRUE.                                               1269
0017                    RETURN                                                     1270
0018              99999 CONTINUE                                                   1271
0019                 STTOP=STACK(TOP)                                              1272
0020                 IF(.NOT.( STYPE.EQ.COLCN )) GO TO 99997                       1274
0021                    IF(.NOT.( STTOP.NE.CCASE )) GO TO 99996                    1275
0022                       CALL ERROR(2,CASS,4,4,CCOLCN,5)                         1276
          C                // ERROR---MISSING CASE STATEMENT, CCLCN IGNORED.       1277
0023                       TOEND=TOEND+1                                           1278
0024                       ECCND=.TRUE.                                            1279
0025              99996    CONTINUE                                                1280
0026                 GO TO 99998                                                   1281
0027              99997 IF(.NOT.( STYPE.EQ.EELSE )) GO TO 99995                    1281
```

```
0028                     IF(.NOT.( STTOP.NE.IIF )) GO TO 99994                    1282
0029                        CALL ERROR(2,IIIF,2,4,ELZ,4)                          1283
          C                 // ERROR---MISSING IF STATEMENT, ELSE IGNORED.        1284
0030                        ECOND=.TRUE.                                          1285
0031      99994          CONTINUE                                                 1286
0032                     GO TO 99998                                              1287
0033      99995          IF(.NOT.( STYPE.EQ.UNTL )) GO TO 99993                   1287
0034                        IF(.NOT.( STTOP.NE.LLOOP )) GO TO 99992               1288
0035                        CALL ERROR(5,ITERAT,9,4,ULNTL,5)                      1289
          C                 // ERROR---MISSING LOOP STATEMENT, UNTIL IGNORED.     1290
0036                        CALL SCAN(REPT,3,DUMM)                                1291
0037                        IF(.NOT.( DUMM )) GO TO 99991                         1292
0038                           TOEND=TOEND+1                                      1293
0039                           GO TO 99990                                        1294
0040      99991             CONTINUE                                              1294
0041                           TOEND=TOEND+6                                      1295
0042      99990             CONTINUE                                              1296
0043                        ECOND=.TRUE.                                          1297
0044      99992          CONTINUE                                                 1298
0045                     GO TO 99998                                              1299
0046      99993          IF(.NOT.( STYPE.EQ.EEND )) GO TO 99989                   1299
0047                        IF(.NOT.( STTOP.NE.SUBFN )) GO TO 99988               1300
```

```
      C              // UNSTACK UNTIL A SUBFN IS FOUND ON THE STACK.                1301
0048                 TTOP=TOP                                                      1302
0049        99987    CONTINUE                                                      1303
0050                    TTOP=TTOP-STFRM(STTOP)                                     1304
0051                    IF(.NOT.( TTOP.LE.0 )) GO TO 99984                         1305
0052                       CALL ERROR(3,EEEND,3,4,EEEND,3)                         1306
      C                    // ERROR---TOO MANY ENDS, END IGNORED.                 1307
0053                       ECOND=.TRUE.                                            1308
0054                       RETURN                                                  1309
0055        99984       CONTINUE                                                   1310
0056                    STTOP=STACK(TTOP)                                          1311
0057        99986    IF(.NOT.( STTOP.EQ.SUBFN )) GO TO 99987                       1312
0058        99985    CONTINUE                                                      1312
0059                 TOP=TTOP                                                      1313
0060                 CALL ERROR(7,0,0,6,0,0)                                       1314
      C              // PREMATURE END SUBPROGRAM CLOSED.                           1315
0061                 FSTRT=7                                                       1316
0062                 SSTRT=7                                                       1317
0063        99988    CONTINUE                                                      1318
0064              GO TO 99998                                                      1319
0065        99989 IF(.NOT.( STYPE.EQ.ENDCAS )) GO TO 99983                       1319
0066                 IF(.NOT.( STTOP.NE.CCASE )) GO TO 99982                       1320
0067                    CALL ERROR(3,ENCASS,7,4,ENCASS,7)                          1321
      C                 // ERROR---TOO MANY ENDCASES, ENDCASE IGNORED.            1322
0068                    ECOND=.TRUE.                                               1323
0069        99982    CONTINUE                                                      1324
0070              GO TO 99998                                                      1325
0071        99983 IF(.NOT.( STYPE.EQ.EENDIF )) GO TO 99981                       1325
0072                 IF(.NOT.( STTOP.NE.IIF )) GO TO 99980                         1326
0073                    CALL ERROR(3,ENIF,5,4,ENIF,5)                              1327
      C                 // ERROR---TOO MANY ENDIFS, ENDIF IGNORED.                1328
0074                    ECOND=.TRUE.                                               1329
0075        99980    CONTINUE                                                      1330
0076              GO TO 99998                                                      1331
0077        99981 IF(.NOT.( STYPE.EQ.RREPT )) GO TO 99979                        1331
0078                 IF(.NOT.( STTOP.EQ.FFOR.OR.STTOP.EQ.LLOOP.OR.STTOP.EQ.WHLE )  1332
```

```
                9       ) GO TO 99978                                                    1332
   0079                    RETURN                                                       1333
   0080                    GO TO 99977                                                  1334
                $
**********01)  IEY002I LABEL*************************************************************
   0081         99978      CONTINUE                                                     1334
   0082                    CALL ERROR(5,ITERAT,9,4,REPT,6)                              1335
                C          // ERROR---NO ITERATIVE STATEMENT FOUND, REPEAT IGNORED.     1336
   0083                    ECOND=.TRUE.                                                 1337
   0084         99977      CONTINUE                                                     1338
   0085                  GO TO 99998                                                    1339
   0086         99979    IF(.NOT.( STYPE.EQ.EEXIT )) GO TO 99976                        1339
   0087                    IF(.NOT.( TPITR.EQ.0 )) GO TO 99975                          1340
   0088                    CALL ERROR(5,ITERAT,9,4,EEEXT,4)                             1341
                C          // ERROR---NO ITERATIVE STATEMENT, EXIT IGNORED.            1342
   0089                    ECOND=.TRUE.                                                 1343
   0090         99975      CONTINUE                                                     1344
   0091                  GO TO 99998                                                    1345
   0092         99976    IF(.NOT.( STYPE.EQ.NCYCLE )) GO TO 99974                       1345
   0093                    IF(.NOT.( TPITR.EQ.0 )) GO TO 99973                          1346
   0094                    CALL ERROR(5,ITERAT,9,4,CCYCLE,5)                            1347
                C          // ERROR---NO ITERATIVE STATEMENT, CYCLE IGNORED.           1348
```

```
0095                        ECOND=.TRUE.                                    1349
0096          99973       CONTINUE                                        1350
0097          99974     CONTINUE                                          1351
0098          99998     CONTINUE                                          1351
0099                    RETURN                                            1352
0100                  END                                                 1353
```

```
FORTRAN IV G LEVEL  21                    ERROR                DATE = 77326          20/47/11

0001                SUBROUTINE ERROR(CODE1,PARM1,LEN1,CODE2,PARM2,LEN2)                          1354
                C       //  CODE1 IS ERROR MESSAGE CODE                                        1355
                C       //  CODE2 IS THE ACTION TAKEN                                          1356
                C       //  PARM1 AND PARM2 MODIFY THE TWO                                     1357
                C       //  LEN1 AND LEN2 ARE THE LENGTH OF MODIFYING PATTERN                  1358
                C       //                                                                     1359
0002                    IMPLICIT INTEGER(A-Z)                                                  1360
0003                    COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG               1361
0004                    COMMON/STTYP2/COMT,CONT,FOTRN,BLANK,SEMI                               1362
0005                    COMMON/NMNIC/CEE,PRTR,FFILE,ORDR                                       1363
0006                    DIMENSION PARM1(1),PARM2(1),ERPAT(45,8),POSN(8),LEN(8),                1364
                    1   ACPAT(35,6),ALEN(6),APSN(6)                                            1365
0007                   DATA ERPAT/1HL,1HA,1HE,1HE,1HL,1HL,1HI,1HN,1HG,1H ,1HT,1HH,1HE,
                    1  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1HH,1HA,1HM,1HP,1HE,1HR,1HS,
                    1  1HP,1HR,1HO,1HG,1HR,1HA,1HM,1H ,1HC,1HL,1HA,1HR,1HI,1HT,1HY,1H.,
                    1  1H ,1H ,
                    2  1HM,1HI,1HS,1HS,1HI,1HN,1HG,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    2  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    2  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    3  1HD,1HA,1HN,1HG,1HL,1HI,1HN,1HG,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    3  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    3  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    4  1HS,1HT,1HA,1HC,1HK,1H ,1HE,1HM,1HP,1HT,1HY,1H.,1H ,1H ,1H ,
                    4  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    4  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    5  1HN,1HO,1H ,1HC,1HC,1HR,1HR,1HE,1HS,1HP,1HO,1HN,1HD,1HI,1HN,1HG,
                    5  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1HS,1HT,1HM,1HT,1H.,
                    5  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
                    6  1HS,1HT,1HA,1HC,1HK,1H ,1HF,1HU,1HL,1HL,
                    7  35*1H ,1HP,1HR,1HE,1HM,1HA,1HT,1HU,1HR,1HE,1H ,1HE,1HN,1HD,
                    8  32*1H ,1HT,1HC,1HC,1H ,1HM,1HA,1HN,1HY,1H ,1HL,1HA,1HB,1HE,1HL,
                    8  1HS,30*1H /
0008                   DATA LEN/5,6,6,0,9,C,0,C/
0009                   DATA POSN/15,9,10,0,18,0,0,0/
0010                   DATA ACPAT/1HS,1HU,1HP,1HP,1HL,1HI,1HE,1HD,1H ,1HA,1HT,
                    1  1HE,1HN,1HD,1H ,1HO,1HF,1H ,1HS,1HT,1HM,1HT,1H.,
```

```
     1  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2  1HL,1HA,1HB,1HE,1HL,1H ,1HC,1HE,1HL,1HE,1HT,1HE,1HD,
     2  1HF,1HR,1HG,1HM,1H ,1HF,1HC,1HR,1HT,1HR,1HA,1HN,1H ,
     2  1HO,1HU,1HT,1HP,1HU,1HT,1H.,1H ,1H ,
     3  1HS,1HT,1HA,1HT,1HE,1HM,1HE,1HN,1HT,1H ,
     3  1HS,1HK,1HI,1HP,1HP,1HE,1HD,1H.,1H ,1H ,1H ,1H ,1H ,
     3  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1HI,1HG,1HN,1HC,1HR,1HE,1HD,
     4  1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4  1H ,1H ,1H ,1H ,1H ,1H ,
     5  1HP,1HR,1HO,1HG,1HR,1HA,1HM,1H ,1HT,1HE,1HR,1HM,1HI,1HN,1HA,
     5  1HT,1HE,1HD,17*1H ,
     6  1HS,1HU,1HB,1HP,1HR,1HC,1HG,1HR,1HA,1HM,1H ,1HC,1HL,1HO,1HS,
     6  1HE,1HD,18*1H /
0011      DATA ALEN/0,0,0,6,0,0/
0012      DATA APSN/0,0,0,1,0,0/
      C    //                                                          1366
0013      WRITE(PRTR,2222) SPIN                                       1367
0014   2222 FORMAT(26H    ++++++OFFENDING CARD: ,80A1)                 1368
0015      LENTH=LEN(CODE1)                                            1369
0016      ALNTH=ALEN(CODE2)                                           1370
0017      POS=POSN(CODE1)                                             1371
```

```
0018                  APCS=APSN(CCDE2)                                          1372
0019                   I=1                                                      1373
0020                  GC TO 99999                                               1373
0021        99998      I= I+1                                                   1373
0022        99999     IF(.NOT.(( I-( LEN1 )).LE.0)) GO TO 99996                 1373
0023                     IT=PCS+I-1                                             1374
0024                     ERPAT(IT,CCDE1)=PARM1(I)                               1375
0025        99997     GO TO 99998                                               1376
0026        99996     CONTINUE                                                  1376
0027                  IF(.NOT.( LENTH.GT.LEN1 )) GO TO 99995                    1377
0028                     PC=PCS+LEN1                                            1378
0029                     PE=PCS+LENTH-1                                         1379
0030                      I=PC                                                  1380
0031                     GC TC 99994                                            1380
0032        99993        I= I+1                                                1380
0033        99994        IF(.NCT.(( I-( PE )).LE.0)) GO TO 99991               1380
0034                        ERPAT(I,CODE1)=BLANK                                1381
0035        99992        GO TC 99993                                           1382
0036        99991        CONTINUE                                              1382
0037        99995     CONTINUE                                                  1383
0038                   I=1                                                      1384
0039                  GO TC 99990                                               1384
0040        99989      I= I+1                                                   1384
0041        99990     IF(.NOT.(( I-( LEN2 )).LE.C)) GC TO 99987                 1384
0042                     IT=APCS+I-1                                            1385
0043                     ACPAT(IT,CCDE2)=PARM2(I)                               1386
0044        99988     GC TO 99989                                               1387
0045        99987     CONTINUE                                                  1387
0046                  IF(.NOT.( ALENTH.GT.LEN2 )) GO TO 99986                   1388
0047                     PC=APCS+LEN2                                           1389
0048                     PE=APCS+ALNTH-1                                        1390
0049                      I=PC                                                  1391
0050                     GC TC 99985                                            1391
0051        99984        I= I+1                                                1391
0052        99985        IF(.NOT.(( I-( PE )).LE.0)) GO TO 99982               1391
0053                        ACPAT(I,CODE2)=BLANK                                1392
```

```
0054        99983       GO TO 99984                                                    1393
0055        99982       CONTINUE                                                       1393
0056        99986    CONTINUE                                                          1394
0057                 WRITE(PRTR,10)   (ERPAT(I,CODE1),I=1,45)                          1395
0058                 WRITE (PRTR,10)     (ACPAT(I,CODE2),I=1,35)                       1396
0059          10     FORMAT(12H     +++++++,80A1)                                      1397
0060                 RETURN                                                           1398
0061              END                                                                 1399
```

```
0001                SUBROUTINE TYHAND(TYPE,EEEXT,CCYCLE,EEXIT)                              1400
0002                   IMPLICIT INTEGER(A-Z)                                                1401
0003                   LOGICAL LBFLG                                                        1402
0004                   COMMON/INPUT/SPIN(80),TOBEG,TOEND,STBEG,STEND,IMAX,LBFLG             1403
0005                   COMMON/SPRINF/SSTRT,INCR,STMNO,BLKNO,NESTNO,INDFLG,CURFLG,SMAX       1404
0006                   COMMON/FRTINF/FBUF(80),FPTR,FMAX,FSTRT,BLNKS(131)                    1405
0007                   COMMON/CTPAT/CGOTO(9),CONTI(8),EQL(1),GOTO(6),IFNOT(9),LEO(5),       1406
                    1  MINUS(1),ONE(1),PLU(1),RSTAR(2)                                     1407
0008                   COMMON/CENTRL/STACK(200),LABEL(200),TOP,LBMAX,LBPTR,STFRM(6),        1408
                    1  TPITR,NEUTR,BKWRD,FRWRD,NEW,OLD,PULOUT,STMAX,NUMB                   1409
0009                   COMMON/STTYP2/CCMT,CONT,FCTRN,BLANK,SEMI                             1410
                C      //                                                                   1411
0010                   IF(.NOT.( LBFLG )) GO TO 99999                                       1412
0011                      IF(.NOT.( TYPE.EQ.EEXIT )) GO TO 99998                            1413
0012                         CALL ERROR(1,EEEXT,4,2,0,0)                                    1414
                C            // ERROR--LABELLING THE EXIT HAMPERS PROGRAM CLEANLINESS-      1415
                C    9       -                                                             1415
                C            // --DELETED FROM THE FORTRAN OUTPUT.                         1416
0013                         GO TO 99997                                                    1417
0014            99998     CONTINUE                                                         1417
0015                         CALL ERROR(1,CCYCLE,5,2,0,0)                                   1418
                C            // ERROR---LABELLING THE CYCLE HAMPERS PROGRAM                 1419
                C            //  CLEANLINESS, DELETED FROM FORTRAN OUTPUT.                 1420
0016            99997     CONTINUE                                                         1421
0017            99999     CONTINUE                                                         1422
                C         //  CHECK IF A NUMERIC LABEL FOLLOWS.                            1423
0018                      K=TOEND+1                                                        1424
0019            99996     CONTINUE                                                         1425
0020                         IF(.NOT.( K.GT.IMAX )) GO TO 99993                            1426
0021                            I=0                                                        1427
0022                            GO TO 99994                                                1428
0023            99993        CONTINUE                                                      1429
0024                         IF(.NOT.( SPIN(K).NE.BLANK )) GO TO 99992                     1430
0025                            I=ISNUM(SPIN(K))                                           1431
0026                            GO TO 99994                                                1432
0027            99992        CONTINUE                                                      1433
```

```
0028                    K=K+1                                                  1434
0029          99995   GO TO 99996                                            1435
0030          99994   CONTINUE                                               1435
0031                  IF(.NOT.( I.EQ.0 )) GO TO 99991                        1436
0032                    I=1                                                  1437
0033                    GO TO 99990                                          1437
0034          99989     I= I+1                                               1437
0035          99990     IF(.NOT.(( I-( 6 )).LE.0)) GO TO 99987               1437
0036                      FBUF(I)=BLANK                                      1438
0037          99988     GO TO 99989                                          1439
0038          99987     CONTINUE                                             1439
0039                    IF(.NOT.( TYPE.EQ.EEXIT )) GO TO 99986               1440
0040                      K=STACK(TPITR-1)                                   1441
0041                      GO TO 99985                                        1442
0042          99986     CONTINUE                                             1442
0043                      K=STACK(TPITR-2)                                   1443
0044          99985     CONTINUE                                             1444
0045                    CALL FOUT(GOTO,1,6)                                  1445
0046                    CALL FOUT(LABEL,K,K+4)                               1446
0047                    CALL FPRNT                                           1447
0048                    GO TO 99984                                          1448
0049          99991   CONTINUE                                               1448
```

```
0050                    TCENC=K                                                    1449
0051                    CALL SCAN1(BLANK,ECCNC)                                    1450
0052                    WRITE(PRTR,4999)                                           1451
0053          4999      FCRMAT( '++++++++++LAEELLED EXIT-CYCLE NCT IMPLEMENTED')   1452
0054         99984    CONTINUE                                                     1453
0055                  STEND=TCEND                                                  1454
0056                  INDFLG=NEUTR                                                 1455
0057                  CURFLG=CLD                                                   1456
0058                  RETURN                                                       1457
0059                END                                                          1458
```

TRANSLATION OF THE SPARKS PREPROCESSOR FROM FORTRAN TO SPARKS

by

RICHARD MANSON STROUD

B.S., UNIVERSITY OF TEXAS, ARLINGTON, 1972

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1978

This report describes the translation of the SPARKS preprocessor from FORTRAN to SPARKS. The SPARKS preprocessor is a program which translates other programs written in the SPARKS language to a target language of FORTRAN.

The SPARKS language adds additional constructs to FORTRAN so that structured code can be written while maintaining a close tie to FORTRAN. The preprocessor written in SPARKS is a structured program with the necessary modularity so that changes and up-dates can be made easily. The preprocessor in SPARKS was examined in detail for code that could be improved. Approximately 375 lines of code were found to be unnecessary and were eliminated. The rewritten preprocessor is a 1470 line program versus the 1845 lines of the original FORTRAN version. The report includes updated documentation and user's guide for the SPARKS language.