

AD-A092 569

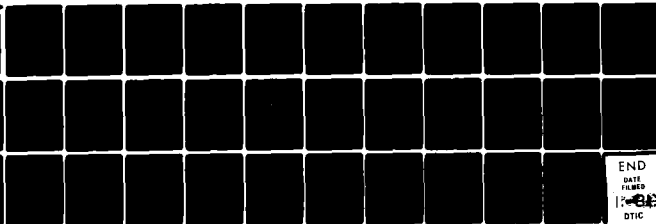
NAVAL RESEARCH LAB WASHINGTON DC
CAL98: A FORTRAN-IV CALCULATOR EMULATION PROGRAM. (U)
NOV 80 D F BIRN, J C COOPER
NRL-MR-4377

F/6 9/2

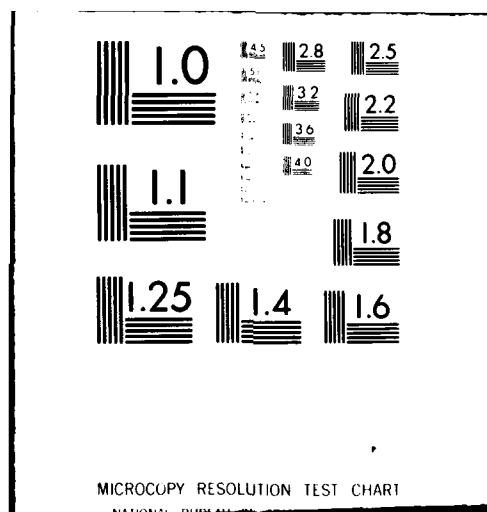
UNCLASSIFIED

NL

For
9/2/64



END
DATE
FILMED
1-82
DTIC





DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

⑨ Final rept.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ⑭ <u>NRL-MR-</u> NRL Memorandum Report <u>4377</u>	2. GOVT ACCESSION NO. <u>AD-A092</u>	3. RECIPIENT'S CATALOG NUMBER <u>569</u>
4. TITLE (and Subtitle) ⑥ <u>CALQS: A FORTRAN-IV CALCULATOR EMULATION PROGRAM</u>		5. TYPE OF REPORT & PERIOD COVERED A final report on one phase of the problem.
7. AUTHOR(s) ⑩ <u>Daniel F. Birn and John C. Cooper</u>		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS <u>Naval Research Laboratory</u> <u>Washington, D.C. 20375</u> ⑫ <u>38</u>		8. CONTRACT OR GRANT NUMBER(s) ⑪ <u>18 Nov 80</u>
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS <u>61153N; RR0240341;</u> <u>61-0027-0-0</u>
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE <u>November 18, 1980</u>
		13. NUMBER OF PAGES <u>37</u>
		15. SECURITY CLASS. (of this report) <u>UNCLASSIFIED</u>
16. DISTRIBUTION STATEMENT (of this Report) <u>Approved for public release; distribution unlimited</u>		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) ⑬ <u>KK0240341</u>		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <u>Computer program</u> <u>Calculator</u> <u>Simulation</u> <u>FORTRAN</u>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <u>CALQS is a FORTRAN-IV program which emulates a multi-functional arithmetic calculator on a mini-computer. Extensive calculation power is provided without requiring any programming skill. This program was designed for and implemented on a Hewlett-Packard model 1000 mini-computer with an RTE operating system. It contains substantial error checking and supports a wide variety of functions not available on hand-held calculators.</u>		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LP-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

1. INTRODUCTION	1
2. FEATURES OF CALQ8	2
3. ACCURACY	4
4. LIMITATIONS	4
5. SOURCE AVAILABILITY	5
6. PROGRAM LOGIC FLOW DESCRIPTION	5
7. ACKNOWLEDGMENT	10
8. REFERENCES	11
9. SAMPLE PROGRAM DIALOG	12
10. FLOWCHART OF CALQ8	14
11. VARIABLE LIST	21
12. PROGRAM LISTING	24

Accession For	
NTIS GMLI	☒
DDC TAB	☐
Unrecorded	☐
Unrecorded	☐
By	
Date	
For	
Special	
A	23 cf.

1. Introduction

Before the advent of the small, hand-held calculator, some large-scale computer systems with multi-terminal operations included software for simple routine calculations¹. Indeed, such programs were also available for single-terminal minicomputers²⁻⁴. The small, portable calculator virtually replaced such programs by the mid 1970's. Today, the existence of inexpensive multi-terminal mini-computers, with true multi-programming available to users at their desks, and the security problems involved with small hand-held calculators have combined to make calculator simulation programs again attractive. Furthermore, the rapidly increasing capabilities and speed of today's minicomputers, and the decreasing cost of terminals, make it possible for such programs to compete favorably with hand-held calculators, particularly in a multi-programming environment. In addition, specialized "functions" can be included very easily in such software.

The program CALQ8 was written to fulfill a need for day to day arithmetic calculations performed in a solution chemistry laboratory. The environment in which this program was developed, tested and is used, includes a Hewlett-Packard Model 1000 minicomputer with an RTE-4b operating system, 128K-words of central memory, a 15 megabyte floating-head disc drive and six terminals. A terminal is generally located in the laboratory space of a given worker and the program CALQ8 is immediately available to him by typing "RUN,CALQ8".

The chief advantages of CALQ8 over a hand-held calculator are (1) it is implemented on an existing computer system used for on-line experiments and other programs, (2) it is readily available in the laboratory environment, and (3) the power of the minicomputer makes file access and incorporation of other features easy. Advantages over previous programs of this sort include (1) increased speed and availability due to hardware improvement, (2) coding in FORTRAN-IV for ease of modification, (3) very flexible "help" capabilities built into the program, (4) extensive variable assignment and labeling, and (5) a very extensive collection of functions and operations.

This document is designed to be a user's guide as well as maintenance manual. The extensive help features of the program obviate the need for many examples to aid the user in implementing the program.

2. Features of CALQ8

CALQ8 is written in FORTRAN-IV and makes use of some of the Hewlett-Packard extensions^{5,6} to that standard language. It is designed to solve basic computations with the same ease as a pocket calculator. However, the accessibility of such a program on a computer makes computations much more convenient. In addition, CALQ8 makes use of the system's capacity to perform a variety of functions not normally found on pocket calculators.

CALQ8, as written, uses memory with a capacity for 26 one-letter variables. If a variable is undefined and used in an equation, its value will be requested. If a variable has been previously defined, its value will be printed and then entered into the computation. For variables consisting of more than one letter, their values are requested but not stored. If a line consists of a multi-letter variable which begins with a V, the line will be repeated and the variables within the line redefined (except for one-letter variables to the left of the V-variable) until ST is entered. In this way a formula can be used repeatedly.

Multicharacter variables may be one-time variables, but may also be the result of an input error. Any portion of an input line not recognized as a number, symbol, or function is considered a variable. In this way, if there is an input error, the computer will ask "WHAT IS" the portion in error, and only that portion need be clarified, not the entire line. This presents a great advantage over pocket calculators in that many input errors will be identified, thus reducing the possibility of a calculation error. Examples of these and other features are shown in section 8.

CALQ8 allows the user to see previous calculation inputs and outputs on the terminal as well as to use the previous two answers in a calculation as ANS and ANS'.

SUM and AVG functions allow the user to sum or average many numbers or the results of any number of lines of computation. These lines may consist of a single formula used as above or may include variables, functions, and operations. One useful feature of CALQ8 is that no answer to a question or entry need be a single number. In each case a calculation, and, in most cases, a variable can be used. Even a function, such as LOG,a,x, may have entire expressions as arguments 'a' and 'x', or may include variables, as only the commas or parentheses will distinguish the extent of 'a'.

Up to ten parenthetical levels may be used by entering any combination of parentheses, brackets, commas, or the absolute value function, or answering a "WHAT IS" question. The program can easily be modified to include many more levels by changing the DIMENSION statements and the error message. A full range of trig, log and other functions is available and the program can easily be modified to include any function left out.

A "help" feature displays formats and definitions of available functions and operations, as well as other helpful information. More detailed information is also available about variables and error messages by properly responding to the program's query at the end of the initial descriptive information. This information is available at any point during execution by typing "\$L" in response to the CALQ8 prompt.

A repeated formula or "subroutine" capability allows the user to make repeated calculations using the same formula.

These features of CALQ8, combined with the fast calculation and decoding features of compiled code make CALQ8 both easier to use and faster to execute than BASIC or other interpreters. In addition, CALQ8 is likely to be a useful, more capable substitute for hand-held calculators in an environment where a computer system is available.

3. Accuracy

CALQ8 is accurate within the limits of the single precision accuracy on whatever system it is being used. It was developed on an HP1000 system which, as for similar systems, would give CALQ8 an accuracy to six or seven digits, depending on the magnitude of the leading digit in the mantissa, and allow numbers ranging from approximately 10^{-38} to 10^{38} .

4. Limitations

CALQ8 uses two Hewlett-Packard Library routines, RMPAR⁶ and CODE⁷. All Hewlett-Packard 1000 systems running FORTRAN-IV will have these subroutines, and, on many other systems, similar routines may be substituted with slight modifications to CALQ8. Non-HP systems may provide similar routines.

A call to RMPAR is used to retrieve the user-terminal logical unit (LU) number in a multi-terminal environment. The array, IPRAM, receives the parameters passed by the system in the "run" string, and IPRAM(1) is assigned the LU number, used throughout the program for reading and writing. This is standard procedure on all Hewlett-Packard 1000 systems⁷. On systems without this feature, however, the user should replace the DIMENSION IPRAM(5) and the three lines following the last DIMENSION statement with another method of determining LU or delete these lines and replace LU with a number if CALQ8 is only to be used on one input/output device.

CALL CODE is used twice in the program. The first time, after statement number 830, it is used to pack the characters IST to IM of LINE into the array IPACK. IPACK is then ready to be read, free field, as RNUM, using another call to CODE. The call to CODE allows an array to be read from or written into as if it were an input/output device. On systems in which CODE is not present, there may be another similar subroutine which the user may want to implement, or one may be created.

A further limitation of the program on other systems is the "_" symbol at the end of some format statements used throughout in order to suppress the line

feed after a write statement. Most non-Hewlett-Packard systems use a carriage control character at the beginning of the FORMAT statement to accomplish the same function. In most cases, the use of this device is not essential and the user may ignore it, except in writing the answer when this symbol is used several times to join portions of the written statement.

The limitations of the number of parenthetical levels allowed, and the maximum lengths of numbers and of lines may be easily changed by changing the DIMENSION statements to make arrays of the desired size and then altering the error messages accordingly.

None of these limitations should present any problem in adapting CALQ8 to other types of systems Fortran IV.

CALQ8, as written, uses memory with a capacity for 26 one-letter variables. This can easily be doubled by using lower case letters also and changing the number 26 to 52 in the DIMENSION statements and DO LOOP 20. With a few minor alterations, two letter variables can be used, increasing program "memory" to include 676 or 2704 variables.

5. Source Availability

A source of CALQ8 will be supplied on request on user-provided HP cartridge tape.

6. Program Logic Flow Description

CALQ8 is designed to solve basic arithmetical equations in the least amount of computer time and with a minimum of memory space. An eighty character input line is read into an array (LINE) of eighty one-character words which is examined character by character to evaluate a line. Section 9 shows, schematically, the program logic flow described in this section. Section 10 contains a list of variables used in CALQ8.

After initialization, the first ASCII character, LINE(1), is examined in its octal form (as shown in section 9 after entry point I) to determine whether it falls within the range which would designate it to be a numeral. If it is a numeral, the computer increments I (the place in LINE being examined) by one to repeat the process with the next character. IST is equal to I before the loop is entered, and when a value for LINE(I) is found representing a non-numeral, the variable IM is set equal to this value for I, thus IM and IST define the beginning and finishing points for reading the number from LINE.

The process for reading the number, as seen after entry point M, utilizes the Hewlett-Packard routine CODE: first, to write LINE(IST) to LINE(IM) into IPACK, packed in words of two characters each, and then to read IPACK in free-field format to get the number, RNUM. This process is not used if the number is represented by a symbol or variable name. For instance, if there is a '#' symbol (which represents π) it is initially tested to see if it is a KSNFU function until the next operation symbol is found when it is identified as a constant symbol, and RNUM is set equal to the corresponding value, (in this case, 3.141593). At this point, entry point M is bypassed and the program proceeds at N. If, following a search for one-argument functions, the word is not identified as a constant symbol, it is considered a variable. Any input error not recognized by the program also falls into this category.

Detection of a variable causes branching to entry point L after which there is a search for specially treated functions. If a variable name consists of one letter, its corresponding value, when found, is stored in memory. One-letter variables are identified and stored by converting the variable name to a number, LE, from 1 to 26, which is used as a subscript in the array, V. If a one letter variable has been previously defined, its name and value are printed, RNUM is set equal to that value (V(LE)), and the program continues at point N. If the variable has not been used before, its corresponding IV value is equal to 888 (meaning inactive, as opposed to 777 which represents an active value for a flag) and it is treated like a multicharacter variable, as explained below, except that the flag, ISINGL, is activated and the value is stored after point Q. After this, as with all other types of variables, the program resumes at N.

If the variable is being defined in the format "x=...", LS is set equal to LE so that, after the program proceeds from point I to write the answer and return to point B, the value is stored.

In the case of all unrecognized variables not being defined, a question is posed as to what the unknown characters represent. This question only covers the portion of the line in question and therefore is highly specific. This can be very useful in correcting errors as the entire line need not be retyped if the portion in question can be adequately described. LINE and KSNFU are saved as IWHAT and KWHATF so that they may be used in reading the the next line which the user enters after the program picks up at point E. Since J does not equal one, however, the value for this line is not output at entry point R where the answer is listed, but the program branches to Q as if the entire line had been in parentheses. The indicator IALPHA represents the value of J (parenthetical level) at which the question was posed and at which the value ANS is determined when LINE, I, I2, and KSNFU may be restored to their previous values. RNUM is set equal to ANS(J+1) as the program resumes at N.

If a multi-character variable begins with a "V", LOOP is activated so that after point C the logic flow continues at F, bypassing the READ LINE, so that the same line is used continuously as the same questions are repeated and the values computed. The loop can be broken by entering ST which causes branching to A.

Parenthetical levels of up to ten may be reached. J is incremented by entering a "(" or a first "1". This is recognized after point I, and J is incremented by 1, returning the flow to point I. A closing ")" or a second "1" or "," is treated like an end of line, recognized at point K, causing KADSB(J) to be set equal to 98 so that, after entry at statement 980, execution proceeds at statement 990 at which, if there is a parenthetical level higher than one, the program resumes at Q instead of R. There the value for ANS(J) is used to determine RNUM, and the parenthetical level is lowered by one. Of the three possible paths that may be taken from entry point Q, the one taken in this instance is that of activating ISKIP (ISKIP 777) before resuming at point I so that the number reading process is bypassed and the already computed value for

RNUM is retained. Another possible path from Q is that used in LOG,a,x functions. If LOGFU(J) is activated by one of these functions, the two commas are interpreted as parentheses so that any number, variable, or calculation between them yield a value for ANS(J+I). This value is used for BASE as the program resumes at point H to read RNUM and convert it to the log a, or $\log a^{-1}$ function of RNUM (depending on the value of KSNFU).

The value for a one-argument function of RNUM is assigned to RNUM after N if KSNFU(J) is activated after point K. If a symbol is not recognized as an operation, it is tested to see if it is a one-argument function such as SIN, LOG, SQRT, ACOT, etc. If it is one, KSNFU(J) is assigned a function number so that, after the flow is sent to H, and RNUM is acquired after point N, RNUM is altered according to the value of KSNFU(J). In the case of trig functions there is a choice of whether to use radians or degrees. If not previously specified, the preferred format, to be used for both input and output is requested. This format may be changed at a later time by entering DEG or RAD which is detected after point L, sets IDR accordingly, and sends the logic flow to point D again. Once IDR is defined, it may either be inactive, in which case the normal radian-using functions are unaltered, or it may be active in which case any RNUM about to be redefined as a trig function of RNUM is converted from degrees to radians and any output RNUM after a trig function is converted from radians to degrees.

For all KSNFU(J) functions, the function symbol is found before the number so that the KSNFU(J) value must be saved until the first RNUM at the same parenthetical level is found. The first RNUM is equal to the first final ANS(J+I) encountered. In the case of a factorial, however, the function symbol occurs after the number, and therefore, a different method of calculation is used. After the symbol (!) is detected, the flow proceeds to M and then, after N, RNUM! is calculated.

Numbers and functions must be separated by one of the operations +, -, /, *, **, and =. These symbols are searched for first, at K, from which, if no non-

numeral characters between operation symbols are found, the flow continues at M where the number between symbols is read. After point O, the functions are computed by heirachy, giving preference to exponentation over multiplication and division which in turn are given preference over addition and subtraction. This is accomplished by having separate flags and variables for each level. At each level, there are at least two flags. There is a primary flag which is activated when the symbol is detected. This is used to set a secondary flag so that the next time the flow passes through O (when I is at the next operation symbol and the preceeding number has been read) a computation is performed. The order in the logic flow as illustrated then determines the heirachy. In the case of multiplication and division, represented by MLDIV, three variables are used: a primary 3-state flag, MLDIV(J); a secondary 3-state flag, MLDIV2(J); and a third variable, RNO(J), to hold the previous value of RNUM. The exponentation process also utilizes a holder, EXP(J), which serves the additional function of a secondary flag. Addition, subtraction, and equals functions make use of the 5-state flags, KADSB(J) and KADSB2(J), which use ANS(J) as the value holder, as this is the last category in the heirachy. If the operator is other than "=", the flow is sent to point H.

If there is an equal sign or end-of-the-line function, the flow may be treated in various ways. End-of-the-line functions are sent to point Q if there is a parenthetical level higher than one. If J=1, and there is no summation taking place, execution is sent to R where the answer is printed; or if ISU (summation indicator) is active, SUM is incremented by ANS(J) and N is incremented by one as the flow is returned to point D where the "+" prompt appears. If there is an "=" function and there is a summation, the summation ends and the answer is written at point R. If the "=" symbol is not the first or second character, the symbol is interpreted as a continuation symbol and another input line is expected without an intervening end of line function so that a long computation may be continued.

At point R where an answer is printed, the various formats for different types of numbers are used. First, it is determined if the output is a SUM,

AVERAGE, or N, in which cases it is specified. In all cases except N, numbers greater than 10^{37} in magnitude are written as + or - INFINITY, integers are printed in I8 format, numbers greater than 10^7 in E9.7 format, and all other real numbers in F9.7 format. If IAV is active, the sum is written first and then there is a return to 1110 to write the average. This last answer is saved as AS, and any previous value for AS is saved as AS2, both of which are used for giving values of ANS and ANS' respectively as constant symbols. From here, the flow returns to point B for initialization and to read in a new line.

7. ACKNOWLEDGMENT

The authors wish to acknowledge the administrative contributions of the Gifted and Talented Internship Program of Fairfax County Virginia which made it possible for one of us (Daniel F. Birn) to participate in the ongoing research of the Naval Research Laboratory and which made development of this computer program possible.

8. REFERENCES

1. For example, CALCTRAN, a calculation-translation program for the IBM-360 Computer, a limited distribution program, available on many IBM computer systems in the late 1960's.
2. A. K. Head, "HEP(PAL) PDP-8," Hep. DCS DECUS8-155, ACM Computer Programming Dir. 106, 1971.
3. J. D. Larson, "FOCAL FORTRAN-CALCULATOR," XDS 900-03.9.002, ACM Comp. Prog. Dir., 252, 1971.
4. a. Hewlett-Packard LOCUS library program HP 22084B Revision 16.10.71, "Integrated Math Calculator Program."
b. "Estructura de un Simulador de Calculadora Analogica (CAN), Revista de Informatica y Automatica 11 37, 1978.
5. RTE FORTRAN-IV Reference Manual, Hewlett-Packard Co., 1979, Appendix D.
6. DOS/RTE Relocatable Library Reference Manual, Hewlett-Packard Co., 1978, 4-29.
7. DOS/RTE Relocatable Library Reference Manual, Hewlett-Packard Co., 1978, 3-23.

9. SAMPLE PROGRAM DIALOG

```

END,CALC8
ENTER CALCULATION
(TYPE 'SL' FOR LIST OF FUNCTIONS, 'Q' TO END)
33+4*(5+6)^2
=497.0002
35*6*((SQRT(4)+8)/2)
=279.3868
3A*B*(C@)/2:
WHAT IS 'A' ?
15
WHAT IS 'B' ?
25
WHAT IS 'C' ?
7SQRT4
=279.3868
3D=ANS
=279.3868
3(VAR1+VAR2)/2
WHAT IS 'VAR1' ?
13
WHAT IS 'VAR2' ?
15
=
4
WHAT IS 'VAR1' ?
14664
WHAT IS 'VAR2' ?
132532
=
13648
WHAT IS 'VAR1' ?
1ST
35+6*75300PS18
WHAT IS '75300PS18' ?
17+8
=
YE
3ASIN6!
IS FORMAT FOR TRIG FUNCTIONS TO BE IN DEGREES OR IN RADIANS?
(PLEASE NOTE THAT FORMAT CAN BE CHANGED AT
ANY TIME BY ENTERING 'DEG' OR 'RAD') D
=+INFINITY
3LOG(5+6,(6!))
=2.743751
3RAD
3ASIN6!
=
3SIN(4/2)
=
1
3SIN4/2
=.7559E-07
3SIN#
=.00000032
3DEG
3SIN90
=
1
3SIN180/2
=.7559E-07

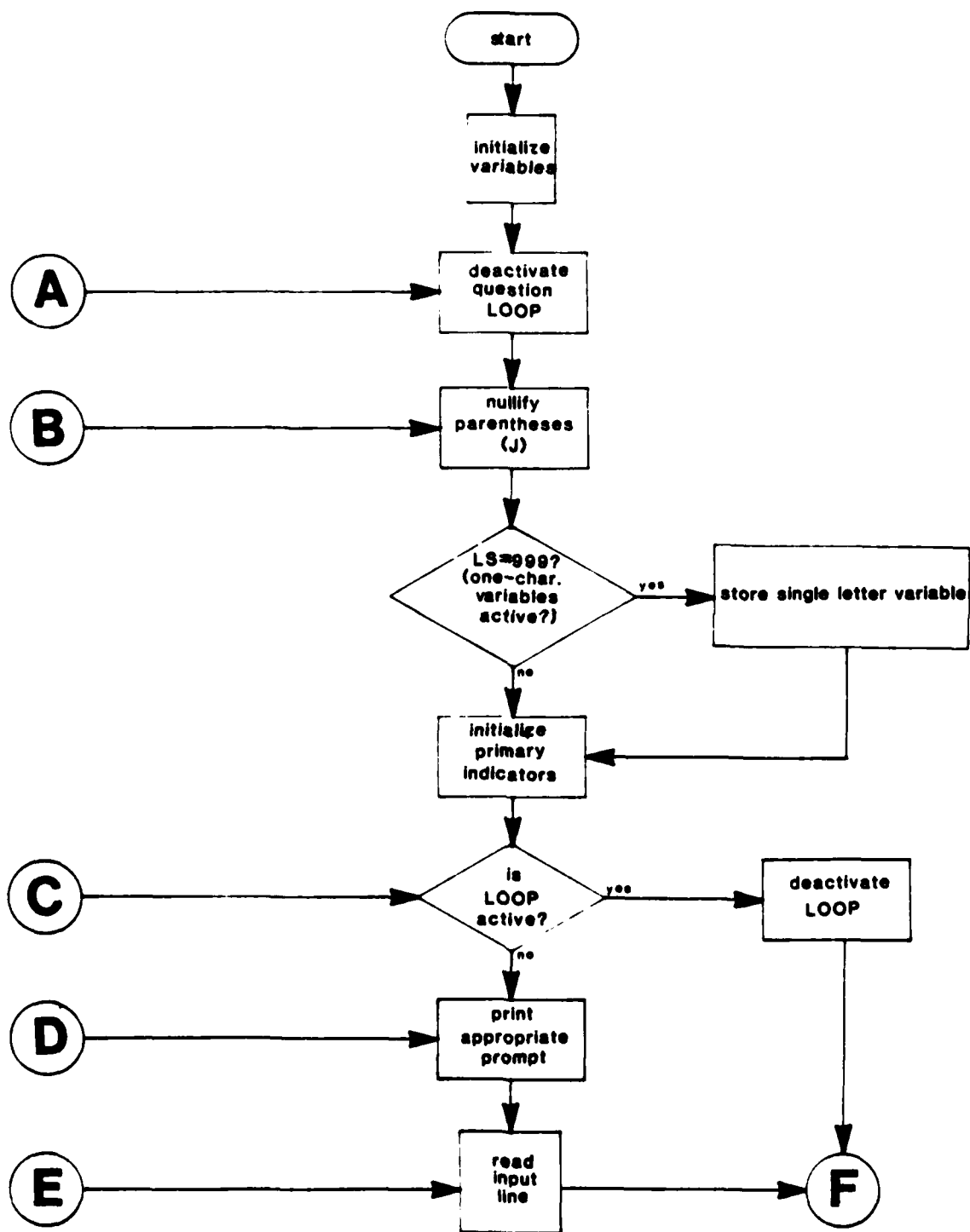
```

```

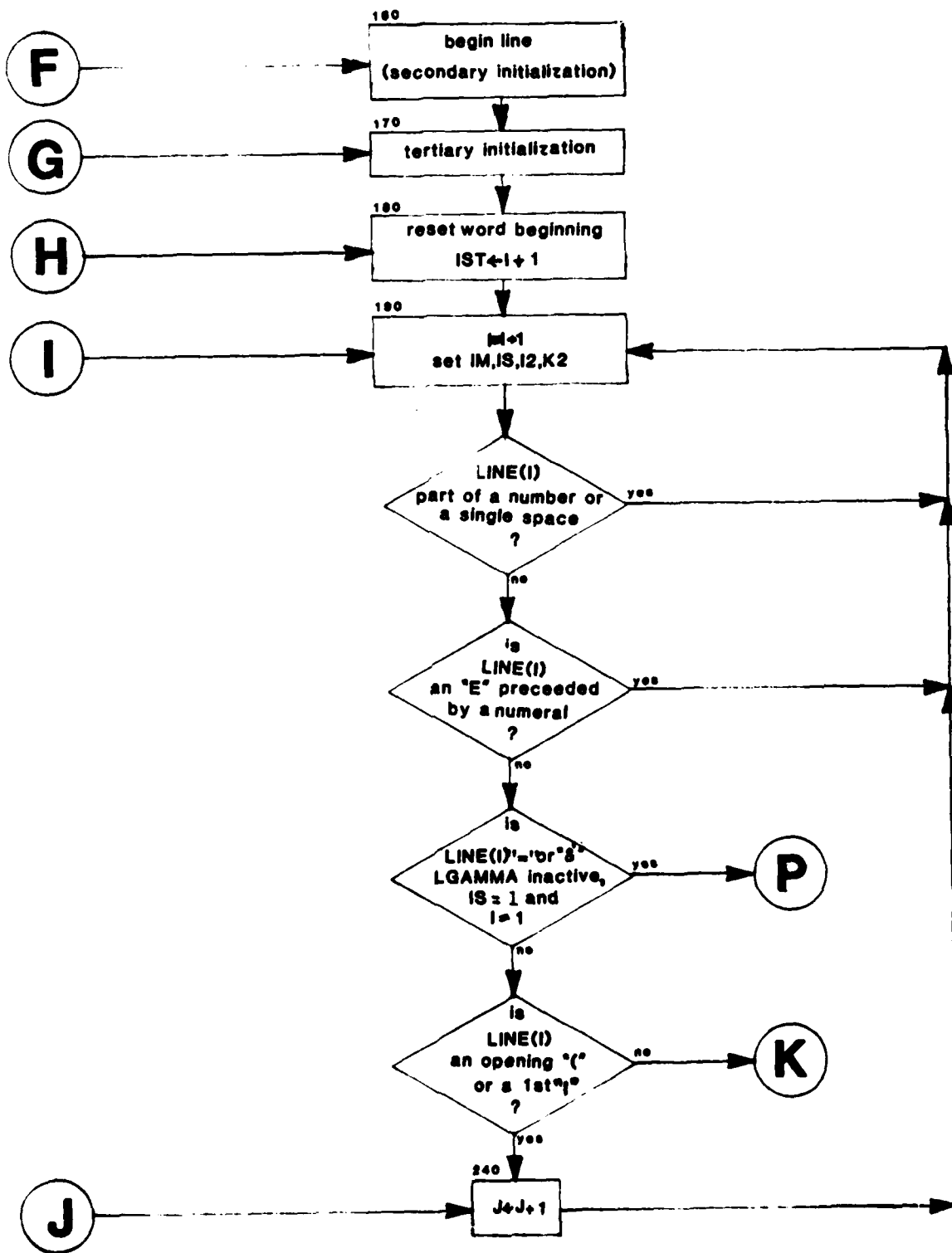
)AUG
+5*8
+ANC
A= 5.10
C= 1.77
+D
D= 279.37
+=
SUM =318.2491
N= 3
AVERAGE =106.0830
)E=ANS
=318.2491
)F=ANS
=106.0830
)G=ANS/2
=53.04151
)9/0
=+INFINITY
)7/2
=.3507E+38
)INVANS
=.1175E-37
)+8E-38
=.8812E-35
)GOOD BYE
WHAT IS GOOD BYE?
)A GREETING
WHAT??
)COMPUTERS, YOU MUST REMEMBER ARE STILL NOT QUITE AS SMART AS HUMANS.
WHAT??
(TYPE "SL" FOR LIST OF FUNCTIONS, "QT" TO END)
)QT

```

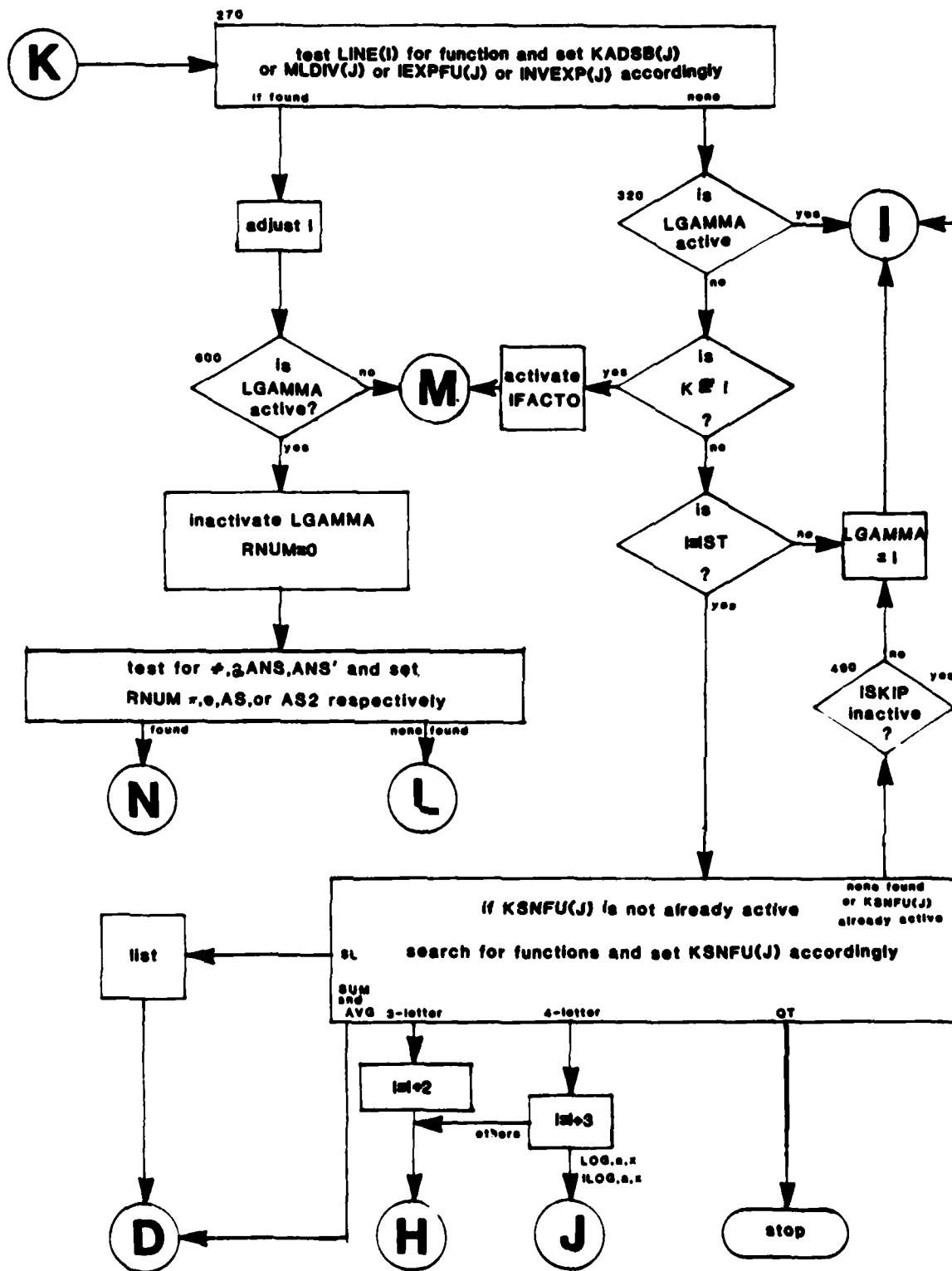
10. FLOWCHART - INITIALIZATION AND READING OF INPUT LINE



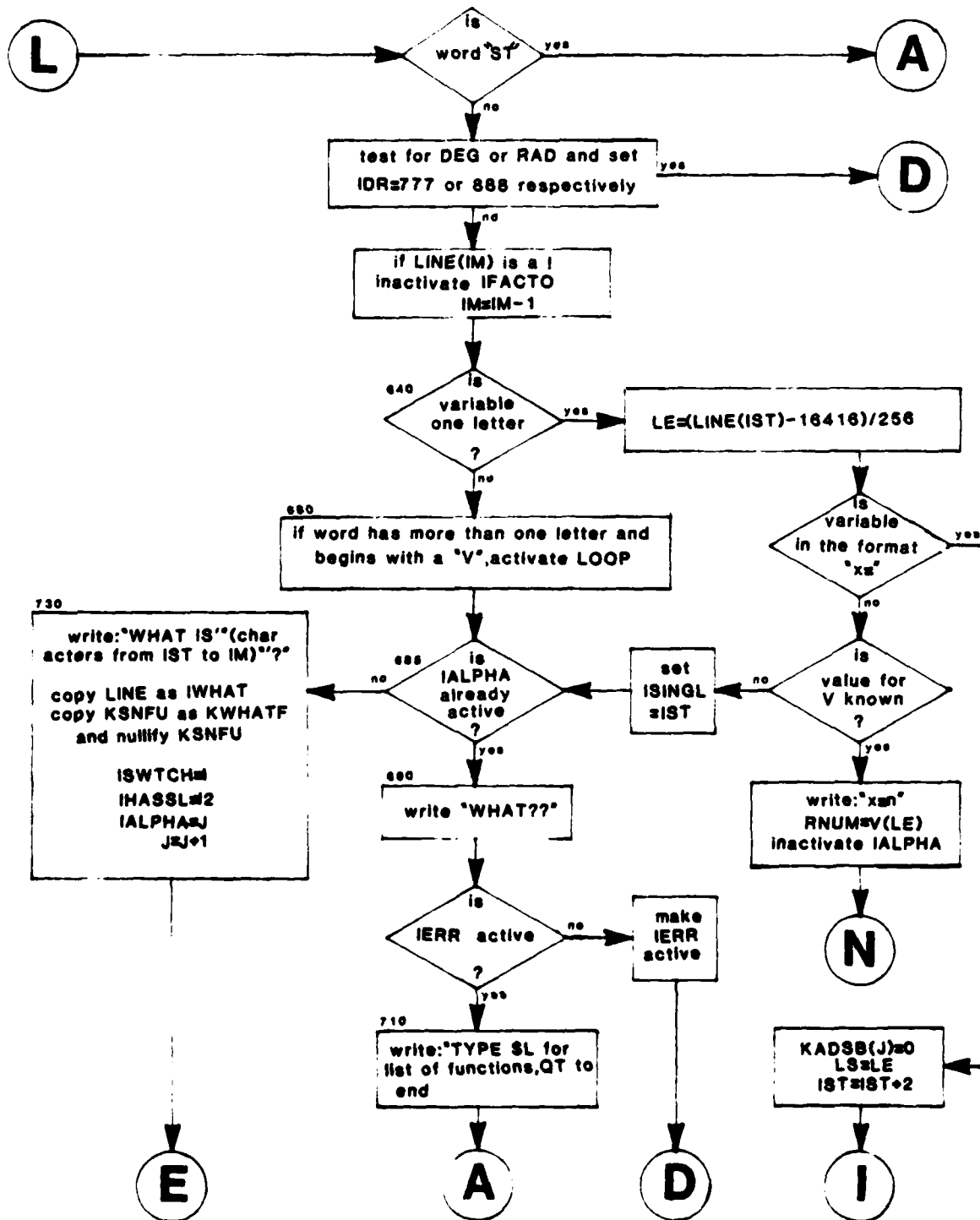
FLOWCHART (CONTINUED) - SEARCH FOR NON-NUMERAL CHARACTERS



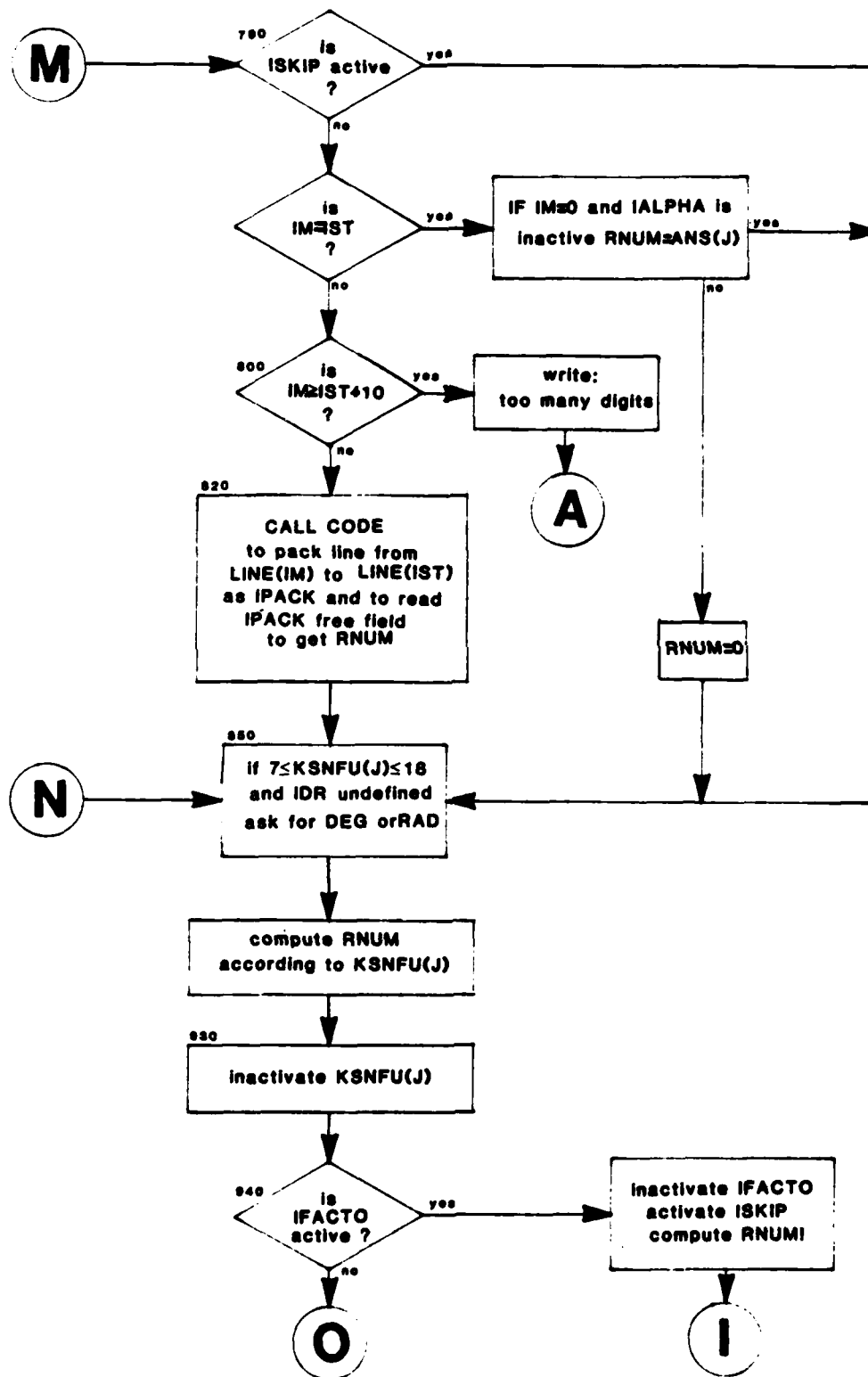
FLOWCHART (CONTINUED) - SEARCH FOR FUNCTIONS, CONSTANTS AND VARIABLES



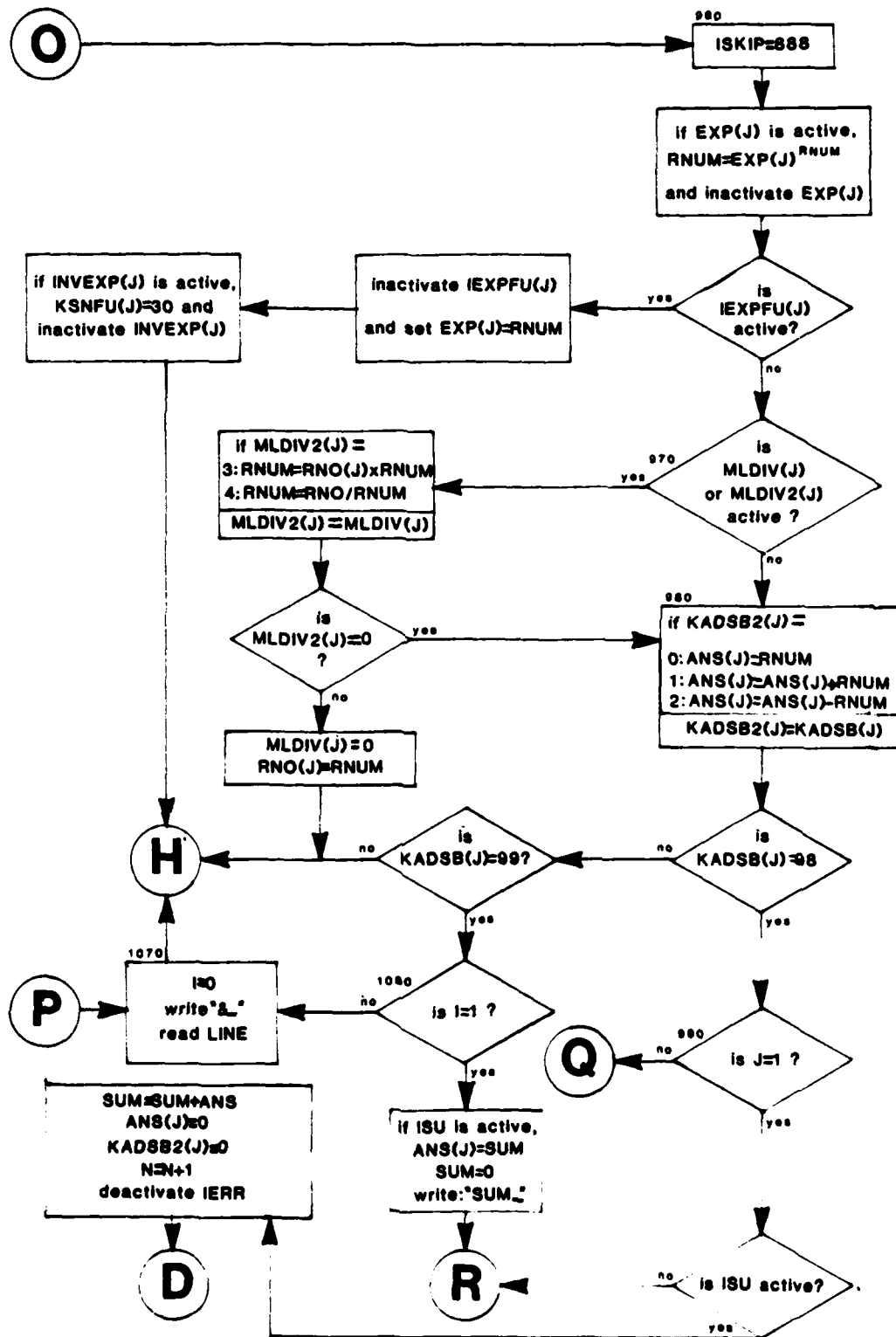
FLOWCHART (CONTINUED) - VARIABLES



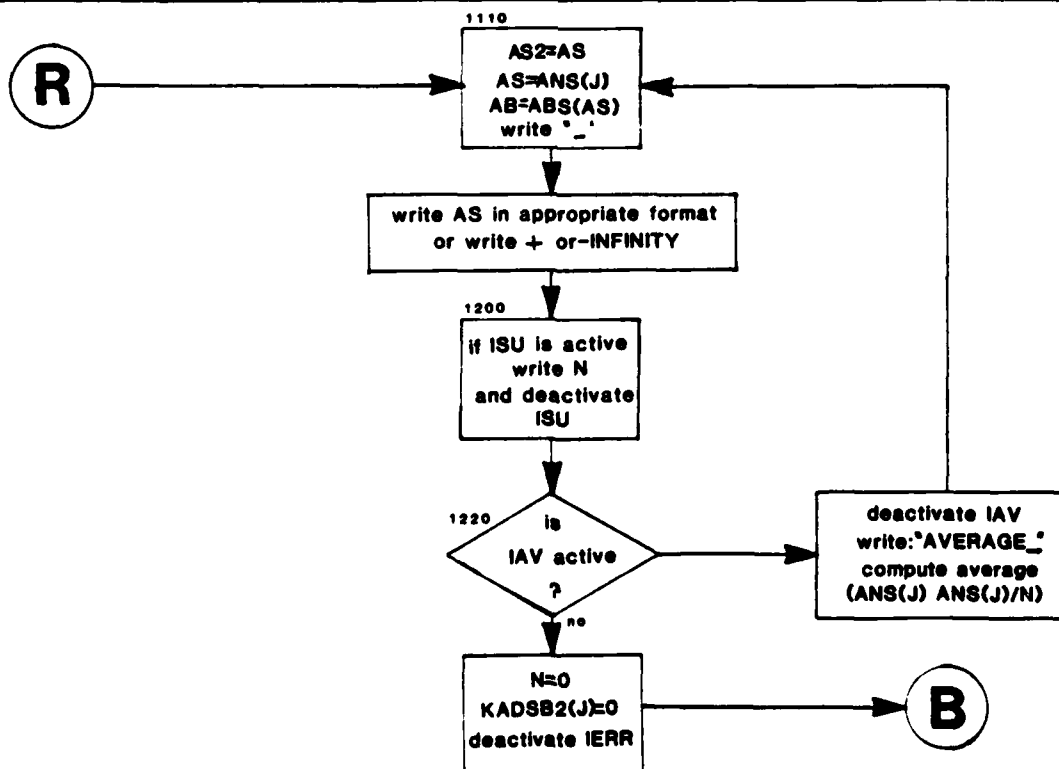
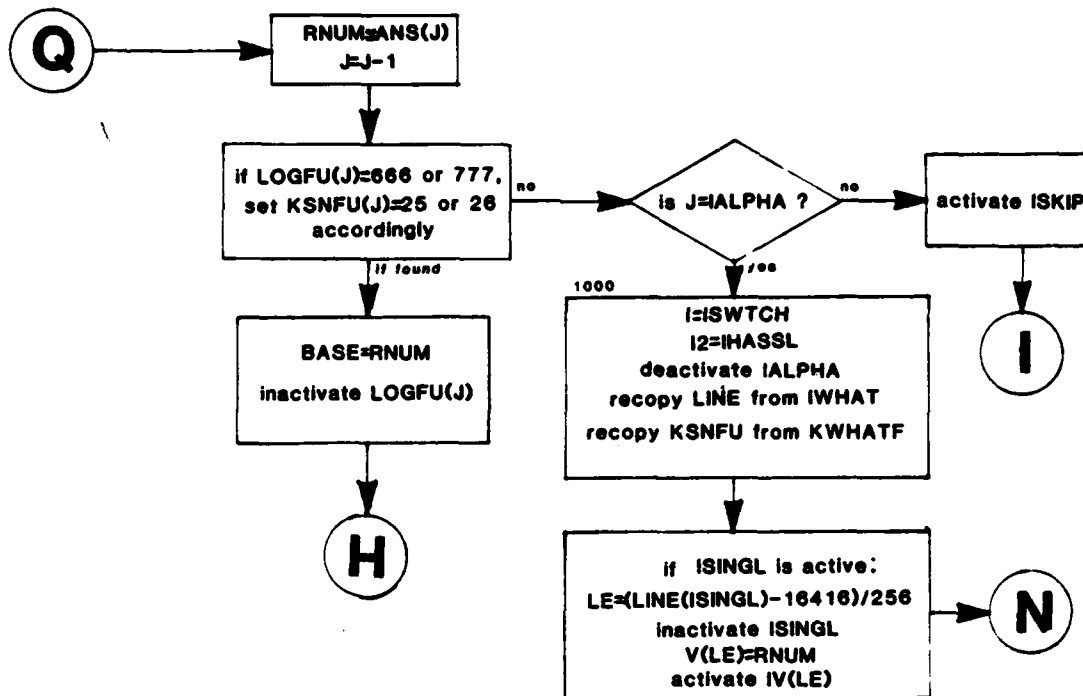
FLOWCHART (CONTINUED) - READING AND COMPUTING RNUM



FLOWCHART (CONTINUED) - HIERARCHY COMPUTATION



FLOWCHART (CONTINUED) - PARENTHESES REDUCTION, WRITING OF ANSWER



11. CALQ8 Variables

AB	=	ABS(ANS(J))
ANS(10)	=	computed answer after going through KADSB2 computation after statement no. 980. (J) is parentheses level.
AS	=	abbreviation for ANS(J) and conserves value for AS2 after average computation.
AS2	=	previous AS used for ANS' function.
BASE	=	log base a in LOG,a,n and ILOG,a,n functions.
EXP(10)	=	number which is an exponent or -9999 indicating inactivity.
F	=	number multiplied by (L:L=1,RNUM) to get factorial
I	=	subscript for LINE(1).
I2	=	I+1 except when 190 loop has been passed through several times in succession in which case any "-" will not be part of RNUM number.
IALPHA	=	value for J at which IWHAT was filled and at which LINE will be computed as RNUM and replaced by IWHAT.
IAV	=	indicator of whether average of sum is to be taken.
IDR	=	indicator of whether trig functions are in degrees or radians.
IERR	=	indicator of whether "What??" question has already been asked.
IEXPFU(10)	=	function indicator for exponentiation.
IFACTO	=	function indicator for factorilization
IHASSL	=	memory of where I2 was when LINE stored in IWHAT.
IM	=	place in LINE where number or word ends.
INVEXP(10)	=	function indicator for deexponentation.
IPACK(5)	=	array into which LINE(IST,IM) is packed in 5A2 format for reading RNUM.
IPRAM(5)	=	array used in CALL RMPAR to find LU number.
IS	=	preserved previous value of I2 to see if first character in number.

ISINGL	= indicator of where single letter variable is or if non-single.
ISKIP	= indicator to skip number computation as value has been found.
IST	= indication of where word or number begins.
ISU	= function indicator for summation.
ISWTCH	= memory of where I was when LINE stored in IWHAT.
IV(26)	= indicator of which single letter variables are undefined.
IWHAT(80)	= storage of LINE when "WHAT IS" question is being asked.
J	= subscript indicating current parentheses level.
K	= LINE(I) (abbreviated name)
K2	= LINE(I+1)
K3	= LINE(I+2)
K4	= LINE(I+3)
KADSB(10)	= function indicator for addition, subtraction, =, and end of line.
KADSB2(10)	= secondary indicator for KADSB(J) to execute function next round.
KSNFU(10)	= function indicator for all trig and other one-argument functions.
KWHATF(10)	= copy of KSNFU when "WHAT IS" question is being asked.
L	= miscellaneous DO LOOP subscript, and question response indicator.
LE	= numerical representative from 1 to 26 of single letter variable
LGAMMA	= marker of whether and where variable exists.
LINE(80)	= array in which line to be computed is stored.
LOGFU(10)	= function indicator for 3 different log functions.
LOOP	= indicator of whether repetitive use of LINE is in effect.
LS	= indicator of whether and which place in V(26) AS is to be defined.
LU	= terminal number as computed in CALL RMPAR.

MLDIV(10)	= function indicator for multiplication and division.
MLDIV2(10)	= secondary indicator to execute MLDIV function, specified previous round.
N	= indicator of how many entries have been summed (also used to compute average).
RNO(10)	= previous RNUM in MLDIV computations.
RNUM	= number presently being worked with to get ANS(J).
SUM	= sum of AS values for lines which were computed under summation function.
V(26)	= memory for single letter variables.
X	= abbreviation for $\text{SQRT}(1-\text{RNUM}^2)$ used in computing arc sin and arc cosine functions.

12. PROGRAM LISTING

PAGE 0001 FTN. 11:47 AM THU. 29 AUG., 1980

```

0001 FTN4,L
0002 PROGRAM CALQ6(3,99),REV 3 - 25 - 80 D BIRN
0003
0004
0005 DANIEL BIRN
0006 OCTOBER 5, 1977
0007 16ST RECENT UPDATE: THURSDAY, AUGUST 25, 1980
0008
0009
0010
0011 DIMENSION IPRAM(5),LINE(80),IWHAT(80),IPACK(5)
0012 DIMENSION HLDIV(10),HLDIV2(10),IEXPFU(10),EXP(10),RNG(10)
0013 DIMENSION <ADSB(10),<ADSB2(10),ANS(10),KSNFU(10),IWHATF(10)
0014 DIMENSION V(26),IV(26),LOGFU(10),INVEXP(10)
0015 CALL RMPAR(IPRAM)
0016 LU=IPRAM(1)
0017 IF(LU.LE.3)LU=1
0018 10 WRITE(LU,10)
0019 FORMAT(' ENTER CALCULATION')
0020 WRITE(LU,720)
0021 IERR=888
0022 N=0
0023 ISU=888
0024 SUM=0
0025 LS=999
0026 IDR=999
0027 ISINGL=777
0028 NEGVAR=888
0029 AS=0
0030 AS2=0
0031 DO20 L=1,25
0032 V(L)=0
0033 IV(L)=888
0034 GO TO 50
0035 30 WRITE(LU,40)
0036 40 FORMAT(' LINE TOO LONG. -- 77 CHARACTERS IS MAXIMUM')
0037 50 LOOP=888
0038 J=1
0039 IF(LS.EQ.999)GO TO 70
0040 V(LS)=AS
0041 IV(LS)=777
0042 LS=999
0043 LGAMMA=999
0044 IALPHA=999
0045 IFACTO=888
0046 ISTOP=0
0047 DO30 L=1,10
0048 LOGFU(L)=888
0049 IF(LOOP.EQ.777)GO TO 150
0050 IF(ISU.NE.777)GO TO 110
0051 WRITE(LU,100)
0052 FORMAT(' + ')
0053 GO TO 130
0054 110 IAV=888
0055 SUM=0
0056 LOOP=888

```

PAGE 0002 CALGB 11:47 AM THU., 29 AUG., 1980

```

0056      WRITE(LU,120)
0057 120  FORMAT(" ")
0058 130  READ(LU,140)LINE
0059 140  FORMAT(80A1)
0060      IF(LINE(80).NE.200408)GO TO 30
0061      GO TO 150
0062 150  LOOP=888
0063 160  I=0
0064      I2=0
0065      ISKIP=888
0066 170  MLDIV(J)=1
0067      MLDIV2(J)=0
0068      EXPFU(J)=888
0069      EXP(J)=-9999
0070      KADSB2(J)=0
0071      INVEXP(J)=888
0072      LGAMMA=999
0073 180  IST=I+1
0074 190  I=I+1
0075      K=LINE(I)
0076      IF(I.GT.79)GO TO 30
0077      IF((K.GE.300408.AND.K.LE.344408).OR.K.EQ.
0078 +270408.OR.(K.EQ.264408.AND.I.EQ.I2.AND.ISKIP.NE.777
0079 +.AND.LGAMMA.EQ.999))GO TO 190
0080      IM=I-1
0081      IS=I2
0082      I2=I+1
0083      K2=LINE(I2)
0084      IF(K.NE.424408.OR.I.EQ.IS.OR.I.EQ.1)GO TO 200
0085      IF(K2.EQ.254408)I=I+1
0086      GO TO 190
0087 200  KADSB(J)=1
0088      IF(K.EQ.244408.OR.K.EQ.564408.OR.K.EQ.764408)GO TO 210
0089      IF(K.EQ.260408.AND.J.GT.1.AND.LOGFU(J-1).NE.888)GO TO 210
0090      IF(K.NE.200408.AND.I.LT.79)GO TO 220
0091      IF(K2.NE.200408)GO TO 190
0092      IF(I.NE.1)GO TO 210
0093      ISTOP=ISTOP+1
0094      IF(ISTOP.GE.3)GO TO 1249
0095      IF(ISTOP.EQ.1)WRITE(LU,209)
0096 209  FORMAT(" WAITING FOR INPUT")
0097      IF(ISTOP.EQ.2)WRITE(LU,208)
0098 208  FORMAT(" GOING!")
0099      GO TO 90
0100 210  KADSB(J)=98
0101      GO TO 600
0102 220  IF(K.NE.760408)GO TO 230
0103      IF(KSNFU(J-1).EQ.20)GO TO 210
0104      KSNFU(J)=20
0105      GO TO 240
0106 230  IF(K.NE.240408.AND.K.NE.554408.AND.K.NE.754408)GO TO 260
0107 240  J=J+1
0108      IF(J.LE.10)GO TO 170
0109      WRITE(LU,250)
0110 250  FORMAT(" TOO MANY '(' -- NINE IS MAXIMUM")

```

PAGE 0003 CALQB 11:47 AM THU., 28 AUG., 1950

```

0111      GO TO 50
0112 C    ** ** ** =(EQUALS) ** ** ** ** ** ** ** ** ** ** ** ** ** 
0113 260  IF(K.NE.364408.AND.K.NE.230408)GO TO 270
0114      IF(I.S.EQ.I.AND.LGAMMA.EQ.999.AND.I.NE.1)GO TO 1070
0115      KADSB(J)=99
0116      GO TO 600
0117 C    ** ** ** ** * (PLUS) ** ** ** * ** ** * 
0118 270  IF(K.EQ.254408)KADSB(J)=1
0119      IF(K.EQ.264408)KADSB(J)=2
0120      IF(KADSB(J).NE.0)GO TO 600
0121      IF(K.EQ.570408)GO TO 290
0122      IF(K.NE.370408)GO TO 280
0123      INVEXP(J)=777
0124      GO TO 290
0125 C    ** DISTINGUISHES BETWEEN "*" (TIMES)AND "**" (EXPONENT) **
0126 280  IF(K.NE.250408)GO TO 310
0127      IF(K2.NE.250408)GO TO 300
0128      I=I2
0129 290  IEXPFU(J)=777
0130      GO TO 600
0131 300  MLDIV(J)=3
0132      GO TO 600
0133 C    ** LOOKS FOR "/" (DIVIDE)**
0134 310  IF(K.NE.274408)GO TO 320
0135      MLDIV(J)=4
0136      GO TO 600
0137 320  IF(LGAMMA.NE.999)GO TO 190
0138 330  IF(LINE(IST).EQ.200408)IST=IST+1
0139      IF(K.NE.204408)GO TO 340
0140      IFACTO=777
0141      GO TO 290
0142 340  IF(I.NE.IST)GO TO 495
0143      K3=LINE(I+2)
0144      IF(KSNFU(J).NE.0)GO TO 490
0145      IF(K.EQ.414408.AND.K2.EQ.474403.AND.
0146      +K3.EQ.514408)KSNFU(J)=8
0147      IF(K.EQ.444408.AND.K2.EQ.470408.AND.
0148      +K3.EQ.530408)KSNFU(J)=30
0149      IF(K3.NE.470408)GO TO 350
0150      IF(K.EQ.520408.AND.K2.EQ.404408)KSNFU(J)=9
0151      IF(K.EQ.514408.AND.K2.EQ.444408)KSNFU(J)=7
0152 350  IF(K.EQ.414408.AND.K2.EQ.514408.AND.K3.EQ.K)KSNFU(J)=13
0153      IF(K.EQ.514408.AND.K2.EQ.424408.AND.K3.EQ.414408)KSNFU(J)=14
0154      IF(K.EQ.414408.AND.K2.EQ.474408.AND.K3.EQ.520408)KSNFU(J)=15
0155      IF(KSNFU(J).NE.0)GO TO 438
0156      K4=LINE(I+3)
0157      IF(K.NE.514408)GO TO 380
0158      IF(K2.NE.504408.OR.K3.NE.
0159      +510408.OR.K4.NE.520408)GO TO 370
0160      KSNFU(J)=6
0161      GO TO 437
0162 370  IF(K2.NE.524408.OR.K3.NE.464408.OR.I.NE.1)GO TO 380
0163      ISU=777
0164      GO TO 70
0165 380  IF(K.NE.404408.OR.K2.NE.530408.OR.K3.NE.434408)GO TO 390

```

PAGE 0004 CPLQ8 11:47 AM THU., 28 AUG., 1980

```

0166      ISU=777
0167      IAV=777
0168      GO TO 90
0169 390   IF(K.NE.460408)GO TO 420
0170      IF(K2.NE.474408.OR.K3.NE.434408)GO TO 411
0171      IF(K4.NE.260408)GO TO 400
0172      I=I+3
0173      LOGFU(J)=777
0174      GO TO 240
0175 400   KSNFU(J)=22
0176      GO TO 438
0177 410   IF(K2.NE.470408)GO TO 420
0178      KSNFU(J)=21
0179      GO TO 439
0180 420   IF(K.NE.444408.OR.K2.NE.460408)GO TO 450
0181      IF(K3.NE.474408.OR.K4.NE.434408)GO TO 440
0182      IF(LINE(I+4).NE.260408)GO TO 430
0183      I=I+4
0184      LOGFU(J)=666
0185      GO TO 240
0186 430   KSNFU(J)=24
0187 437   I=I+1
0188 438   I=I+1
0187 439   I=I+1
0190      GO TO 180
0191 440   IF(K3.NE.470408)GO TO 450
0192      KSNFU(J)=23
0193      GO TO 438
0194 450   IF(K.NE.404408)GO TO 470
0195      IF(K2.NE.414408)GO TO 460
0196      IF(K3.EQ.474408.AND.K4.EQ.514408)KSNFU(J)=11
0197      IF(K3.EQ.514408.AND.K4.EQ.K2)KSNFU(J)=16
0198      IF(K3.EQ.474408.AND.K4.EQ.520408)KSNFU(J)=13
0199 460   IF(K2.EQ.514408.AND.K3.EQ.424408.AND.K4.EQ.414408)KSNFU(J)=17
0200      IF(K4.NE.470408)GO TO 470
0201      IF(K2.EQ.514408.AND.K3.EQ.444408)KSNFU(J)=10
0202      IF(K2.EQ.520408.AND.K3.EQ.404408)KSNFU(J)=12
0203 470   IF(KSNFU(J).NE.0)GO TO 437
0204 490   IF(ISKIP.EQ.777)GO TO 190
0205      IF(I.NE.1)GO TO 495
0206      IF(K.EQ.220408.AND.K2.EQ.
0207      504408.AND.K3.EQ.520408)GO TO 1250
0208      IF(K.EQ.220408.AND.K2.EQ.460408)GO TO 500
0209 495   LGAMMA=I
0210      GO TO 191
0211 500   WRITE(LU,510)
0212 510   FORMAT(" Do you want this listed on the printer or screen?_")
0213      READ(LU,520)L
0214 520   FORMAT(A1)
0215      IF(L.EQ.514408)GO TO 550
0216      IF(L.NE.500408)GO TO 500
0217      WRITE(6,560)
0218      WRITE(6,530)
0219      WRITE(6,590)
0220      WRITE(6,540)

```

```

0221 540 FORMAT('1')
0222 GO TO 90
0223 550 WRITE(LU,560)
0224 560 FORMAT(' CALQ8 FUNCTIONS ARE AS FOLLOWS:/'
0225 ' " + " ..... PLUS"/
0226 ' " - " ..... MINUS"/
0227 ' " * " ..... TIMES"/
0228 ' " / " ..... DIVIDED BY"/
0229 ' " ** " OR " ^ " ..... EXPONENT"/
0230 ' " SQRT " ..... SQUARE ROOT"/
0231 ' " INV " ..... INVERSE"/
0232 ' " ( ) " ..... INVERSE POWER (i.e. x^(1/y))"/
0233 ' " SIN " X, " COS " X, " TAN " X"/
0234 ' " CSC " X, " SEC " X, " COT " X TRIG FUNCTIONS OF X"/
0235 ' " ADD AN " A " BEFORE THESE FOR THE ARC FUNCTIONS"/
0236 ' " SUM " ..... WILL SUM SINGLE NUMBERS OR"/
0237 ' " FULL COMPUTATIONS LINE BY LINE. USE "=" KEY FOR ANSWER"/
0238 ' " AVG " ..... GIVES SUM WITH AVERAGE"/
0239 ' " PI " ..... THE NUMBERS PI AND E RESPECTIVELY"/
0240 ' " ! " ..... FACTORIAL"/
0241 ' " ABS " ..... ABSOLUTE VALUE OF " X " OR AN EXPRESSION"/
0242 ' " ( ) " ..... BRACKETS"/
0243 ' " LOG " X, " LN " X, " LOG " a, X: LOG10, LOGe, AND LOGa OF X"/
0244 ' " ILOG " X, " ILN " X, " ILOG " a, X: INVERSE OF ABOVE FUNCTIONS"/
0245 ' " = " or " & " ..... END OF SUMMATION OR LINE CONTINUED"/
0246 ' " Q " ..... END PROGRAM"/
0247 ' " Do you want to find out about variables and errors? " )
0248 READ(LU,570)
0249 570 FORMAT(A1)
0250 IF(L.EQ.470408)GO TO 90
0251 WRITE(LU,580)
0252 580 FORMAT(' Variables are used as follows:/'
0253 ' " Any sequence of characters which are not recognized as a "/
0254 ' " function or number are made variables. This includes "/
0255 ' " improperly placed functions and numbers with an accidental "/
0256 ' " letter beside or within. In the case of an error, the "/
0257 ' " program will ask "WHAT IS" (the sequence) after which the "/
0258 ' " user should restate properly the entire quoted portion of "/
0259 ' " the line. In the case of an error entered on this second "/
0260 ' " attempt (the prompter this time being a "?"), a simple "/
0261 ' " "WHAT?" is asked to which the user still has one last "/
0262 ' " chance to enter an unerrored line after the "?". If "/
0263 ' " another "WHAT?" comes back again (along with a brief "/
0264 ' " instructory message), the user must begin the line "/
0265 ' " all over again."/
0266 ' " However, this feature of CALQ8 is useful for another "/
0267 ' " purpose -- that of being a variable in an equation. If a "/
0268 ' " variable of one letter from A to Z is used, the value will "/
0269 ' " be stored and can be used in any further equation. A "/
0270 ' " variable of this kind may be defined or changed by entering "/
0271 ' " "X=" for instance. If one such variable is used in an "/
0272 ' " equation but has not been previously defined, the program "/
0273 ' " will ask for and assign its value."/
0274 ' " If a variable of more than one character beginning with "/
0275 WRITE(LU,585)

```

```

0276 585 FORMAT(22X,"PRESS RETURN WHEN FINISHED READING_")
0277 X=0
0278 READ(LU,*)X
0279 CONTINUE
0280 WRITE(LU,570)
0281 590 FORMAT(" a 'V' is included in the line, the program will "
0282 + "continue"/
0283 + " to ask for values for the variables until 'ST' is pressed"/
0284 + " and will continue to give answers for each cycle."/
0285 + " One-letter variables in this format will only be treated"/
0286 + " as such if they are entered before the initial multi-"/
0287 + " character variable beginning with V. Like for SUM and"/
0288 + " any other portion where values are entered(including"/
0289 + " the base for LOG(a), the entered values for variables"/
0290 + " need not be direct numbers but may consist of a whole line"/
0291 + " of computation."/
0292 + " in such cases as well as in others it may be useful to"/
0293 + " use ANS as a variable which has the value of the previous"/
0294 + " line. A line is stored by entering 'X=ANS', for "/
0295 + " example. Also, ANS' is used to represent the value"/
0296 + " of the line before that represented by ANS.")
0297 GO TO 90
0298 600 IF(LINE(IST).EQ.20040B)IST=IST+1
0299 IF(LGAMMA.EQ.999)GO TO 770
0300 LGAMMA=999
0301 RNUM=0
0302 IF(LINE(IST).EQ.21440B)RNUM=3.1415927
0303 IF(LINE(IST).EQ.40040B)RNUM=2.7182818
0304 IF(RNUM.NE.0)GO TO 350
0305 IF(LINE(IST).NE.40440B.OR.LINE(IST+1).NE.47040B.OR.
0306 +LINE(IST+2).NE.51440B)GO TO 610
0307 RNUM=AS
0308 IF(LINE(IST+3).EQ.23440B)RNUM=AS2
0309 GO TO 850
0310 610 IF(LINE(IST).EQ.50440B.AND.LINE(IST+1).EQ.52040B)GO TO 1250
0311 IF(LINE(IST).EQ.51440B.AND.LINE(IST+1).EQ.52040B)GO TO 50
0312 IF(LINE(IST).NE.42040B.OR.LINE(IST+1).NE.42440B.OR.LINE(IST+2)
0313 +.NE.43440B)GO TO 620
0314 IDR=777
0315 GO TO 90
0316 620 IF(LINE(IST).NE.51040B.OR.LINE(IST+1).NE.40440B.OR.
0317 +LINE(IST+2).NE.42040B)GO TO 630
0318 IDR=888
0319 GO TO 90
0320 630 IF(LINE(IM).NE.20440B)GO TO 640
0321 IFACTO=777
0322 IM=IM-1
0323 640 IF(IST.NE.IM)GO TO 680
0324 LE=(LINE(IST)-16416)/256
0325 IF(IST.NE.1.OR.LINE(IST+1).NE.36440B)GO TO 650
0326 KADSB(J)=0
0327 LS=LE
0328 IST=IST+2
0329 GO TO 190
0330 650 IF(IV(LE).EQ.777.AND.LOOP.NE.777)GO TO 660

```

PAGE 0007 CALQS 11:47 AM THU., 28 AUG., 1980

```

0331      ISINGL=IST
0332      GO TO 685
0333  660  RNUM=V(LE)
0334      IALPHA=999
0335      WRITE(LU,670)LINE(IST),RNUM
0336  670  FORMAT(1X,A1,"=",F8.2)
0337      GO TO 850
0338  680  IF(LINE(IST).NE.264408)GO TO 681
0339      IST=IST+1
0340      NEGVAR=777
0341      GO TO 640
0342  681  IF(LINE(IST).EQ.530408)LOOP=777
0343  685  IF(IALPHA.EQ.999)GO TO 730
0344  690  WRITE(LU,700)
0345  700  FORMAT(" WHAT??")
0346      IF(IERR.EQ.777)GO TO 711
0347      IERR=777
0348      GO TO 90
0349  710  WRITE(LU,720)
0350  720  FORMAT(" (TYPE '%L' FOR LIST OF FUNCTIONS, 'QT' TO END)")
0351      GO TO 50
0352  730  D0740  L=1,80
0353  740  IWHAT(L)=LINE(L)
0354      D0750  L=1,10
0355      KWHAT(L)=KSNFU(L)
0356  750  KSNFU(L)=0
0357      WRITE(LU,760)
0358  760  FORMAT(" WHAT IS ' '")
0359      WRITE(LU,770)(IWHAT(L),L=IST,IM)
0360  770  FORMAT(A1," ")
0361      WRITE(LU,780)
0362  780  FORMAT(" ?"/" ?_")
0363      ISWCH=I
0364      IALPHA=J
0365      IHASSL=I2
0366      J=J+1
0367      GO TO 130
0368  790  IF(ISKIP.EQ.777)GO TO 850
0369      IF(LINE(IST).EQ.200408)IST=IST+1
0370      L=IM-IST
0371      IF(L.GE.0)GO TO 800
0372      RNUM=0
0373      IF(IM.EQ.0.AND.IALPHA.EQ.999)RNUM=ANS(J)
0374      GO TO 850
0375  800  IF(L.LE.9.OR.(L.LE.9.AND.LINE(IST).EQ.264408))GO TO 820
0376      WRITE(LU,810)
0377  810  FORMAT(" NUMBER HAS TOO MANY DIGITS")
0378      GO TO 50
0379  820  D0830  L=1,5
0380  830  IPACK(L)=200403
0381      CALL CODE
0382      WRITE(IPACK,840)(LINE(L),L=IST,IM)
0383  840  FORMAT(10A1)
0384      CALL CODE
0385      READ(IPACK,*):RNUM

```

PAGE 0018 CALQB 11:47 AM THU., 28 AUG., 1950

```

0386 850 IF(KSNFU(J).EQ.0)GO TO 940
0387 IF(KSNFU(J).EQ.20)RNUM=ABS(RNUM)
0388 IF(KSNFU(J).LT.7.OR.KSNFU(J).GT.18)GO TO 920
0389 IF(IDR.NE.999)GO TO 900
0390 IDR=777
0391 WRITE(LU,860)
0392 860 FORMAT(" IS FORMAT FOR TRIG FUNCTIONS TO BE IN DEGREES OR IN"
0393 " RADIANS?"/" (PLEASE NOTE THAT FORMAT CAN BE CHANGED AT "
0394 " ANY TIME ")
0395 870 WRITE(LU,880)
0396 880 FORMAT(" BY ENTERING 'DEG' OR 'RAD')_")
0397 READ(LU,890)L
0398 390 FORMAT(A1)
0399 IF(L.EQ.510408)IDR=888
0400 IF(IDR.NE.888.AND..L.NE.420408)GO TO 870
0401 900 IF(KSNFU(J).LT.10.OR.(KSNFU(J).GT.12.AND.KSNFU(J).LT.15)
0402 +OR.KSNFU(J).GT.18)GO TO 910
0403 IF(ABS(RNUM).LE.(1.0))X=SQRT(1-RNUM**2)
0404 IF(KSNFU(J).GE.16.AND.KSNFU(J).LE.18)RNUM=1.0/RNUM
0405 IF(KSNFU(J).EQ.10.OR.KSNFU(J).EQ.16)RNUM=ATAN(RNUM/X)
0406 IF(KSNFU(J).EQ.11.OR.KSNFU(J).EQ.17)RNUM=ATAN(X/RNUM)
0407 IF(KSNFU(J).EQ.12.OR.KSNFU(J).EQ.18)RNUM=ATAN(RNUM)
0408 IF(IDR.EQ.777)RNUM=RNUM/0.01745329
0409 GO TO 930
0410 910 IF(IDR.EQ.777)RNUM=RNUM*0.01745329
0411 IF(KSNFU(J).EQ.7.OR.KSNFU(J).EQ.13)RNUM=SIN(RNUM)
0412 IF(KSNFU(J).EQ.8.OR.KSNFU(J).EQ.14)RNUM=COS(RNUM)
0413 IF(KSNFU(J).EQ.9.OR.KSNFU(J).EQ.15)RNUM=TAN(RNUM)
0414 920 IF(KSNFU(J).EQ.30.OR.(KSNFU(J).GE.13.AND.
0415 +KSNFU(J).LE.15)RNUM=1.0/RNUM
0416 IF(KSNFU(J).GE.7.AND.KSNFU(J).LE.10)OR.KSNFU(J).EQ.30)GO TO 930
0417 IF(KSNFU(J).EQ.23)BASE=2.7182813
0418 IF(KSNFU(J).EQ.24)BASE=10.0
0419 IF(KSNFU(J).GE.23.AND.KSNFU(J).LE.25)RNUM=BASE**RNUM
0420 IF(RNUM.LT.(0.0))GO TO 690
0421 IF(KSNFU(J).EQ.6)RNUM=SQRT(RNUM)
0422 IF(RNUM.EQ.(0.0))GO TO 690
0423 IF(KSNFU(J).EQ.21.OR.KSNFU(J).EQ.26)RNUM=-LOG(RNUM)
0424 IF(KSNFU(J).EQ.26)RNUM=RNUM/ALOG(BASE)
0425 IF(KSNFU(J).EQ.22)RNUM=ALOGT(KNUM)
0426 930 KSNFU(J)=0
0427 940 IF(IFACT0.NE.777)GO TO 960
0428 IFACT0=888
0429 ISKIP=777
0430 F=1
0431 DO950 L=1,RNUM
0432 950 F=F*L
0433 RNUM=F
0434 GO TO 190
0435 955 WRITE(LU,956)RNUM
0436 956 FORMAT(" NUMBER TO THE POWER OF 'F9.4' CANNOT BE NEGATIVE")
0437 GO TO 90
0438 960 ISKIP=888
0439 IF(EXP(J).LT.(-9998))GO TO 965
0440 L=1

```

PAGE 0609 CALQ8 11:47 AM THU., 28 AUG., 1980

```

0441      IF(EXP(J).GE.0)GO TO 964
0442      EXP(J)=-EXP(J)
0443      IF(AINT(RNUM).NE.RNUM)GO TO 955
0444      IF(AINT(RNUM/2).NE.(RNUM/2))L=-L
0445      964 RNUM=L*EXP(J)*RNUM
0446      965 EXP(J)=-9999
0447      IF(IEXPFB(J).NE.777)GO TO 970
0448      IEXPFB(J)=888
0449      EXP(J)=RNUM
0450      IF(INVEXP(J).EQ.777)KSNFU(J)=30
0451      INVEXP(J)=888
0452      GO TO 180
0453      970 IF(MLDIV(J).EQ.0.AND.MLDIV2(J).EQ.0)GO TO 980
0454      IF(MLDIV2(J).EQ.3)RNUM=RNO(J)*RNUM
0455      IF(MLDIV2(J).EQ.4)RNUM=RNO(J)/RNUM
0456      MLDIV2(J)=MLDIV(J)
0457      IF(MLDIV2(J).EQ.0)GO TO 980
0458      MLDIV(J)=0
0459      RNO(J)=RNUM
0460      GO TO 180
0461      980 IF(KADSB2(J).EQ.0)ANS(J)=RNUM
0462      IF(KADSB2(J).EQ.1)ANS(J)=ANS(J)+RNUM
0463      IF(KADSB2(J).EQ.2)ANS(J)=ANS(J)-RNUM
0464      KADSB2(J)=KADSB(J)
0465      IF(KADSB2(J).EQ.99)GO TO 1090
0466      IF(KADSB2(J).NE.98)GO TO 180
0467      IF(J.EQ.1)GO TO 1060
0468      RNUM=ANS(J)
0469      J=J-1
0470      IF(LOGFU(J).EQ.777)GO TO 1040
0471      IF(LOGFU(J).EQ.666)GO TO 1030
0472      IF(J.EQ.ALPHA)GO TO 1000
0473      ISKIP=777
0474      GO TO 190
0475      1000 I=ISWTCB
0476      I2=IHASSL
0477      IALPHA=999
0478      IF(NEGVAR.EQ.777)RNUM=-RNUM
0479      NEGVAR=888
0480      DO1010 L=1,30
0481      1010 LINE(L)=IWHAT(L)
0482      DO1020 L=1,10
0483      1020 KSNFU(L)=KWHATF(L)
0484      IF(ISINGL.EQ.777)GO TO 850
0485      LE=(LINE(ISINGL)-16416)/256
0486      ISINGL=777
0487      V(LE)=RNUM
0488      IV(LE)=777
0489      GO TO 850
0490      1030 KSNFU(J)=25
0491      GO TO 1050
0492      1040 KSNFU(J)=26
0493      IF(RNUM.GT.0)GO TO 1050
0494      WRITE(LU,1045)RNUM
0495      1045 FORMAT(IX,F3.4," AS LOG BASE IS NOT POSITIVE")

```

PAGE 0010 CALQG 11:47 AM THU., 28 AUG., 1960

```
0496      GO TO 90
0497 1050  BASE=RNUM
0498      LOGFU(J)=888
0499      GO TO 180
0500 1050  IF(ISU.NE.777)GO TO 1111
0501      SUM=SUM+ANS(J)
0502      ANS(J)=0
0503      XADSB2(J)=0
0504      N=N+1
0505      IERR=888
0506      GO TO 90
0507 1070  I=0
0508      WRITE(LU,1090)
0509 1080  FORMAT(" & ")
0510      READ(LU,140)LINE
0511      GO TO 130
0512 1090  IF(I.NE.1)GO TO 1070
0513      IF(ISU.NE.777)GO TO 1110
0514      ANS(J)=SUM
0515      SUM=0
0516      WRITE(LU,1100)
0517 1100  FORMAT(" SUM_ ")
0518 1110  AS=AS
0519      AS=ANS(J)
0520      AB=ABS(AS)
0521      WRITE(LU,1120)
0522 1120  FORMAT(" = ")
0523      IF(AB.LT.(1E+7).AND.AB.GT.(1E-7).OR.AB.EQ.(0.0))GO TO 1160
0524      IF(AB.LT.(9.999E+37))GO TO 1140
0525      L=254408
0526      IF(AS.LT.0)L=254408
0527      WRITE(LU,1130)L
0528 1130  FORMAT(A1,"INFINITY")
0529      GO TO 1200
0530 1140  WRITE(LU,1150)AS
0531 1150  FORMAT(E9.7)
0532      GO TO 1200
0533 1160  IF(AB.GT.(32767.0))GO TO 1180
0534      IF((AINT(AS)).NE.AS)GO TO 1180
0535      L=AS
0536      WRITE(LU,1170)L
0537 1170  FORMAT(I8)
0538      GO TO 1200
0539 1180  WRITE(LU,1190)AS
0540 1190  FORMAT(F8.7)
0541 1200  IF(ISU.NE.777)GO TO 1220
0542      WRITE(LU,1210)N
0543 1210  FORMAT(" N=",I4)
0544      ISU=888
0545 1220  IF(IAS.NE.777)GO TO 1240
0546      ANS(J)=ANS(J)/N
0547      WRITE(LU,1230)
0548 1230  FORMAT(" AVERAGE_ ")
0549      IAV=888
0550      GO TO 1110
```

PAGE 0011 CALQB 11:47 AM THU., 29 AUG., 1980

```
0551 1240 N=0
0552      XADSB2(J)=0
0553      IERR=888
0554      GO TO 60
0555 1249 WRITE(LU,1248)
0556 1248 FORMAT(" GONE!")
0557 1250 CONTINUE
0558      END
```

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 05789 COMMON = 00000

CALQ8 24042 37276 REV 3 - 25 - 80 J 81RN

FMIO	37277	40575	24998-16002 REV.1926	790417
RMPAR	40576	40640	781106	24998-16001
ERR0	40641	40730	771122	24998-16001
TAN	40731	41035	780424	24998-16001
ABS	41036	41044	750701	24998-16001
SNCS	41045	41206	780424	24998-16001
AMT	41207	41307	750701	24998-16001
ALOG	41310	41422	780424	24998-16001
ALOGT	41423	41433	770518	24998-16001
SBRT	41434	41535	780424	24998-16001
ATAN	41536	41704	780424	24998-16001
RTOI	41705	42000	780921	24998-16001
FPWR	42001	42042	781106	24998-16001
ATOR	42043	42140	760301	24998-16001
FCM	42141	42155	750701	24998-16001
PNAME	42156	42223	771121	24998-16001
EROLE	42224	42224	750701	24998-16001
CMRS	42225	42310	780424	24998-16001
FRMTR	42311	45746	24998-16002 REV.1926	790503
FMTLE	45747	45747	24998-16002 REV.1931	781107
REIO	45750	46074	92067-16268 REV.1903	790316
EXP	46075	46213	780424	24998-16001
CFER	46214	46271	750701	24998-16001

11 PAGES RELOCATED 11 PAGES REQ'D NO PAGES EMA NO PAGES MSGS
LINKS:BP PROGRAM:BG LOAD:TE COMMON:NC
/LOADR:CALQ8 READY AT 11:51 AM THU., 28 AUG., 1980

/LOADR:\$END

**DATA
FILM**