LLL 8080 Basic Interpreter Program

PART II

By John Dickenson and Jerry Barber

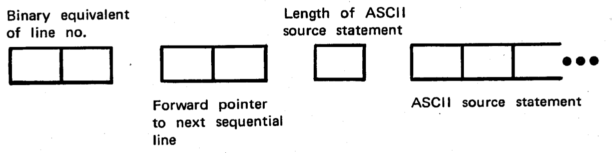
INTRODUCTION

This article is part #2 of a series of four articles covering the LLL 8080 BASIC Interpreter program released to the public domain by Lawrence Livermore Laboratories. This article covers the description of the BASIC Interpreter and includes the assembly listing of the LLL 8080 BASIC Interpreter program.

DESCRIPTION OF BASIC INTERPRETER

Following is a brief description of the BASIC interpreter. Hopefully, with this description, it will not be a major project to modify the BASIC to satisfy the reader's specific needs.

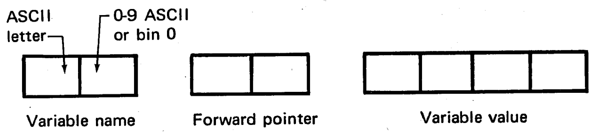
Formats — Source statements are stripped of blanks on input (character strings enclosed in " "s are an exception) and stored as is in memory, using the following format:


Binary equivalent of line no.
Length of ASCII source statement
Forward pointer to next sequential line
ASCII source statement

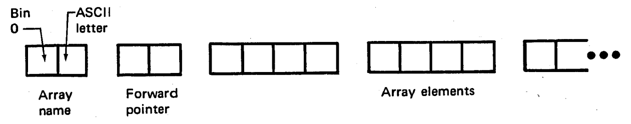
The forward pointer links statements by ascending line numbers. The last line's forward pointer (supposedly an end statement) has value 177777_8 to indicate end of the list.

The symbol table is built up at run time and begins after the most recently entered source statement (the variable STSPAC points to where the symbol table will start). Symbol table entries are shown below:

SCALAR-VARIABLE FORMAT



ARRAY-VARIABLE FORMAT



Subroutines — Following is a list of potentially useful subroutines, with a brief description of each subroutine:

ALPHA	— Value pointed to by H and L is tested to see if it is an ASCII letter. CY = 1 => Yes CY = 0 => No
NUMB	— Same as above but tests for a decimal number (ASCII 0-9).
CHAR2	— Inputs a character from the teletype to a register.
CHAR5	— Same as above for HSR (High Speed Paper Tape Reader).
CHK1	— Checks to see if HL are equal to 177777_8 (−1). CY = 1 => Yes CY = 0 => No.
CONV (CVRT)	— One of the floating-point routines. Converts floating-point number to a character string. Output is padded to the output buffer.
COPDH	— Copies floating-point number pointed to by D, E to location pointed to by H, L; uses copy.
COPY	— One of the floating-point routines. Copies floating-point value pointed to by A, L to location pointed to by H, C.
CUB	— Converts the integer-character string pointed to by H, L to its binary equivalent. Vale returns in D, E registers.
DCOMP	— Double-byte comparison routine. Compares value in CB to the value in ED.

Z = 1	=>	CB = ED
CY = 1	=>	CB > ED
CY = 0	=>	CB ≷ ED.

DFXL — One of the floating-point routines. Used to float an unsigned integer H, L point to first of four bytes; integer is right justified in first three bytes.

EVAL — Evaluates an expression the first element of which is pointed to by H, L and the length of which is in C. Used to evaluate expressions wherever they are legal in BASIC. C usually contains the length of the source statement line containing the expression.

FINPT — One of the floating-point routines. Converts character string to floating-point number. The variable HLINP contains a pointer to the character string, and the variable CREG contains the length of line containing character string. Mode = 0 => data comes from teletype (i.e., only delimiters are g's). Mode = 1 => data comes from source statements.

FIX — Fixes a floating-point number. DE points to number to be fixed. Error code 13 is given if number is too big to fix.

FSYM — Finds symbols in symbol table. BC contains symbol. Returns with HL pointing to symbol value.
CY = 1 => symbol was found.
CY = 0 and a scalar => symbol not found, but inserted and initialized to 0.
CY = 0 and an array => not found, no action taken: HL are meaningless.

LADD — Floating-point add routine.

LSUB — Floating-point subtract routine.

LOIU — Floating-point divide routine.

LMUL — Floating-point multiply routine.

LMCM — One of the floating-point routines. Compares two floating-point values, HL Point to first floating-point values and HB point to second floating-point value.
z =1 => Equality
Cy=1 => first < second
(Note: compares absolute only, does not reference mantissa sign.)

MCHK — Waits for flag from port 3. Proper mask is sent in register B.

MEMFUL — Checks to see if memory is full. HL point to location of memory to be checked. Memory is considered full if it is within 50_{10} locations of the current value of stack pointer.

MULT — Multiplies two two-byte binary numbers. HL point to last byte of four bytes. First two contain first number. Last two contain second number. Answer returns in BCDE.

NSRCH — Routine to locate source line in memory passed binary value of line number in DE. Returns address of line in HL, CY=1 => not found.

OUTR — Used by CONV (CURT) to pad output to output buffer.

PAD — Pads characters to output buffer. A contains character; B contains number of pads.

SYMSRT — Checks a character string to see if it is a BASIC symbol. HL contains address pointing to 1st character of symbol, C contains length of line that contains symbol. A contains type of symbol sought.

0=command	1=keyword
z=operator or delimiter	3=function

Returns with 377_8 in a register if nothing found. Otherwise A contains symbol number in appropriate KDAT table. Thus, for symbol type 2, if a 4 is returned, the symbol found was the fourth one (starting with 0) in table KDAT3 (KDAT concatenated with 2 and 1 or A'). CIS is updated, but HL is not.

TTYIW — Inputs a line from teletype. Stores starting address at location pointed to by HL. Line edits. Returns length of line in A register (maximum line length is 72 characters).

VALUE — Called with HL pointing to A variable, constant, or function; C contains line length, returns with DE pointing to floating-point value. HL, C are updated.

VAR — Called with HL pointing to character string, C has line length. Determines if character string is a variable. If so, returns with CY=1, DE pointing to value (subscripts of arrays are evaluated, etc.). HL, C updated. If not, a variable returns CY=0, HL, C untouched.

WRIT — Dumps contents of output buffer to teletype. Uses entry WRIT1 with D register equal to one to suppress CR/LF.

ZROL — Part of floating-point subroutines. Writes a floating-point zero, starting at location pointed to by HL.

The preceding list contains those subroutines most likely to be used by someone modifying BASIC. If you plan on using one of the routines, you should examine it and its comments carefully.

Variables — Following is a list of interpreter variables, with a description of each variable:

MEMST — Assembly time variable. Contains the first available RAM location. This is where active variables start.

MEMEND — Assembly time variable. Contains the last available location in RAM.

SEND	— Has value 6, used with RST instruction to print characters via ODT.
OBUFF	— Output buffer, the first location contains the number of characters in the buffer + 1.
IBUF	— Input buffer, occupies same area as OBUFF.
STLINE	— Points to first source line to be executed. If no source, contains 177777_8.
NLINE, NL2, NL4, NL6	— Contain address, binary-equivalent line number, forward pointer, and length of next input line.
KLINE, KL2, KL4, KL6	— Same as above, but used by a subroutine that inserts lines in sequential order (insert).
PLINE, PL2, PL4, PL6	— Subroutine insert to order statements sequentially.
KASE, LEN	— Temporary storage for command mode routines.
MULT1, MULT2	— Used to store binary values to be multiplied.
SBSAV	— Temporary storage for call-statement processor.
STSPAC	— Next available location in memory, symbol table starts here at run time.
LPNT	— Pointer to the current line at run time.
CPNT	— Pointer to current character in current line at run time.
KFPNT	— Point to next sequential line at run time.
FREG1, FREG2	— Two floating-point registers.
HLINP, CREG	— Temporary storage for HL and C registers for routine INP.
NXTSP	— Pointer to next available space of memory for symbol table.
GREG	— General register, in and out instructions are stored here and executed for get and put functions.
MODE	— Indicates to INP routine whether input data comes from source or teletype.
MESCR	— Temporary storage for call-statement processor. Points to next available space after symbol table. Area after the symbol table is used to store intermediate results of expressions or constants passed to user subroutines.
VARAD	— Temporary storage space for input-statement processor.
VEND	— Assembly time variable. Indicates end of interpreter variable-storage area and where FWAM pointer is to go.
FWAM	— First word of available memory pointer. This is where user source programs go.

Some of the above variables occupy the same area of memory. This is because some variables are used only in the command mode and others only at runtime. To conserve space, they share the same memory locations.

New BASIC Statements — To add additional statements to the BASIC, use the following procedure. First, insert the statement keyword in the data tables for subroutine SYMSRT. Then, insert the starting address of the statement processor in the interpreter JUMP table. Finally, the statement processor itself must be inserted.

The keyword must be entered in the table KDAT2. The first byte must be the keyword length and the next bytes hold the ASCII-coded keyword. The table must end with A 377_8. If the keyword is the Nth entry in the table, on return from SYMSRT, the A register will hold N-1 if the keyword is found.

The starting address of the statement processor must be inserted into table JTBL. The order of keywords in KDAT2 must correspond with statement processor addresses in JTBL since, on return from SYMSRT, the A register times two is used as offset in JTBL to determine processor address.

The statement processor must be placed somewhere in memory. Generally, the first thing done in the statement processors is to load the pointer to the statement (LHLD CPNT) and increment past the keyword (since HL is not updated by SYMSRT). On entry, C contains the number of characters in the line minus those checked by SYMSRT. The end of the processor should be a "JMPIEND" instruction.

New Functions — New functions must be added to SYSSRT Data Table KDAT4 in the same manner as for key words. The function itself must be placed in subroutine "VALUE." Presently, the only function in VALUE is GET.

Message Lines — The following description tells how to incorporate messages into BASIC output routines. Currently, to output a message to the teletype, the user executes an LXI H,ODATA, then a call to FORMK where K is an integer indicating which message is wanted (i.e., K=z indicates "TURN ON PUNCH"). FORM pads the message into the output buffer. Then A "CALL WRIT" writes the contents of the buffer.

Suppose the message "POTATO BASIC" is to be added. Preceding the form 9 instruction, we will insert "FOR10: INR L." At the end of the ODATA table, we add "DB ODAT8 and 377Q.". And, after message ODAT 7, we add ODAT8 DB *12, "POTATO BASIC." Now, the following program segment:

```
LXI    H,ODATA
CALL   FOR10
CALL   WRIT,
```

will cause "POTATO BASIC" to be output to the teletype.

SEE MICROCOMPUTER SOFTWARE DEPOSITORY PROGRAM INDEX FOR COPIES OF THIS PROGRAM

```
8080 MACRO ASSEMBLER, VER 2.1  ERRORS = 0
                        ;
                        ;       BAS80 - BASIC INTERPRETER FOR INTEL 8080 MICRO-PR
                        ;
                        ;        WRITTEN AT THE UNIVERSITY OF IDAHO BY:
                        ;
                        ;          GERALD R. BARBER
                        ;          DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER
                        ;          MASSACHUSETTES INSTITUTE OF TECHNOLOGY
                        ;          CAMBRIDGE, MASSACHUTTES
                        ;
                        ;          JOHN A. TEETER
                        ;          EAST OF CASCADE
                        ;          SOUTH OF LANDMARK
                        ;          DEADWOOD, IDAHO
                        ;
                        ;          JOHN W. DICKINSON
                        ;          DEPT. OF ELECTRICAL ENGINEERING
                        ;          UNIVERSITY OF IDAHO
                        ;          MOSCOW, IDAHO
                        ;
                        ;
                        ;       THIS FILE CONTAINS THE EDIT AND EXECUTION PORTION
                        ;       OF THE BASIC INTERPRETER.  IT IS DESIGNED TO EXEC
                        ;       WITH THE COMPANION FILE CONTAINING THE FLOATING
                        ;       POINT ROUTINES.
                        ;
                        ;       I/O IS ACCOMPLISHED WITH A TTY AND A HIGH SPEED
                        ;       PAPER TAPE READER.  TTY I/O IS DONE THROUGH
                        ;       ODT ROUTINES.  PAPER TAPE INPUT IS HANDLED
                        ;       MANUALY.
                        ;
                        ; DEFINE LINKAGES FROM FLT PNT PACKAGE TO I/O ROUTINES
                        ;
0FFA                            ORG     7772Q
0FFA    C3 73 19                JMP     INP
0FFD    C3 8B 17                JMP     OUTR
                        ;
                        ; DEFINE ADDRESSES OF ACTIVE VARIABLES
                        ;
2100                    MEMST   EQU     20400Q          ;MUST BE ON PAGE BOUNDA
27FF                    MEMEND  EQU     23777Q          ;END OF RAM MEMORY
0006                    SEND    EQU     6               ;RST FOR ODT
2100                    OBUFF   EQU     MEMST           ;INPUT AND OUTPUT BUFFE
2101                    IBUF    EQU     MEMST+1         ;SAME AREA
2149                    STLINE  EQU     MEMST+111Q      ;FIRST LINE OF USER PRO
214B                    NLINE   EQU     MEMST+113Q      ;POINTER TO NEW LINE OF
214D                    NL2     EQU     MEMST+115Q      ;BINARY VALUE OF NEW LI
214F                    NL4     EQU     MEMST+117Q      ;FORWARD POINTER OF NEW

2151                    NL6     EQU     MEMST+121Q      ;LENGTH OF NEW LINE
2152                    KLINE   EQU     MEMST+122Q      ;POINTER TO CURRENT LIN
2154                    KL2     EQU     MEMST+124Q      ;BINARY EQU. OF CURRENT
2156                    KL4     EQU     MEMST+126Q      ;FORWARD POINTER OF CUR
2158                    KL6     EQU     MEMST+130Q      ;LENGTH OF NEW LINE
2159                    PLINE   EQU     MEMST+131Q      ;PREVIOUS LINE POINTER-
215B                    PL2     EQU     MEMST+133Q      ;BINARY LINE NUMBER OF
215D                    PL4     EQU     MEMST+135Q      ;FORWARD POINTER OF PRE
215D                    SBSAV   EQU     PL4             ;RETURN ADD. SAVE FOR C
215F                    PL6     EQU     MEMST+137Q      ;LENGTH OF PREVIOUS LIN
2160                    KASE    EQU     MEMST+140Q      ;SPARE STORAGE USED AS
2161                    LEN     EQU     MEMST+141Q      ;ALSO SPARE-OFTEN LENGT
2162                    MULT1   EQU     MEMST+142Q      ;FIRST OF TWO WORDS USE
2164                    MULT2   EQU     MEMST+144Q      ;SECOND WORD
2159                    NXTSP   EQU     PLINE           ;POINTER TO NEXT AVAILA
214B                    STSPAC  EQU     NLINE           ;START OF SYM. TABLE
0FCD                    ZROL    EQU     7715Q           ;FLOATING PNT. ROUTINE
2152                    LPNT    EQU     KLINE           ;SOURCE LINE PTR DURING
2158                    KLEN    EQU     KL6             ;SPARE-USED AS NEEDED
215B                    CPNT    EQU     PL2             ;CHAR. PTR DURING EXEC.
2156                    KFPNT   EQU     KL4             ;FWD LINE PTR DURING EX
2180                    FREG2   EQU     MEMST+200Q      ;FLOATING POINT REG. #2
2184                    CREG    EQU     MEMST+204Q      ;TEMP. SPACE FOR ROUTIN
0FD0                    LADD    EQU     7720Q           ;FLOATING POINT ROUTINE
0FD3                    LMUL    EQU     7723Q           ;FLOATING POINT ROUTINE
0FD6                    LDIV    EQU     7726Q           ;FLOATING POINT ROUTINE
0FD9                    LSUB    EQU     7731Q           ;FLOATING POINT ROUTINE
0FDC                    DFXL    EQU     7734Q           ;FLOATING POINT ROUTINE
0FDF                    LMCM    EQU     7737Q           ;FLOATING POINT ROUTINE
2186                    HLINP   EQU     MEMST+206Q      ;TEMP. SPACE FOR ROUTIN
2177                    GREG    EQU     MEMST+167Q      ;GENERAL PURPOSE REGIST
217C                    FREG1   EQU     MEMST+174Q      ;FLOATING PNT. REGISTER
2166                    SCR     EQU     MEMST+146Q      ;SCRATCH AREA FOR I/O R
0FE5                    CONV    EQU     7745Q           ;FLOATING PNT. OUTPUT R
2185                    MODF    EQU     MEMST+205Q      ;MODE FLAG FOR ROUTINE
0FE8                    FINPT   EQU     7750Q           ;FLOATING PNT. INPUT RO
0FEB                    MULT    EQU     7753Q           ;ADD. VALUE MULT. ROUTI
0FEE                    PTVAL   EQU     7756Q           ;DEFINE LPNT,KFPNT ETC.
0FF1                    DCOMP   EQU     7761Q           ;DOUBLE BYTE COMPARISON
0FF4                    MCHK    EQU     7764Q           ;CHECK MASK
0FF7                    CHAR2   EQU     7767Q           ;INPUT CHAR. FROM TTY
2188                    MESCR   EQU     MEMST+210Q      ;DEFINE MEMORY SCR AREA
218A                    VARAD   EQU     MEMST+212Q      ;TEMP SPACE FOR INP. ST
218C                    VNAME   EQU     MEMST+214Q      ;TEMP SPACE FOR 'FOR-NE
218E                    VLOC    EQU     MEMST+216Q      ;TEMP SPACE FOR 'FOR-NE
2190                    FLIMT   EQU     MEMST+220Q      ;TEMP SPACE FOR 'FOR-NE
2194                    NEST    EQU     MEMST+224Q      ;NESTING STACK-POINTER
2196                    STAC    EQU     MEMST+226Q      ;FOR-NEXT NESTING STACK
0014                    STSIZ   SET     20              ;STACK SIZE, ALLOWS 10
2196                    TOPNS   EQU     STAC            ;TOP OF STACK
21AA                    BOTNS   EQU     STAC+STSIZ      ;BOTTOM OF STACK

21AA                    VEND    EQU     MEMST+252Q      ;DEF. END OF VAR. STORA
                        ;
                        ; DEFINE FWA MEMORY AND START OF SUBTABLE
                        ;
21AA                            ORG     VEND
21AA    AD 21           FWAM:   DW      SCMEM
21AC    FF              SUBS:   DB      377Q            ;NOTHING IN SUB TABLE
21AD                    SCMEM   EQU     $
                        ;
                        ; MAIN ROUTINE--HANDLES ALL USER INPUT
                        ;
1000                            ORG     10000Q
                        ;
1000    21 00 21        M1:     LXI     H,OBUFF         ;INITIALIXE THE OUTPUT
1003    36 01                   MVI     M,1             ;TO HAVE LENGTH ONE.
1005    21 49 21                LXI     H,STLINE        ;PUT -1 INTO STLINE
1008    36 FF                   MVI     M,377Q
100A    2C                      INR     L
100B    36 FF                   MVI     M,377Q          ;-1 IS ALWAYS END OF LI
100D    2A AA 21                LHLD    FWAM            ;GET ADDRES OF FWA MEM.
1010    22 4B 21                SHLD    NLINE           ;STORE IN FREE SPACE PN
1013    31 F5 27        M1A:    LXI     SP,MEMEND-10    ;SET STACK TO HIGH MEMO
1016    21 5A 14        M2:     LXI     H,ODATA         ;SET HL TO ODATA (ASCII
1019    CD 48 14                CALL    FORM1           ;PAD "READY"
101C    CD D0 12                CALL    WRIT
101F    2A 4B 21        M3:     LHLD    NLINE           ;GET NEXT LINE FOR ACCE
1022    23                      INX     H
1023    23                      INX     H               ;MOVE PAST POINTERS TO
1024    23                      INX     H
1025    23                      INX     H
1026    23                      INX     H
1027    CD C9 13                CALL    TTYIN           ;GET INPUT FROM TTY
102A    4F                      MOV     C,A             ;MOVE NUMBER OF CHAR TO
102B    FE 00                   CPI     0               ;IS IT 0
102D    CA 1F 10                JZ      M3              ;YEP SO TRY AGAIN
1030    CD 22 12                CALL    ALPHA           ;IS INPUT AN ALPHA STRI
1033    DA 42 10                JC      M4              ;YEP--GO DECIPHER
1036    CD 14 12                CALL    NUMB            ;IS INPUT A NUMERIC STR
1039    D4 9E 11                CNC     WHAT            ;NOPE -- FIND OUT WHATS
103C    CD 8F 10                CALL    INSERT          ;YEP-- GO INSERT NEW LI
103F    C3 1F 10                JMP     M3              ;LOOP ON PROCESS
1042    3E 00           M4:     MVI     A,0             ;SET UP SYMBOL TYPE A=0
1044    CD F7 12                CALL    SYMSRT          ;GO FIND COMMAND
1047    3C              M4A:    INR     A
1048    CC 9E 11                CZ      WHAT            ;REG. A RETURNED WITH -
104B    3D                      DCR     A
104C    CA 78 16                JZ      RUN             ;ITS A RUN COMMAND
104F    3D                      DCR     A
1050    CC AA 11                CZ      TAPE            ;THE DUDE WANTS A PLST
1053    CA 16 10                JZ      M2
1056    3D                      DCR     A

1057    CC D1 11                CZ      LIST            ;HE WANTS A LISTING
105A    CA 16 10                JZ      M2
105D    3D                      DCR     A
105E    CA 00 10                JZ      M1              ;IT WAS A PURGE COMMAND
1061    3D                      DCR     A
1062    C4 9E 11                CNZ     WHAT            ;IF ITS NOT ZERO--THEN
                                                        ;ELSE ITS PTAPE
                        ;
                        ; ROUTINE TO INPUT FROM HSR
                        ; THIS CONDITION IS DEFAULT NOW
                        ;
1065    CD 08 15        PTAPE:  CALL    CHAR5           ;GET CHAR. FROM READER
1068    FE 00                   CPI     0               ;IS IT LEADER?
106A    CA 65 10                JZ      PTAPE           ;YEP -- WAIT FOR GOOD I
106D    CD D5 14        PT1:    CALL    HSRIN           ;GOT CHAR.
1070    4F                      MOV     C,A
1071    FE 00                   CPI     0               ;IS IT STILL LEADER?
1073    CA 65 10                JZ      PTAPE           ;YEP-- KEEP WAITING
1076    CD 22 12                CALL    ALPHA           ;IS IT ALPHA STRING?
1079    DA 42 10                JC      M4              ;YEP-- SO GO INTERPRET
107C    CD 8F 10                CALL    INSERT          ;NOPE--SO GO INSERT IT
107F    CD 08 15                CALL    CHAR5           ;GET NEXT CHAR.
1082    FE 00                   CPI     0               ;IS IT LEADER?
1084    CA 16 10                JZ      M2              ;YEP--GO BACK TO KEYBOA
1087    23                      INX     H
1088    23                      INX     H               ;MOVE PAST POINTERS OF
1089    23                      INX     H
108A    23                      INX     H
108B    23                      INX     H
108C    C3 6D 10                JMP     PT1             ;LOOP ON PROCESS
                        ;
                        ; ROUTINE TO HANDLE ALL SOURCE LINE INPUT.
                        ; THIS INCLUDES INSERTION, DELEATION, AND
                        ; ADDITION OF NEW SOURCE LINES.
                        ;
108F    2B              INSERT: DCX     H               ;THE ROUTINE SORTS LINI
1090    71                      MOV     M,C             ;THE TEXT USING THE FOR
1091    23                      INX     H               ;STORED IN THE TABLE.
1092    CD 2A 12                CALL    CVB             ;GET BINARY FOR NEW LIN
1095    FE 05                   CPI     5               ;IS IT MIN. VALUE?
1097    DA A2 10                JC      ISR1A           ;ITS A VALID NUMBER
109A    C4 9E 11                CNZ     WHAT            ;INVALID NUMBER--REPORT
109D    7B                      MOV     A,E
109E    17                      RAL
109F    DC 9E 11                CC      WHAT
10A2    2A 4B 21        ISR1A:  LHLD    NLINE           ;POINT TO NEW LINE
10A5    72                      MOV     M,D
10A6    23                      INX     H
10A7    73                      MOV     M,E
10A8    21 4B 21                LXI     H,NLINE
10AB    CD EE 0F                CALL    PTVAL           ;FIND THE POINTER VALUE
10AE    2A 49 21                LHLD    STLINE          ;GET START OF SOURCE
10B1    CD A3 12                CALL    CHK1            ;CHECK FOR NO TEXT
10B4    D2 D4 10                JNC     ISRT3           ;THERE IS SOMETHING-STA
10B7    2A 4B 21                LHLD    NLINE           ;THIS IS FIRST LINE
10BA    22 49 21                SHLD    STLINE          ;MOVE NLINE TO STLINE
10BD    16 FF           ISRT1:  MVI     D,377Q          ;SET UP -1 FOR END OF L
10BF    5A                      MOV     E,D
10C0    CD 8B 11                CALL    STPNT           ;STORE IT AWAY IN POINT
10C3    23                      INX     H
10C4    7E              ISRT2:  MOV     A,M             ;GETLINE LENGTH
10C5    C6 05                   ADI     5               ;ADJUST TO LEN+5
10C7    2A 4B 21                LHLD    NLINE
10CA    85                      ADD     L
10CB    6F                      MOV     L,A
10CC    3E 00                   MVI     A,0
10CE    8C                      ADC     H
10CF    67                      MOV     H,A
10D0    22 4B 21                SHLD    NLINE           ;STEP PAST NEW LINE
10D3    C9                      RET
10D4    22 52 21        ISRT3:  SHLD    KLINE           ;SET UP CURRENT LINE
10D7    21 52 21        ISRT4:  LXI     H,KLINE         ;GET CURRENT LINE AND
10DA    CD EE 0F                CALL    PTVAL           ;SET UP POINTERS
10DD    21 4D 21                LXI     H,NL2           ;GET BINARY VALUE
10E0    56                      MOV     D,M             ;OF NEW LINE
10E1    2C                      INR     L
10E2    5E                      MOV     E,M
10E3    21 54 21                LXI     H,KL2           ;GET CURRENT LINE BINAR
10E6    46                      MOV     B,M
10E7    2C                      INR     L
10E8    4E                      MOV     C,M
10E9    CD F1 0F                CALL    DCOMP           ;HOW DO THEY COMPARE?
10EC    CA 21 11                JZ      ISRT6           ;REPLACE/DELETE
10EF    DA 7A 11                JC      ISR12           ;KLINE>NLINE
10F2    2A 56 21                LHLD    KL4             ;MOVE ON TO NEXT LINE
10F5    CD A3 12                CALL    CHK1            ;ARE WE DONE?
10F8    DA 0F 11                JC      ISRT5
10FB    E5                      PUSH    H
10FC    2A 52 21                LHLD    KLINE
10FF    22 59 21                SHLD    PLINE           ;SET UP PREVIOUS LINE
1102    21 59 21                LXI     H,PLINE
1105    CD EE 0F                CALL    PTVAL           ;SET POINTERS
1108    E1                      POP     H
1109    22 52 21                SHLD    KLINE           ;SET UP CURRENT LINE
110C    C3 D7 10                JMP     ISRT4           ;LOOP ON PROCESS
110F    2A 4B 21        ISRT5:  LHLD    NLINE           ;GET NLINE ADDRESS
1112    CD 91 11                CALL    NOLINE          ;IS IT 0 LEN. LINE?
1115    C8                      RZ                      ;YES-DO NOTHING
1116    EB                      XCHG
1117    2A 52 21                LHLD    KLINE           ;GET CURRENT LINE
111A    CD 8B 11                CALL    STPNT           ;STORE AWAY POINTERS
111D    EB                      XCHG
111E    C3 BD 10                JMP     ISRT1           ;GO SET END OF LIST
1121    2A 4B 21        ISRT6:  LHLD    NLINE           ;REPLACE OR DELETE
1124    CD 91 11                CALL    NOLINE          ;NOLINE MEANS DELETE
1127    C2 48 11                JNZ     ISRT8           ;GO REPLACE IF NOT 0
112A    2A 49 21                LHLD    STLINE          ;GET STLINE
112D    EB                      XCHG
112E    2A 52 21                LHLD    KLINE           ;IF STLINE=KLINE THEN L
1131    E5                      PUSH    H               ;NOW ZERO
1132    C1                      POP     B
1133    CD F1 0F                CALL    DCOMP
1136    2A 56 21                LHLD    KL4
1139    CA 44 11                JZ      ISRT7           ;IF NOT STLINE THEN GO
113C    EB                      XCHG
113D    2A 59 21                LHLD    PLINE
1140    CD 8B 11                CALL    STPNT
1143    C9                      RET
1144    22 49 21        ISRT7:  SHLD    STLINE
1147    C9                      RET
1148    2A 56 21        ISRT8:  LHLD    KL4             ;REPLACE LINE
114B    EB                      XCHG
114C    2A 4B 21                LHLD    NLINE
114F    CD 8B 11                CALL    STPNT
1152    2A 52 21        ISRT9:  LHLD    KLINE
1155    EB                      XCHG
1156    2A 49 21                LHLD    STLINE
1159    E5                      PUSH    H
115A    C1                      POP     B
115B    CD F1 0F                CALL    DCOMP
115E    CA 71 11                JZ      ISR11           ;IF KLINE =STLINE GOTO
1161    2A 4B 21                LHLD    NLINE
1164    EB                      XCHG
1165    2A 59 21                LHLD    PLINE
1168    CD 8B 11                CALL    STPNT           ;PREVIOUS LINE PNTS. TO
116B    21 51 21        ISR10:  LXI     H,NL6           ;SET UP NEW LINE TO
116E    C3 C4 10                JMP     ISRT2           ;POINT TO M(KL4)
1171    2A 4B 21        ISR11:  LHLD    NLINE           ;NLINE=STLINE
1174    22 49 21                SHLD    STLINE
1177    C3 6B 11                JMP     ISR10
117A    2A 52 21        ISR12:  LHLD    KLINE           ;NLINE IN FRONT OF KLIN
117D    EB                      XCHG
117E    2A 4B 21                LHLD    NLINE
1181    CD 91 11                CALL    NOLINE
1184    C8                      RZ
1185    CD 8B 11                CALL    STPNT
1188    C3 52 11                JMP     ISRT9
                        ;
                        ; ROUTINE TO STORE POINTERS INTO MEM ARRAY
                        ; HL CONTAIN ADDR. OF FIRST WORD OF ENTRY.
                        ; THE CORESPONDING ELEMENTS (HL2,HL4,HL6) CONTAIN
                        ; THE NEW ENTRIES.
                        ;
118B    23              STPNT:  INX     H
118C    23                      INX     H
118D    73                      MOV     M,E             ;MOVE HL2/NL2/PL2 INTO
118E    23                      INX     H
118F    72                      MOV     M,D
1190    C9                      RET
                        ;
                        ; ROUTINE TO CHECK NEW LINE FOR SOURCE STMT.
                        ;
1191    E5              NOLINE: PUSH    H
1192    23                      INX     H
1193    23                      INX     H
1194    23                      INX     H
1195    23                      INX     H
1196    4E                      MOV     C,M             ;MOVE LENGTH OF LINE IN
1197    23                      INX     H
1198    CD 71 12                CALL    LENGTH          ;POINT TO FIRST CHAR.
119B    E1                      POP     H
119C    B9                      CMP     C               ;COMPARE LEN TO C
119D    C9                      RET
```

```
                          ;
                          ; ROUTINE TO RESPOND WITH 'WHAT?' FOR UNIDENTIFIED
                          ; COMMAND.
                          ;
119E     21 5A 14         WHAT:     LXI     H,ODATA          ;GET OUTPUT BUFFER ADDR
11A1     CD 44 14                   CALL    FORM7            ;PAD IT
11A4     CD D0 12                   CALL    WRIT             ;DUMP IT
11A7     C3 13 10                   JMP     MIA              ;JUMP TO RESTART.
                          ;
                          ; ROUTINE TO PUNCH PAPER TAPE OF SOURCE.
                          ; CALLED IN RESPONSE TO TAPE COMMAND
                          ;
11AA     F5               TAPE:     PUSH    PSW
11AB     C5                         PUSH    B
11AC     21 5A 14                   LXI     H,ODATA          ;GET OUTPUT BUFFER ADDR
11AF     CD 47 14                   CALL    FORM2            ;PAD IN TURN ON PUNCH
11B2     CD D0 12                   CALL    WRIT             ;DUMP IT
11B5     3E 00                      MVI     A,0              ;SET UP TO DUMP LEADER
11B7     C1                         POP     B
11B8     06 40                      MVI     B,100Q
11BA     F5                         PUSH    PSW
11BB     C5                         PUSH    B
11BC     CD AD 12                   CALL    PAD              ;PAD 100 0'S
11BF     CD D0 12                   CALL    WRIT
11C2     C1                         POP     B
11C3     C5                         PUSH    B
11C4     CD D1 11                   CALL    LIST             ;GO DO STANDARD LIST
11C7     C1                         POP     B
11C8     F1                         POP     PSW
11C9     CD AD 12                   CALL    PAD              ;GO PAD AND DUMP TRAILE
11CC     CD D0 12                   CALL    WRIT
11CF     F1                         POP     PSW
11D0     C9                         RET
                          ;
                          ; ROUTINE TO LIST TO TTY THE SOURCE STMTS.
                          ; ALSO USED TO DUMP TAPE.
                          ; ALLOWS THE USER TO INPUT FIRST LINE OR
                          ; OR FIRST AND LAST LINES.
                          ;
11D1     2A 49 21         LIST:     LHLD    STLINE
11D4     CD A3 12                   CALL    CHK1             ;CHECK FOR NO SOURCE
11D7     DA 13 10                   JC      MIA              ;GO RESTART
11DA     22 59 21                   SHLD    PLINE            ;SET UP TO START LOOKIN
11DD     21 FF FF                   LXI     H,177777Q
11E0     22 52 21                   SHLD    KLINE
11E3     0D                         DCR     C
11E4     C4 4D 15                   CNZ     BOUND            ;GO FIND FIRST LINE
11E7     2A 59 21                   LHLD    PLINE            ;STORE IT AWAY
11EA     23               LIS1:     INX     H
11EB     23                         INX     H
11EC     46                         MOV     B,M
11ED     23                         INX     H                ;GET LEN. IN REG B
11EE     4E                         MOV     C,M
11EF     C5                         PUSH    B
11F0     23                         INX     H
11F1     CD 49 14                   CALL    FORM5            ;GO PAD LINE
11F4     CD D0 12                   CALL    WRIT             ;GO DUMP LINE
11F7     C1                         POP     B
11F8     2A 52 21                   LHLD    KLINE            ;CHECK FOR LAST LINE
11FB     EB                         XCHG
11FC     CD F1 0F                   CALL    DCOMP
11FF     C8                         RZ
1200     68                         MOV     L,B
1201     61                         MOV     H,C              ;MOVE TO NEXT LINE
1202     CD 08 12                   CALL    QUITT            ;CHECK FOR INTERRUPTION
1205     C3 EA 11                   JMP     LIS1             ;NONE - CONTINUE
                          ;
                          ; THIS ROUTINE CHECKS PORT 2 FOR A CNTRL/S CHARACTER
                          ; IF ONE IS FOUND THEN EXECUTION IS TO BE INTERRUPTED
                          ; CONTROL IS PASSED TO MIA
                          ;
1208     DB 03            QUITT:    IN      3                ;TEST FLAG PORT
120A     1F                         RAR                      ;FLAG TO CY
120B     D0                         RNC                      ;NOTHING THERE
120C     DB 02                      IN      2                ;FLAG WAS SET, GET DATA
120E     FE 93            QTCHK:    CPI     223Q             ;WAS IT CNTRL/S?
1210     CA 13 10                   JZ      MIA              ;YES
1213     C9                         RET                      ;NO, RETURN
                          ;
                          ; ROUTINES NUMB AND ALPHA CHECK IF CONTENTS OF MEMORY
                          ; LOCATION IN HL CONTAIN ASCII NUMERIC OR ALPHBETIC
                          ; CHARACTER. RETURN CY=1 IF YES, CY=0 IF NO.
                          ;
1214     C5               NUMB:     PUSH    B
1215     06 B0                      MVI     B,260Q           ;SET UP ASCII 0
1217     0E BA                      MVI     C,272Q
1219     7E               C1:       MOV     A,M
121A     B8                         CMP     B
121B     3F                         CMC
121C     D2 20 12                   JNC     BAC
121F     B9                         CMP     C
1220     C1               BAC:      POP     B
1221     C9                         RET
1222     C5               ALPHA:    PUSH    B
1223     06 C1                      MVI     B,301Q           ;SET UP ASCII A
1225     0E DB                      MVI     C,333Q
1227     C3 19 12                   JMP     C1
                          ;
                          ; ROUTINE TO CONVERT ASCII NUMERIC CHAR. STRING TO
                          ; EQUIVALENT BINARY NUMBER. RETURNS EQUIVALENT IN
                          ; DE REG.  LENGTH OF LINE PASSED IN REG C AND
                          ; RETURNED POINTING TO LAST NUMERIC CHAR. LENGTH
                          ; OF CHAR STRING RETURNED IN REG A.
                          ;
122A     E5               CVB:      PUSH    H
122B     C5                         PUSH    B
122C     CD 71 12                   CALL    LENGTH           ;GET LENGTH OF STRING
122F     F5                         PUSH    PSW
1230     E5                         PUSH    H
1231     FE 00                      CPI     0                ;IS LENGTH=0?
1233     CA 68 12                   JZ      CVB2             ;YEP--GO TO CVB2
1236     21 60 21                   LXI     H,KASE           ;SET UP TEMP VAR KASE
1239     77                         MOV     M,A              ;SAVE LENGTH IN KASE
123A     2C                         INR     L                ;SET UP TO MULT. BY
123B     71                         MOV     M,C              ;DEC 10
123C     21 0A 00                   LXI     H,10
123F     22 62 21                   SHLD    MULT1
1242     21 00 00                   LXI     H,0
1245     22 64 21                   SHLD    MULT2
1248     21 65 21                   LXI     H,MULT2+1
124B     CD EB 0F         CVB1:     CALL    MULT
124E     E3                         XTHL
124F     7E                         MOV     A,M              ;GET NEXT CHAR. AND
1250     DE B0                      SBI     260Q             ;STRIP ASCII FROM IT
1252     82                         ADD     D                ;THE CONVERSION ALGORIT
1253     57                         MOV     D,A
1254     3E 00                      MVI     A,0              ;DE?0
1256     8B                         ADC     E                ;DO WHILE A>0
1257     5F                         MOV     E,A              ;     DE?(HL(M)-260B)+D
1258     23                         INX     H                ;    END:
1259     E3                         XTHL
125A     72                         MOV     M,D
125B     2C                         INR     L
125C     73                         MOV     M,E
125D     E5                         PUSH    H
125E     21 61 21                   LXI     H,LEN
1261     35                         DCR     M
1262     2D                         DCR     L
1263     35                         DCR     M
1264     E1                         POP     H
1265     C2 4B 12                   JNZ     CVB1
1268     E1               CVB2:     POP     H                ;THE PROCESS IS COMPLET
1269     F1                         POP     PSW              ;  RESULT IN DE
126A     C1                         POP     B
126B     21 61 21                   LXI     H,LEN
126E     4E                         MOV     C,M
126F     E1                         POP     H
1270     C9                         RET
                          ;
                          ; ROUTINE TO EVALUATE LENGTH OF ASCII NUMERIC
                          ; CHAR STRING: PASSED ADD OF FIRST CHAR IN HL REG.
                          ; RETURNS LENGTH IN REG A.
                          ;
1271     C5               LENGTH:   PUSH    B                ;SAVE REGISTERS
1272     E5                         PUSH    H
1273     06 00                      MVI     B,0              ;INITIALIZE REG B
1275     CD 14 12         NLE1:     CALL    NUMB             ;IS HL(M) AN ASCII NUMB
1278     D2 84 12                   JNC     NLE2             ;NO DONE- REG B CONTAIN
127B     23                         INX     H                ;YEP- GET NEXT ONE
127C     04                         INR     B
127D     0D                         DCR     C                ;CHECK FOR END OF LINE
127E     CA 84 12                   JZ      NLE2
1281     C3 75 12                   JMP     NLE1             ;LOOP ON PROCESS
1284     78               NLE2:     MOV     A,B              ;MOVE LENGTH INTO A
1285     E1                         POP     H
1286     C1                         POP     B
1287     C9                         RET
                          ;
                          ; ROUTINE TO LOCATE SOURCE LINE IN MEM. PASSED BIN VALUE
                          ; OF LINE NUMBER IN DE (LOW,HIGH) REG. RETURNS ADDRESS OF
                          ; SOURCE LINE IN HL REGS.(HIGH,LOW). CY SET=> NOT FOUND.
                          ;
1288     2A 49 21         NSRCH:    LHLD    STLINE           ;GET FIRST LINE
128B     CD A3 12         L2:       CALL    CHK1             ;IS IT -1 (END OF LIST)
128E     D8                         RC                       ;YES- NO SOURCE-RETURN
128F     46                         MOV     B,M              ;MOVE LINE NUMB. INTO B
1290     23                         INX     H
1291     4E                         MOV     C,M
1292     CD F1 0F                   CALL    DCOMP            ;HOW DO THEY COMPARE?
1295     CA A0 12                   JZ      FOUND
1298     23                         INX     H                ;BUMP HL TO GET POINTER
1299     7E                         MOV     A,M
129A     23                         INX     H
129B     66                         MOV     H,M              ;MOVE TO NEXT LINE AND
129C     6F                         MOV     L,A
129D     C3 8B 12                   JMP     L2
12A0     2B               FOUND:    DCX     H                ;MATCH--POINT TO FIRST
12A1     B7                         ORA     A
12A2     C9                         RET
                          ;
                          ; ROUTINE TO COMPARE CONTENTS OF HL TO 177777Q.
                          ; RETURNS CY=1 IF YES: CY=0 IF NO.
                          ;
12A3     C5               CHK1:     PUSH    B                ;SAVE REGISTERS
12A4     E5                         PUSH    H
12A5     06 00                      MVI     B,0              ;MOVE 1 INTO BC
12A7     0E 01                      MVI     C,1
12A9     09                         DAD     B                ;DOUBLE ADD TO SET STAT
12AA     E1                         POP     H
12AB     C1                         POP     B
12AC     C9                         RET
                          ;
                          ; ROUTINE TO PAD OUTPUT BUFFER WITH CONTENTS OF REG A.
                          ; REG B CONTAINS NUMBER OF CHAR TO PAD.
                          ;
12AD     C5               PAD:      PUSH    B                ;SAVE REGISTERS
12AE     D5                         PUSH    D
12AF     E5                         PUSH    H
12B0     21 00 21                   LXI     H,OBUFF          ;GET OBUFF ADDRESS
12B3     4D                         MOV     C,L              ;POINT TO PROPER ENTRY
12B4     6E                         MOV     L,M
12B5     57                         MOV     D,A
12B6     3E 49                      MVI     A,73             ;CHECK FOR LINE LENGTH`
12B8     BD               P1:       CMP     L
12B9     C2 C2 12                   JNZ     P2               ;ROOM LEFT--GO FILL
12BC     69                         MOV     L,C              ;OBUFF IS FULL- AUTO PR
12BD     77                         MOV     M,A
12BE     CD D0 12                   CALL    WRIT
12C1     2C                         INR     L                ;RE-INITIALIZE L
12C2     72               P2:       MOV     M,D
12C3     2C                         INR     L
12C4     05                         DCR     B                ;DECREMENT NUMB. CHAR.
12C5     C2 B8 12                   JNZ     P1               ;GO TO P1 IF ANY LEFT
12C8     7A                         MOV     A,D              ;RESTORE PROPER STATUS
12C9     45                         MOV     B,L
12CA     69                         MOV     L,C
12CB     70                         MOV     M,B
12CC     E1                         POP     H
12CD     D1                         POP     D
12CE     C1                         POP     B
12CF     C9                         RET
                          ;
                          ; ROUTINE TO DUMP OUTPUT BUFFER TO TTY.
                          ;
12D0     16 00            WRIT:     MVI     D,0
12D2     F5               WRIT1:    PUSH    PSW
12D3     E5                         PUSH    H
12D4     C5                         PUSH    B
12D5     21 00 21                   LXI     H,OBUFF          ;GET OBUFF ADDR.
12D8     E5                         PUSH    H
12D9     4E                         MOV     C,M              ;GET NUMB. OF CHAR.
12DA     0D                         DCR     C                ;IS OBUFF EMPTY
12DB     CA F0 12                   JZ      W2
12DE     2C                         INR     L
12DF     7E               W1:       MOV     A,M              ;MOVE CHAR.
12E0     F7                         RST     SEND             ;PRINT VIA ODT
12E1     2C                         INR     L                ;MOVE TO NEXT CHAR.
12E2     0D                         DCR     C
12E3     C2 DF 12                   JNZ     W1
12E6     15                         DCR     D
12E7     CA F0 12                   JZ      W2
12EA     3E 8D                      MVI     A,215Q           ;PUT OUT CR. LF.
12EC     F7                         RST     SEND             ;PRINT VIA ODT
12ED     3E 8A                      MVI     A,212Q
12EF     F7                         RST     SEND             ;PRINT VIA ODT
12F0     E1               W2:       POP     H
12F1     36 01                      MVI     M,1
12F3     C1                         POP     B
12F4     E1                         POP     H
12F5     F1                         POP     PSW
12F6     C9                         RET
                          ;
                          ; ROUTINE TO LOCATE COMMANDS, KEY WORDS, OPERATORS,
                          ; AND FUNCTIONS.  HL CONTAINS ADD OF FIRST CHAR.:
                          ; REG C CONTAINS LENGTH OF LINE: RETURNS SYMBOL NUMBER
                          ; IF FOUND IN REG A, 377Q IN A IF NOT FOUND.
                          ; ON ENTRY REG. A CONTAINS TYPE OF ENTRY SOUGHT:
                          ;
                          ;  0 FOR COMMAND
                          ;  1 FOR KEYWORD
                          ;  2 FOR OPERATOR AND DELIMITER
                          ;  3 FOR FUNCTION
                          ;
12F7     D5               SYMSRT:   PUSH    D
12F8     C5                         PUSH    B
12F9     E5                         PUSH    H
12FA     E5                         PUSH    H
12FB     21 61 21                   LXI     H,LEN            ;SAVE C IN LEN
12FE     71                         MOV     M,C
12FF     21 4B 13                   LXI     H,KDATA
1302     1E 00                      MVI     E,0
1304     85                         ADD     L
1305     6F                         MOV     L,A
1306     6E                         MOV     L,M
1307     4E               S2:       MOV     C,M
1308     2C               S3:       INR     L                ;LEN IN C AND BUMP L
1309     46                         MOV     B,M              ;FIRST CHAR. IN B
130A     E3                         XTHL                     ;FIRST CHAR OF USER STR
130B     7E                         MOV     A,M              ;AND COMPARE
130C     B8                         CMP     B
130D     C2 23 13                   JNZ     S4               ;NO MATCH-TRY AGAIN
1310     0D                         DCR     C                ;SET UP FOR NEXT CHAR
1311     CA 3F 13                   JZ      S5               ;IF DONE EXIT
1314     E5                         PUSH    H
1315     21 61 21                   LXI     H,LEN            ;DECREMENT LINE LENGTH
1318     35                         DCR     M
1319     E1                         POP     H
131A     CA 22 13                   JZ      S4A              ;IF 0 THEN MOVE ON
131D     23                         INX     H                ;REPEAT PROCESS
131E     E3                         XTHL
131F     C3 08 13                   JMP     S3
1322     0C               S4A:      INR     C
1323     E1               S4:       POP     H                ;GET ADDR. OF TABLE
1324     79                         MOV     A,C
1325     85                         ADD     L
1326     54                         MOV     D,H
1327     E1                         POP     H
1328     C1                         POP     B
1329     C5                         PUSH    B
132A     E5                         PUSH    H
132B     E5                         PUSH    H
132C     21 61 21                   LXI     H,LEN            ;SET UP PROPER ENTRY
132F     71                         MOV     M,C              ;POINT TO PROPER ENTRY
1330     6F                         MOV     L,A
1331     62                         MOV     H,D
1332     7E                         MOV     A,M
1333     1C                         INR     E
1334     4F                         MOV     C,A
1335     3C                         INR     A
1336     C2 08 13                   JNZ     S3
1339     21 61 21                   LXI     H,LEN
133C     34                         INR     M
133D     1E FF                      MVI     E,377Q           ;E=-1 IF NO MATCH
133F     7B               S5:       MOV     A,E              ;MOVE SYMBOL NUMBER INT
1340     21 61 21                   LXI     H,LEN
1343     5E                         MOV     E,M
1344     1D                         DCR     E
1345     E1                         POP     H
1346     E1                         POP     H
1347     C1                         POP     B
1348     4B                         MOV     C,E              ;MOVE NUMBER OF CHAR. L
1349     D1                         POP     D
134A     C9                         RET
                          ;
                          ;****************************************************
                          ;THE CODE FROM HERE TO THE NEXT LINE OF *'S MUST BE ON ON
                          ;
134B     4F               KDATA:    DB      KDAT1 AND 377Q
134C     64                         DB      KDAT2 AND 377Q
134D     9E                         DB      KDAT3 AND 377Q
134E     C0                         DB      KDAT4 AND 377Q
134F     03 D2 D5 CE      KDAT1:    DB      3,322Q,325Q,316Q    ;RUN
1353     03 D0 CC D3                DB      3,320Q,314Q,323Q    ;PLS
1357     03 CC C9 D3                DB      3,314Q,311Q,323Q    ;LIS
135B     03 D3 C3 D2                DB      3,323Q,303Q,322Q    ;SCR
135F     03 D0 D4 C1                DB      3,320Q,324Q,301Q    ;PTA
1363     FF                         DB      377Q
1364     03 CC C5 D4      KDAT2:    DB      3,314Q,305Q,324Q    ;LET
1368     03 D0 D2 C9                DB      3,320Q,322Q,311Q    ;PRI
136C     03 D2 C5 CD                DB      3,322Q,305Q,315Q    ;REM
1370     03 D3 D4 CF                DB      3,323Q,324Q,317Q    ;STO
1374     03 C5 CE C4                DB      3,305Q,316Q,304Q    ;END
1378     03 C7 CF D4                DB      3,307Q,317Q,3240    ;GOT
137C     02 C9 C6                   DB      2,311Q,306Q         ;IF
137F     03 C9 CE D0                DB      3,311Q,316Q,320Q    ;INP
1383     03 C4 C9 CD                DB      3,304Q,311Q,315Q    ;DIM
1387     03 C3                      DB      3,'C'+200Q          ;CAL
1389     C1                         DB      'A'+200Q
138A     CC                         DB      'L'+200Q
138B     04 C7                      DB      4,'G'+200Q          ;GOSU
138D     CF                         DB      'O'+200Q
138E     D3                         DB      'S'+200Q
138F     D5                         DB      'U'+200Q
1390     03 D2                      DB      3,'R'+200Q          ;RET
1392     C5                         DB      'E'+200Q
1393     D4                         DB      'T'+200Q
1394     03 C6                      DB      3,'F' OR 200Q       ;FOR
```

```
1396    CF                          DB      'O' OR 200Q
1397    D2                          DB      'R' OR 200Q
1398    04 CE                       DB      4,'N' OR 200Q       ;NEXT
139A    C5                          DB      'E' OR 200Q
139B    D8                          DB      'X' OR 200Q
139C    D4                          DB      'T' OR 200Q
139D    FF                          DB      377Q
                        ;
                        ;DELIMITERS HAVE FOLLOWING VALUES:
                        ;
                        ;           <       0
                        ;           >       1
                        ;           ,       2
                        ;           =       3
                        ;           )       4
                        ;           ;       5
                        ;           THEN    6
                        ;           TO      7
                        ;           STEP    8
                        ;           *       9
                        ;           /       10
                        ;
                        ;           +       11
                        ;           -       12
                        ;
139E    01 BC 01 BE     KDAT3:      DB      1,274Q,1,276Q       ;'<','>'
13A2    01 AC 01 BD                 DB      1,254Q,1,275Q       ;',','='
13A6    01 A9                       DB      1,251Q              ;')'
13A8    01 BB                       DB      1,';'+200Q          ;';'
13AA    04                          DB      4                   ;THEN
13AB    D4                          DB      'T' OR 200Q
13AC    C8                          DB      'H' OR 200Q
13AD    C5                          DB      'E' OR 200Q
13AE    CE                          DB      'N' OR 200Q
13AF    02                          DB      2                   ;TO
13B0    D4                          DB      'T' OR 200Q
13B1    CF                          DB      'O' OR 200Q
13B2    04                          DB      4                   ;STEP
13B3    D3                          DB      'S' OR 200Q
13B4    D4                          DB      'T' OR 200Q
13B5    C5                          DB      'E' OR 200Q
13B6    D0                          DB      'P' OR 200Q
13B7    01 AA                       DB      1,'*'+200Q          ;'*'
13B9    01 AF 01 AB                 DB      1,257Q,1,253Q       ;'/','+'
13BD    01 AD                       DB      1,255Q              ;'-'
13BF    FF                          DB      377Q
13C0    03 C7 C5 D4     KDAT4:      DB      3,307Q,305Q,324Q ;GET
13C4    03 D0 D5 D4                 DB      3,320Q,325Q,324Q ;PUT
13C8    FF                          DB      377Q
                        ;
                        ;*************************************************
                        ; ROUTINE TO INPUT SOURCE LINE FROM TTY. PASSED ADD
                        ; OF FIRST CHAR IN HL. RETURNS LENGTH OF LINE IN REG A
                        ;
13C9    F5              TTYIN:      PUSH    H
13CA    06 00                       MVI     B,0
13CC    CD F7 0F        TIN1:       CALL    CHAR2
13CF    FE 99                       CPI     231Q                ;CNTRL Y?
13D1    CA 01 14                    JZ      TIN5
13D4    FE FF                       CPI     377Q                ;RUBOUT?
13D6    CA F6 13                    JZ      TIN2
13D9    FE DF                       CPI     337Q                ;BACK ARROW (RUBOUT)?
13DB    CA F9 13                    JZ      TIN2+3
13DE    FE 8A                       CPI     212Q                ;LF?
13E0    CA CC 13                    JZ      TIN1
13E3    FE 8D                       CPI     215Q                ;CR
13E5    CA 08 14                    JZ      TIN4
13F8    FE 8C                       CPI     214Q                ;FORM FEED?
13EA    CA CC 13                    JZ      TIN1                ;IGNORE
13ED    77                          MOV     M,A
13EE    23                          INX     H
13EF    04                          INR     B
13F0    CD 14 15                    CALL    MEMFJL

13F3    C3 CC 13                    JMP     TIN1
13F6    3E DF           TIN2:       MVI     A,337Q
13F8    F7                          RST     SEND                ;PRINT VIA ODT
13F9    2B                          DCX     H
13FA    05                          DCR     B
13FB    F2 CC 13                    JP      TIN1
13FE    E1                          POP     H
13FF    AF                          XRA     A                   ;ZERO A
1400    C9                          RET
1401    3E DC           TIN5:       MVI     A,334Q
1403    F7                          RST     SEND                ;PRINT VIA ODT
1404    3E 00           TIN5A:      MVI     A,0
1406    F1                          POP     H
1407    C9                          RET
1408    3E 8A           TIN4:       MVI     A,212Q
140A    F7                          RST     SEND                ;PRINT VIA ODT
140B    0E 00           TIN4A:      MVI     C,0
140D    E1                          POP     H
140E    78                          MOV     A,B
140F    B9                          CMP     C
1410    C8                          RZ
                        ;
                        ; ROUTINE TO REMOVE BLANKS FROM SOURCE UNLESS ENCLOSED IN
                        ;
1411    D5                          PUSH    D                   ;SAVE REG'S
1412    F5                          PUSH    H
1413    E5                          PUSH    H
1414    1E A2                       MVI     E,'"'+200Q          ;INIT E FOR COMPARES
1416    16 00                       MVI     D,0                 ;D=1=>WITHIN QUOTES, LE
1418    AF              PK1:        XRA     A                   ;CLEAR A
1419    BA                          CMP     D                   ;CHECK INPUT MODE
141A    7E                          MOV     A,M                 ;GET CHAR
141B    C2 2B 14                    JNZ     QSTRG               ;WITHIN QUOTE STRING
141E    BB                          CMP     E                   ;IS IT 1ST "?
141F    C2 26 14                    JNZ     $+7                 ;NO - PROCEED
1422    14                          INR     D                   ;YES, SET FLAG
1423    C3 30 14                    JMP     QSTR1               ;CONTINUE
1426    FE A0                       CPI     240Q                ;IS IT A SPACE?
1428    CA 35 14                    JZ      PK2                 ;YES - LEAVE OUT
142B    BB              QSTRG:      CMP     E                   ;2ND "?
142C    C2 30 14                    JNZ     $+4                 ;NO - CONTINUE
142F    15                          DCR     D                   ;RESET FLAG
1430    E3              QSTR1:      XTHL                        ;GET DESTINATION ADDRES
1431    77                          MOV     M,A                 ;SAVE
1432    23                          INX     H                   ;BUMP PNTR.
1433    E3                          XTHL                        ;GET SOURCE ADD.
1434    0C                          INR     C                   ;BUMP CHAR. CNT
1435    23              PK2:        INX     H                   ;BUMP PNTR.
1436    05                          DCR     B                   ;DCR INPUT LINE CHAR CN
1437    C2 18 14                    JNZ     PK1                 ;MORE - GO AGAIN
143A    79                          MOV     A,C                 ;CHAR CNT TO A

143B    E1                          POP     H                   ;RESTORE REG'S, RETURN
143C    E1                          POP     H
143D    D1                          POP     D
143E    C9                          RET
                        ;
                        ; ROUTINES TO  PAD MESSAGES TO  OUTPUT BUFFER.
                        ; FOR12 PADS 'UNDERFLOW'
                        ; FOR11 PADS 'OVERFLOW'
                        ; FOR10 PADS 'ZERODIVIDE'
                        ; FORM9 PADS 'INPUT ERROR, TRY AGAIN'
                        ; FORM8 PADS 'MEMORY FULL'
                        ; FORM7 PADS 'WHAT?'
                        ; FORM4 PADS 'IN LINE'
                        ; FORM3 PADS 'ERROR'
                        ; FORM2 PADS 'TURN ON PUNCH'
                        ; FORM1 PADS 'READY'
                        ; FORM5 PADS SOURCE LINE, PASSED ADDRESS OF
                        ; LENGTH OF LINE IN HL REGS.
                        ; FORM6 PADS CHAR STRING, PASSED ADD OF FIRST CHAR IN
                        ; HL, LENGTH OF STRING IN REG C
                        ;
143F    2C              FOR12:      INR     L                   ;THE ENTRY POINT INCREM
1440    2C              FOR11:      INR     L                   ;PROPER DATA POINT
1441    2C              FOR10:      INR     L
1442    2C              FORM9:      INR     L
1443    2C              FORM8:      INR     L
1444    2C              FORM7:      INR     L
1445    2C              FORM4:      INR     L
1446    2C              FORM3:      INR     L
1447    2C              FORM2:      INR     L
1448    6F              FORM1:      MOV     L,M                 ;POINT TO PROPER BUFFER
1449    4E              FORM5:      MOV     C,M                 ;MOVE LENGTH INTO C
144A    79                          MOV     A,C                 ;AND STORE IN A
144B    FE 00                       CPI     0                   ;IS IT 0?
144D    C8                          RZ
144E    23              F1:         INX     H                   ;INCREMENT TO GET FIRST
144F    7E              FORM6:      MOV     A,M                 ;THE PAD LOOP
1450    06 01                       MVI     B,1
1452    CD AD 12                    CALL    PAD
1455    0D                          DCR     C
1456    C2 4E 14                    JNZ     F1
1459    C9                          RET
                        ;
                        ;*************************************************
                        ; THE CODE FROM HERE TO THE NEXT LINE OF *'S MUST BE ON O
                        ;
145A    64              ODATA:      DB      ODAT1 AND 377Q
145B    6A                          DB      ODAT2 AND 377Q
145C    78                          DB      ODAT3 AND 377Q
145D    81                          DB      ODAT4 AND 377Q
145E    8B                          DB      ODAT5 AND 377Q

145F    91                          DB      ODAT6 AND 377Q
1460    A0                          DB      ODAT7 AND 377Q
1461    B7                          DB      ODAT8 AND 377Q
1462    C2                          DB      ODAT9 AND 377Q
1463    CB                          DB      ODA10 AND 377Q
1464    05 52 45 41     ODAT1:      DB      5,'READY'
1468    44 59
146A    0D 54 55 52     ODAT2:      DB      13,'TURN ON PUNCH'
```

```
146E    4E 20 4F 4E
1472    20 50 55 4E
1476    43 48
1478    08 8D 8A 45     ODAT3:      DB      8,215Q,212Q,'ERROR '
147C    52 52 4F 52
1480    20
1481    09 20 49 4E     ODAT4:      DB      9,' IN LINE '
1485    20 4C 49 4E
1489    45 20
148B    05 57 48 41     ODAT5:      DB      5,'WHAT?'
148F    54 3F
1491    0E 4D 45 4D     ODAT6:      DB      14,'MEMORY FULL',215Q,212Q,'?'
1495    4F 52 59 20
1499    46 55 4C 4C
149D    8D 8A 3F
14A0    16 49 4E 50     ODAT7:      DB      22,'INPUT ERROR, TRY AGAIN'
14A4    55 54 20 45
14A8    52 52 4F 52
14AC    2C 20 54 52
14B0    59 20 41 47
14B4    41 49 4E
14B7    0A 49 4E 44     ODAT8:      DB      10,'INDEFINITE'
14BB    45 46 49 4E
14BF    49 54 45
14C2    08 4F 56 45     ODAT9:      DB      8,'OVERFLOW'
14C6    52 46 4C 4F
14CA    57
14CB    09 55 4E 44     ODA10:      DB      9,'UNDERFLOW'
14CF    45 52 46 4C
14D3    4F 57
                        ;
                        ;*************************************************
                        ; ROUTINE TO INPUT SOURCE LINE FROM HSR. PASSED ADD
                        ; OF FIRST CHAR IN HL. RETURNS LENGTH OF LINE IN REG A
                        ;
14D5    E5              HSRIN:      PUSH    H                   ;SAVE ADDR.
14D6    06 00                       MVI     B,0                 ;INIT. NUMB. OF CHAR. R
14D8    C3 DE 14                    JMP     PIN1A
14DB    CD 08 15        PIN1:       CALL    CHAR5               ;GET A CHAR.
14DE    FE 99           PIN1A:      CPI     231Q                ;CNTRL Y?
14E0    CA 04 14                    JZ      TIN5A
14E3    FE FF                       CPI     377Q                ;DEL?
14E5    CA 00 15                    JZ      PIN3

14E8    FE DF                       CPI     337Q                ;CNTR A?
14EA    CA 00 15                    JZ      PIN3
14ED    FE 8A                       CPI     212Q                ;CR?
14EF    CA 0B 14                    JZ      TIN4A
14F2    FE 8D                       CPI     215Q                ;CR?
14F4    CA DB 14                    JZ      PIN1
14F7    77                          MOV     M,A                 ;GOOD CHAR--STORE IT AW
14F8    23                          INX     H                   ;MOVE TO NEXT SLOT
14F9    04                          INR     B
14FA    CD 14 15                    CALL    MEMFUL              ;IS MEM. FULL?
14FD    C3 DB 14                    JMP     PIN1                ;LOOP ON PROCESS
1500    2B              PIN3:       DCX     H                   ;DELETE THE LAST CHAR.
1501    05                          DCR     B                   ;MUST INSURE THE LINE I
1502    F2 DB 14                    JP      PIN1
1505    F1                          POP     H
1506    AF                          XRA     A                   ;ZERO A
1507    C9                          RET
                        ;
                        ; ROUTINE TO INPUT CHAR FROM HSR
                        ;
1508    C5              CHAR5:      PUSH    B
1509    06 04                       MVI     B,4                 ;THE INPUT REQUEST
150B    D3 05                       OUT     5
150D    CD F4 0F                    CALL    MCHK                ;GO GET CHAR.
1510    DB 05                       IN      5                   ;THE TERMINATE REQUEST
1512    C1                          POP     B
1513    C9                          RET
                        ;
                        ; ROUTINE TO INSURE SOURCE DOES NOT OVERFLOW MEM SPACE
                        ; COMPARES CURENT MEM ADDRESS TO SP.
                        ;
1514    C5              MEMFUL:     PUSH    B                   ;SAVE REG.B,D,H
1515    D5                          PUSH    D
1516    E5                          PUSH    H
1517    3E 32                       MVI     A,50                ;MOVE DEC 50 INTO B TO
1519    85                          ADD     L                   ;INSURE MARGINE OF STAC
151A    47                          MOV     B,A
151B    3E 00                       MVI     A,0
151D    8C                          ADC     H
151E    4F                          MOV     C,A                 ;ESTABLISH MEM+50 IN BC
151F    21 00 00                    LXI     H,0
1522    39                          DAD     SP                  ;MOVE SP INTO HL
1523    55                          MOV     D,L
1524    5C                          MOV     E,H                 ;SET UP FOR DCOMP
1525    CD F1 0F                    CALL    DCOMP
1528    E1                          POP     H
1529    D1                          POP     D
152A    C1                          POP     B
152B    D0                          RNC                         ;CY=0 THEN OK TO CONTIN
152C    21 5A 14                    LXI     H,ODATA
152F    CD 43 14                    CALL    FORM8               ;MEMORY FULL--PAD
1532    CD D0 12                    CALL    WRIT                ;MESSAGE AND WRITE
1535    CD F7 0F                    CALL    CHAR2               ;GET USER RESPONCE
1538    CD AD 12                    CALL    PAD                 ;MUST BE 1,2 OR 3 FOR C
153B    CD D0 12                    CALL    WRIT
153E    DE B0                       SBI     260Q
1540    FE 04                       CPI     4
1542    CC 9E 11                    CZ      WHAT                ;IMPROPER RESPONCE
1545    31 F5 27                    LXI     SP,MEMEND-10        ;RESET SP
1548    0E 01                       MVI     C,1
154A    C3 47 10                    JMP     M4A                 ;LOOP ON CONTROL SECTIO
                        ;
                        ; ROUTINE TO EVALUATE BOUNDS FOR LIST AND PLIST
                        ; COMMANDS. RETURNS PLINE AS FIRST LINE, KLINE
                        ; AS LAST LINE TO BE LISTED.
                        ;
154D    2A 4B 21        BOUND:      LHLD    NLINE               ;GET ADDR. OF NEW LINE
1550    3E 09                       MVI     A,9
1552    85                          ADD     L                   ;INCREMENT 9 TO GET TO
1553    6F                          MOV     L,A
1554    3E 00                       MVI     A,0
1556    8C                          ADC     H
1557    67                          MOV     H,A                 ;THE ADDR. IS NOW THERE
1558    E5                          PUSH    H
1559    CD 14 12                    CALL    NUMB                ;SEE IF THERE IS A NUMB
155C    D4 9E 11                    CNC     WHAT                ;IT IS NOT
155F    CD 2A 12                    CALL    CVB                 ;CONVERT NUMBER TO BINA
1562    F5                          PUSH    PSW
1563    C5                          PUSH    B
1564    CD A3 15                    CALL    BND2                ;GO FIND FIRST LINE
1567    C1                          POP     B
1568    2B                          DCX     H
1569    22 59 21                    SHLD    PLINE               ;STORE IT WAY AS FIRST
156C    F1              BND1:       POP     PSW
156D    E1                          POP     H
156E    3C                          INR     A
156F    85                          ADD     L                   ;SET UP TO GET UPPER BO
1570    6F                          MOV     L,A
1571    3E 00                       MVI     A,0
1573    8C                          ADC     H
1574    67                          MOV     H,A                 ;UPPER BOUND ADDR IN HL
1575    3E 00                       MVI     A,0
1577    B9                          CMP     C                   ;IS THAT END OF LINE?
1578    C8                          RZ
1579    0D                          DCR     C
157A    CD 14 12                    CALL    NUMB                ;IS NEXT CHAR NUMB?
157D    D4 9E 11                    CNC     WHAT                ;NOT NUMBER --WHZTS UP
1580    D5                          PUSH    D
1581    CD 2A 12                    CALL    CVB                 ;CONVERT TO BINARY
1584    D5                          PUSH    D
1585    C5                          PUSH    B
1586    CD A3 15                    CALL    BND2                ;GO FIND SECOND BOUND

1589    C1                          POP     B
158A    23                          INX     H
158B    56                          MOV     D,M
158C    23                          INX     H
158D    5E                          MOV     E,M                 ;SETUP TO STORE IT IN K
158E    EB                          XCHG
158F    22 52 21                    SHLD    KLINE
1592    D1                          POP     D
1593    E1                          POP     H
1594    79                          MOV     A,C
1595    FE 00                       CPI     0                   ;ARE WE AT END OF LINE
1597    C2 9E 11                    JNZ     WHAT
159A    44                          MOV     B,H
159B    4D                          MOV     C,L
159C    CD F1 0F                    CALL    DCOMP
159F    D0                          RNC
15A0    C3 9E 11                    JMP     WHAT
15A3    2A 49 21        BND2:       LHLD    STLINE              ;SET UP TO SEARCH FROM
15A6    46              BND3:       MOV     B,M
15A7    23                          INX     H
15A8    4E                          MOV     C,M
15A9    CD F1 0F                    CALL    DCOMP               ;COMPARE BINARY OF REQU
15AC    D8                          RC                          ;TO CURRENT LINE
15AD    C8                          RZ
15AE    E5                          PUSH    H
15AF    23                          INX     H
15B0    7E                          MOV     A,M
15B1    23                          INX     H
15B2    66                          MOV     H,M
15B3    6F                          MOV     L,A
15B4    CD A3 12                    CALL    CHK1                ;MOVE TO NEXT LINE
15B7    C1                          POP     B                   ;IS THAT ALL?
15B8    D2 A6 15                    JNC     BND3                ;NOPE--LOOP ON PROCESS
15BB    C5                          PUSH    B
15BC    E1                          POP     H
15BD    C9                          RET
```

```
                     ; ROUTINE TO OUTPUT ERROR MSG. TO USER.
                     ; REG A CONTAINS BCD ERROR NUMBER, HL
                     ; LOADED WITH VALUE OF KLINE.
15BE   21 13 10      ERROR:   LXI    H,M1A         ;RETURN ADDRESS
15C1   E5                     PUSH   H             ;PUT ON STACK
15C2   21 5A 14               LXI    H,ODATA       ;OUTPUT BUFFER DATA TAB
15C5   E5                     PUSH   H
15C6   57                     MOV    D,A           ;SAVE ERROR NUMB. IN D
15C7   CD 46 14               CALL   FORM3         ;PAD 'ERROR '
15CA   06 01                  MVI    B,1           ;INIT FOR PADS
15CC   48                     MOV    C,B           ;INIT AS CNTR.
15CD   7A                     MOV    A,D           ;GET ERROR NUMB.
15CE   07                     RLC                  ;ROTATE HIGH 4 BITS TO

15CF   07                     RLC
15D0   07                     RLC
15D1   07                     RLC
15D2   E6 0F         ERRR1:   ANI    17Q           ;MASK
15D4   C6 B0                  ADI    260Q          ;CONVERT TO ASCII
15D6   CD AD 12               CALL   PAD           ;PAD IT
15D9   7A                     MOV    A,D           ;GET ERROR NUMB.
15DA   0D                     DCR    C             ;ANOTHER PASS?
15DB   F2 D2 15               JP     ERRR1         ;YES
15DE   E1                     POP    H             ;NO-CONTINUE
15DF   CD 45 14      ERLN:    CALL   FORM4         ;PAD 'IN LINE'
15E2   2A 52 21               LHLD   KLINE
15E5   23                     INX    H
15E6   23                     INX    H
15E7   23                     INX    H
15E8   23                     INX    H
15E9   4E                     MOV    C,M
15EA   23                     INX    H
15EB   CD 71 12               CALL   LENGTH
15EE   4F                     MOV    C,A
15FF   CD 4F 14               CALL   FORM6         ;PAD LINE #
15F2   CD D0 12               CALL   WRIT          ;WRITE MESSAGE
15F5   C9                     RET
                     ;
                     ; THIS ROUTINE INCREMENTS H AND L AND
                     ; DECR. C(CHARS IN LINE) SHOULD C RESULT
                     ; IN 0 THEN THE ERROR CORRES. TO ENTRY PNT.
                     ; IS GIVEN
                     ;
15F6   3E 07         ICP7:    MVI    A,7
15F8   C3 07 16               JMP    INCPT
15FB   3E 08         ICP8:    MVI    A,8
15FD   C3 07 16               JMP    INCPT
1600   3E 04         ICP4:    MVI    A,4
1602   C3 07 16               JMP    INCPT
1605   3E 02         ICP2:    MVI    A,2
1607   23            INCPT:   INX    H
1608   0D                     DCR    C
1609   C0                     RNZ
160A   C3 BE 15               JMP    ERROR
                     ;
                     ; FSYM FINDS SYMBOLS IN TABLE
                     ; B,C CONTAIN SYMBOL
                     ; RET WITH B,C,D,E SAME
                     ; H AND L PNT TO VALUE (1ST BYTE)
                     ; CY=1   => FOUND
                     ; CY=0   AND A SCALAR VAR. => INSERTED
                     ;     AND SET TO 0
                     ; CY=0   AND AN ARRAY  => NO ACTION,
                     ;     H AND L PNT TO LAST ENTRY IN SYMBOL TABLE
                     ;
160D   D5            FSYM:    PUSH   D
160E   AF                     XRA    A
160F   B0                     ORA    B             ;SET CARRY IF NOT
1610   CA 14 16               JZ     AR            ;AN ARRAY AND SAVE
1613   3F                     CMC
1614   F5            AR:      PUSH   PSW
1615   2A 59 21               LHLD   NXTSP         ;GET NEXT AVAILABLE
1618   C5                     PUSH   B             ;SPACE PNTR.
1619   44                     MOV    B,H
161A   4D                     MOV    C,L           ;CHECK TO SEE
161B   2A 4B 21               LHLD   STSPAC        ;IF SYMBOL TABLE
161E   54                     MOV    D,H           ;EMPTY
161F   5D                     MOV    E,L
1620   CD F1 0F               CALL   DCOMP         ;DOUBLE BYTE COMPARE
1623   C1                     POP    B             ;GET VAR. BACK
1624   CA 44 16               JZ     NOSYM
1627   CD A3 12      LUKON:   CALL   CHK1          ;CHECK FOR END
162A   DA 46 16               JC     NOENT
162D   54                     MOV    D,H           ;SAVE OLD PNTR
162E   5D                     MOV    E,L
162F   78                     MOV    A,B
1630   BE                     CMP    M             ;DO VARIABLES MATCH
1631   C2 3B 16               JNZ    NOMAT
1634   23                     INX    H
1635   79                     MOV    A,C
1636   BE                     CMP    M
1637   CA 71 16               JZ     ENTRY
163A   2B                     DCX    H
163B   23            NOMAT:   INX    H             ;NO MATCH GET NEW PNT.
163C   23                     INX    H
163D   7E                     MOV    A,M
163E   23                     INX    H
163F   66                     MOV    H,M
1640   6F                     MOV    L,A
1641   C3 27 16               JMP    LUKON
                     ; ARRIVE HERE IF SYMBOL TABLE IS EMPTY
1644   1B            NOSYM:   DCX    D             ; =STSPAC-2 SO STPNT WO
1645   1B                     DCX    D
                     ; ARRIVE HERE WHEN NO ENTRY FOUND
1646   2A 59 21      NOENT:   LHLD   NXTSP         ;ADD. OF FREE MEMORY
1649   EB                     XCHG                 ;TO DE, HL HAVE LAST SY
164A   F1                     POP    PSW           ;ARRAY?
164B   D2 76 16               JNC    FBAC          ;YES, RETURN
164E   CD 8B 11               CALL   STPNT         ;UPDATE PNTR
1651   EB                     XCHG                 ;NXTSP TO HL
1652   70                     MOV    M,B           ;STORE VAR.
1653   23                     INX    H
1654   71                     MOV    M,C
1655   23                     INX    H
1656   E5                     PUSH   H
1657   23                     INX    H             ;STORE NXTSP+8 IN NXTSP

1658   23                     INX    H
1659   23                     INX    H
165A   23                     INX    H
165B   23                     INX    H
165C   23                     INX    H
165D   22 59 21               SHLD   NXTSP
1660   CD 14 15               CALL   MEMFUL        ;MEMORY FULL?
1663   E1                     POP    H             ;SET FWD PNT. TO -1
1664   36 FF                  MVI    M,377Q
1666   23                     INX    H
1667   36 FF                  MVI    M,377Q
1669   23                     INX    H             ;INIT TO FLT. PNT. 0
166A   CD CD 0F               CALL   ZROL
166D   B7                     ORA    A             ;CLEAR CY
166E   C3 76 16               JMP    FBAC          ;RESET CARRY AND RETURN
1671   F1            ENTRY:   POP    PSW           ;VAR FOUND
1672   23                     INX    H             ;MOVE PNT. TO FIRST BYT
1673   23                     INX    H             ;OF FLT. PNT. NO.
1674   23                     INX    H
1675   37                     STC                  ;SET CY AND RET.
1676   D1            FBAC:    POP    D             ;RESTORE D
1677   C9                     RET
                     ;
                     ;
                     ; RUN - THE INTERP.
                     ;
                     ;
                     ; INIT. NXTSP
                     ;
1678   2A 4B 21      RUN:     LHLD   STSPAC
167B   22 59 21               SHLD   NXTSP
167E   21 AA 21               LXI    H,BOTNS       ;INIT SP FOR NESTING ST
1681   22 94 21               SHLD   NEST
1684   21 13 10               LXI    H,M1A         ;PRECAUTION, IN CASE RE
1687   E5                     PUSH   H             ;EXECUTED BEFORE A GOSU
1688   E5                     PUSH   H
1689   2A 49 21               LHLD   STLINE        ;START OF SOURCE
168C   CD 08 12      ILOOP:   CALL   QUITT         ;CHECK FOR INTERRUPTION
168F   CD A3 12               CALL   CHK1          ;HL=-1 => NO MORE SOURC
1692   D2 9A 16               JNC    SORCE
1695   3E 01                  MVI    A,1
1697   C3 BE 15               JMP    ERROR         ;ERROR 1, NO END STMT.
169A   22 52 21      SORCE:   SHLD   LPNT
169D   E5                     PUSH   H
169E   21 52 21               LXI    H,LPNT        ;DEFINE VALUES OF
16A1   CD EE 0F               CALL   PTVAL         ;KFPNT,KLEN
16A4   3A 58 21               LDA    KLEN          ;CHAR'S IN LINE TO C
16A7   4F                     MOV    C,A
16A8   0C                     INR    C
16A9   E1                     POP    H             ;MOVE PNTR. TO 1ST CHAR
16AA   23                     INX    H             ;IN SOURCE REC.

16AB   23                     INX    H
16AC   23                     INX    H
16AD   23                     INX    H
16AE   CD 05 16      L1:      CALL   ICP2          ;INCR. H,L DCR C
16B1   CD 22 12               CALL   ALPHA         ;FIND FIRST LETTER
16B4   D2 AE 16               JNC    L1
16B7   AF                     XRA    A
16B8   3C                     INR    A             ;LETTER FOUND
16B9   CD F7 12               CALL   SYMSRT        ;DETERMINE KEYWORD
16BC   FE FF                  CPI    377Q
16BE   C2 C6 16               JNZ    GKEY
16C1   3E 02                  MVI    A,2           ;BAD KEYWORD
16C3   C3 BE 15               JMP    ERROR
16C6   22 5B 21      GKEY:    SHLD   CPNT
16C9   21 D6 16               LXI    H,JTBL        ;LOAD JUMP TABLE PNTR.
16CC   87                     ADD    A             ;DOUBLE A
```

```
16CD   5F                     MOV    E,A
16CE   16 00                  MVI    D,0
16D0   19                     DAD    D             ;PNT. TO PROPER PROC.
16D1   7E                     MOV    A,M           ;ADD. IN JUMP TABLE
16D2   23                     INX    H             ;GET PROC. ADD.
16D3   66                     MOV    H,M
16D4   6F                     MOV    L,A
16D5   E9                     PCHL                 ;INDIRECT JUMP TO PROC.
16D6   DD 1B         JTBL:    DW     LET           ;JMP TABLE
16D8   95 1A                  DW     PRI
16DA   8C 1B                  DW     IEND          ;REM STMT. - NO ACTION
16DC   13 10                  DW     M1A           ;STOP STMT.-RETURN TO E
16DE   F2 16                  DW     ENDD
16E0   00 17                  DW     GOTO
16E2   7D 1C                  DW     IFRT
16E4   4C 1B                  DW     INPUT
16E6   20 17                  DW     DIM
16E8   5A 1D                  DW     CALLP
16EA   D0 1D                  DW     GOSUB
16EC   E5 1D                  DW     RETRN
16EE   E7 1D                  DW     FOR
16F0   D9 1E                  DW     NEXT
                     ;
                     ; END PROCESSOR
                     ;
16F2   2A 56 21      ENDD:    LHLD   KFPNT         ;CHECK TO SEE IF MORE
16F5   CD A3 12               CALL   CHK1          ;SOURCE AFTER END
16F8   DA 13 10               JC     M1A
16FB   3E 03                  MVI    A,3           ;MORE SOURCE ERROR 3
16FD   C3 BE 15               JMP    ERROR
                     ;
                     ; GO TO PROCESSOR
                     ;
1700   2A 5B 21      GOTO:    LHLD   CPNT          ;GOTO STMT. PROC.
1703   23            GSENT:   INX    H             ;INCREMENT PAST KEYWORD

1704   23                     INX    H
1705   23                     INX    H
1706   CD 00 16               CALL   ICP4          ;POSSIBLE ERROR 4
1709   CD 2A 12      GTRA:    CALL   CVB           ;GET DESTINATION
170C   B7                     ORA    A             ;MAKE SURE IT WAS OK
170D   C2 15 17               JNZ    OKN
1710   3E 04                  MVI    A,4
1712   C3 BE 15               JMP    ERROR
1715   CD 88 12      OKN:     CALL   NSRCH         ;GET NEXT LPNT
1718   D2 8C 16               JNC    ILOOP         ;MAKE SURE IT EXISTED
171B   3E 05                  MVI    A,5
171D   C3 BE 15               JMP    ERROR         ;NON-EXISTENT
                     ;
                     ; DIMENSION PROCESSOR
                     ;
1720   2A 5B 21      DIM:     LHLD   CPNT          ;DIM STMT. PROC.
1723   23                     INX    H             ;PNT TO FIRST VAR.
1724   23                     INX    H
1725   23                     INX    H
1726   CD 22 12      DLOOP:   CALL   ALPHA         ;CHECK IF IT IS A VAR.
1729   DA 31 17               JC     OKLET
172C   3E 06         ER6:     MVI    A,6           ;ERROR 6
172E   C3 BE 15               JMP    ERROR
1731   46            OKLET:   MOV    B,M
1732   CD F6 15               CALL   ICP7          ;INCR.CPNT
1735   3E A8                  MVI    A,250Q        ;CHECK FOR (
1737   BE                     CMP    M
1738   C2 2C 17               JNZ    ER6
173B   CD F6 15               CALL   ICP7          ;INCR. CPNT
173E   CD 2A 12               CALL   CVB           ;CONV. TO BIN NO.
1741   85                     ADD    L             ;UPDATE CPNT
1742   6F                     MOV    L,A           ;ED CONTAIN ARRAY LEN.
1743   3E 00                  MVI    A,0
1745   8C                     ADC    H             ;C CONT. NO. CHARS LEFT
1746   67                     MOV    H,A           ;IN LINE
1747   3E A9                  MVI    A,251Q        ;CHECK FOR )
1749   BE                     CMP    M
174A   C2 2C 17               JNZ    ER6
174D   E5                     PUSH   H
174E   C5                     PUSH   B             ;SAVE B,C,H,L
174F   48                     MOV    C,B           ;SET UP FOR CALL TO FSY
1750   06 00                  MVI    B,0
1752   CD 0D 16               CALL   FSYM
1755   D2 5F 17               JNC    NDOU
1758   C1                     POP    B
1759   E1                     POP    H
175A   3E 11                  MVI    A,11H         ;ERROR 11
175C   C3 BE 15               JMP    ERROR         ;DUPLICATE ARRAY DEF.
175F   D5            NDOU:    PUSH   D             ;SAVE DIM. LENGTH
1760   EB                     XCHG                 ;ADD. OF LAST SYM. TAB.
1761   2A 59 21               LHLD   NXTSP         ;GET ADD. OF AVAILABLE

1764   EB                     XCHG                 ;SET UP FOR CALL
1765   CD 8B 11               CALL   STPNT         ;STORE NEW PNTR
1768   EB                     XCHG                 ;NXTSP TO HL
1769   D1                     POP    D             ;RESTORE D
176A   36 00                  MVI    M,0           ;INSERT VAR IN SYMB. TA
176C   23                     INX    H
176D   71                     MOV    M,C
176E   23                     INX    H
176F   36 FF                  MVI    M,377Q        ;FPNT TO -1
1771   23                     INX    H
1772   36 FF                  MVI    M,377Q
1774   23                     INX    H             ;PNTS TO FIRST DATA
1775   7A                     MOV    A,D           ;GET ONE'S COMPLEMENT O
1776   2F                     CMA                  ;NUMBER OF ELEMENTS
1777   4F                     MOV    C,A           ;IN ARRAY TO B,C
1778   7B                     MOV    A,E
1779   2F                     CMA
177A   47                     MOV    B,A
177B   CD CD 0F      CONT:    CALL   ZROL          ;ZEROE OUT ELEMTS.
177E   23                     INX    H             ; OF ARRAY
177F   23                     INX    H
1780   23                     INX    H
1781   23                     INX    H
1782   03                     INX    B
1783   E5                     PUSH   H
1784   CD 14 15               CALL   MEMFUL        ;MEMORY FULL?
1787   60                     MOV    H,B
1788   69                     MOV    L,C
1789   CD A3 12               CALL   CHK1
178C   E1                     POP    H
178D   D2 7B 17               JNC    CONT
1790   22 59 21               SHLD   NXTSP         ;NEW VALUE OF NXTSP.
1793   C1                     POP    B             ;RESTORE REG'S
1794   E1                     POP    H
1795   23                     INX    H
1796   0D                     DCR    C             ;MORE ELEMTS IN LINE?
1797   CA 8C 1B               JZ     IEND
179A   0D                     DCR    C
179B   CA 2C 17               JZ     ER6
179E   3E AC                  MVI    A,254Q        ;NEXT ELEMENT A ,
17A0   BE                     CMP    M
17A1   23                     INX    H
17A2   CA 26 17               JZ     DLOOP
17A5   C3 2C 17               JMP    ER6
                     ;
                     ; ROUTINE TO COPY CONTENTS PNTED TO
                     ; BY DE TO LOCATION H,L
                     ;
17A8   F5            COPDH:   PUSH   PSW           ;SAVE REGISTERS
17A9   C5                     PUSH   B
17AA   D5                     PUSH   D

17AB   E5                     PUSH   H
17AC   06 04                  MVI    B,4           ;COUNT
17AE   1A            COPD1:   LDAX   D             ;GET FROM SOURCE
17AF   77                     MOV    M,A           ;PUT TO DESTINATION
17B0   13                     INX    D             ;BUMP PNTRS, CNT
17B1   23                     INX    H
17B2   05                     DCR    B
17B3   C2 AE 17               JNZ    COPD1
17B6   E1                     POP    H             ;RESTORE REGISTERS
17B7   D1                     POP    D
17B8   C1                     POP    B
17B9   F1                     POP    PSW
17BA   C9                     RET
                     ;
                     ; OUTR PADS OUTPUT FROM CONV INTO
                     ; OUTPUT BUFFER USING ROUTINE PAD
                     ; ALL REG'S MAINTAINED
                     ;
17BB   C5            OUTR:    PUSH   B             ;SAVE REG B
17BC   06 01                  MVI    B,1           ;PAD ONCE
17BE   CD AD 12               CALL   PAD           ;DO IT
17C1   C1                     POP    B             ;RESTORE B AND RET.
17C2   C9                     RET
                     ;
                     ; VALUE RETURNS IN D(H),E(L) PNTR.
                     ; TO THE VALUE OF A TOKEN
                     ; C,H,L ARE UPDATED
                     ; A,B ARE DESTROYED
                     ;
17C3   CD 6D 18      VALUE:   CALL   VAR           ;IS IT A VARIABLE?
17C6   D8                     RC                   ;YES - ALL DONE
17C7   3E 03                  MVI    A,3           ;NO CHEK IF A FUNC.
17C9   CD F7 12               CALL   SYMSRT
17CC   FE FF                  CPI    377Q
17CE   CA 3B 18               JZ     KONT          ;NOT A FUNCTION -
17D1   FE 01                  CPI    1             ;WAS IT PUT(--)?
17D3   C2 D9 17               JNZ    GET           ;NO - OK
17D6   C3 78 1C               JMP    ER10          ;ILLEGAL USE OF FUNCTIO
17D9   23            GET:     INX    H             ;OK, IT'S GET(--)
17DA   23                     INX    H             ;UPDATE H,L
17DB   23                     INX    H
17DC   79                     MOV    A,C           ;CHECK FOR PREMATURE EO
17DD   B7                     ORA    A
17DE   CA 22 1A               JZ     ER8
17E1   3E A8                  MVI    A,250Q        ;CHEK FOR (
17E3   BE                     CMP    M
17E4   C2 22 1A               JNZ    ER8
17E7   CD F8 15               CALL   ICP8          ;BUMP PNTR'S
17EA   CD C7 19               CALL   EVAL          ;GET PORT #
17ED  E5                 PUSH   H               ;SAVE REG H,L
17EE  21 7C 21           LXI    H,FREG1
17F1  CD A8 17           CALL   COPDH           ;COPY IT
17F4  EB                 XCHG
17F5  E1                 POP    H               ;RESTORE H,L
17F6  CD 15 19           CALL   FIX             ;FIX IT
17F9  13                 INX    D
17FA  13                 INX    D
17FB  13                 INX    D               ;GET LOWEST BYTE TO
17FC  1A                 LDAX   D               ;REG D
17FD  57                 MOV    D,A
17FE  79                 MOV    A,C
17FF  B7                 ORA    A               ;EOL?
1800  CA 22 1A           JZ     ER8
1803  3E A9              MVI    A,251Q          ;CHECK FOR )
1805  BE                 CMP    M
1806  C2 22 1A           JNZ    ER8
1809  23                 INX.   H               ;BUMP PNTR'S
180A  0D                 DCR    C
180B  E5                 PUSH   H               ;SAVE H,L,B,C
180C  C5                 PUSH   B               ;STORE PROGRAM SEGMENT
180D  01 77 21           LXI    B,GREG          ;IN RAM,START AT GREG
1810  21 36 18           LXI    H,RINST         ;ADD. OF INST'S
1813  1E 05              MVI    E,5             ;NUMB. OF BYTES
1815  7E        V1:      MOV    A,M             ;GET BYTE
1816  02                 STAX   B               ;STORE IN RAM
1817  23                 INX    H
1818  03                 INX    B
1819  1D                 DCR    E               ;BUMP PNTR'S,DCR CNT
181A  C2 15 18           JNZ    V1
181D  21 78 21           LXI    H,GREG+1        ;STORE PORT #
1820  72                 MOV    M,D             ;IN RAM
1821  C3 77 21           JMP    GREG            ;OK - TRANSFER
1824  21 79 21  HOME:    LXI    H,GREG+2        ;SET UP FOR FLOAT
1827  77                 MOV    M,A             ;STORE AWAY INPUT
1828  2B                 DCX    H
1829  AF                 XRA    A               ;ZERO OUT HIGHER BYTES
182A  77                 MOV    M,A             ;BUT CHAR. DOESN'T MATT
182B  2B                 DCX    H
182C  77                 MOV    M,A
182D  CD DC 0F           CALL   DFXL            ;FLOAT IT
1830  11 77 21           LXI    D,GREG          ;FIX D,E RESTORE C,H,L
1833  C1                 POP    B
1834  E1                 POP    H
1835  C9                 RET
1836  DB 00     RINST:   IN     0               ;RAM INSTRUCTIONS
1838  C3 24 18           JMP    HOME
183B  CD 14 12  KONT:    CALL   NUMB            ;NUMBER
183E  DA 47 18           JC     OKK
1841  3E AE              MVI    A,256Q          ;DEC. PNT.?
1843  BE                 CMP    M
1844  C2 22 1A           JNZ    ER8
1847  3E 01     OKK:     MVI    A,1             ;MODE=1, IE. INPUT FROM
1849  CD 53 18           CALL   RDKON           ;READ CONSTANT TO GREG
184C  DA 7C 1F           JC     ER9             ;IF ERROR THEN CY=1
184F  11 77 21           LXI    D,GREG          ;PNTS. TO CONSTANT
1852  C9                 RET
                ;
                ; THIS ROUTINE READS A CONSTANT INTO GREG FROM ASCII
                ; CHARACTERS POINTED TO BY HL AND C
                ; ENTER WITH A=0 => DATA FROM TTY
                ; ENTER WITH A=1 => DATA FROM SOURCE
                ; RETURN WITH CY=1 => ERROR IN CONVERSION
                ;
1853  32 85 21  RDKON:   STA    MODE            ;SAVE MODE FOR ROUT. IN
1856  22 86 21           SHLD   HLINP           ;SAVE HL FOR ROUT. INP
1859  79                 MOV    A,C
185A  32 84 21           STA    CREG            ;SAVE C FOR ROUT. INP
185D  21 77 21           LXI    H,GREG          ;WHERE VALUE WILL GO
1860  0E 66              MVI    C,SCR AND 377Q  ;SET UP AND CALL FINPT
1862  CD E8 0F           CALL   FINPT
1865  2A 86 21           LHLD   HLINP           ;RETORE H,L AND C
1868  3A 84 21           LDA    CREG
186B  4F                 MOV    C,A
186C  C9                 RET                    ;DCNE
                ;
                ; VAR DECIDES WHETHER A TOKEN IS
                ; A VARIABLE IF SO CY=1 AND
                ; ADDRESS IS CCMPUTED,(SUBSCRIPT IS
                ; EVALUATED ETC.), RETURNS WITH DE PNTING
                ; TO VAR. REFERENCED H,L,C,UPDATED
                ; A,B DESTROYED
                ; IF NOT A VARIBLE CY=0
                ; H,L,C ARE LEFT UNTOUCHED
                ;
186D  CD 22 12  VAR:     CALL   ALPHA           ;1ST CHAR A LETTER?
1870  D0                 RNC                    ;NO-NOT VAR.
1871  23                 INX    H               ;BUMP PNTR'S
1872  0D                 DCR    C
1873  C2 7F 18           JNZ    MORE            ;MORE TO LINE
1876  C5        SC1:     PUSH   B               ;SAVE B,EOL
1877  0E 00              MVI    C,0             ;SET FOR CALL TO FSYM
1879  2B                 DCX    H               ;GET SINGLE LETTER
187A  46                 MOV    B,M             ;VAR TO B
187B  23                 INX    H
187C  C3 B1 18           JMP    SCALR
187F  CD 22 12  MORE:    CALL   ALPHA           ;2ND A LETTER?
1882  D2 94 18           JNC    SFSG            ;SO FAR SO GOOD
1885  C5                 PUSH   B               ;SAVE C
1886  3E 02              MVI    A,2             ;CHECK FOR DELIMITER
1888  CD F7 12           CALL   SYMSRT
188B  C1                 POP    B               ;RESTORE C
188C  3C                 INR    A               ;FOUND?
188D  C2 76 18           JNZ    SC1             ;YES
1890  0C        BUPT:    INR    C               ;NOT A VAR.
1891  2B                 DCX    H               ;BACK UP PNTR'S
1892  B7                 ORA    A               ;CY=0 AND RET
1893  C9                 RET
1894  CD 14 12  SFSG:    CALL   NUMB            ;TEST FOR NUMBER
1897  D2 B9 18           JNC    ARCK            ;MAYBE AN ARRAY
189A  23                 INX    H               ;ITS A SCALAR
189B  0D                 DCR    C               ;BUMP PNTR'S
189C  CA AA 18           JZ     SLOAD           ;EOL
189F  C5                 PUSH   B               ;SAVE C
18A0  3E 02              MVI    A,2             ;SET UP FOR SYMSRT
18A2  CD F7 12           CALL   SYMSRT          ;TEST FOR LEGAL
18A5  C1                 POP    B               ;GET C BACK
18A6  3C                 INR    A               ;DELIMITER FOUND?
18A7  CA 22 1A           JZ     ER8             ;NO, ERROR
18AA  2B        SLOAD:   DCX    H               ;MOVE BACK,
18AB  C5                 PUSH   B               ;SAVE C,
18AC  4E                 MOV    C,M             ;GET VAR. INTO
18AD  2B                 DCX    H               ;B,C FOR FSYM
18AE  46                 MOV    B,M
18AF  23                 INX    H
18B0  23                 INX    H
18B1  EB        SCALR:   XCHG                   ;SAVE H,L IN D,E
18B2  CD 0D 16           CALL   FSYM            ;GET PNTR TO VALUE
18B5  EB                 XCHG                   ;RESTORE H,L PNTR TO DE
18B6  C1                 POP    B               ;GET C REG BACK
18B7  37                 STC                    ;SET CY,RET
18B8  C9                 RET
18B9  7E        ARCK:    MOV    A,M             ;ARRAY CHEK, GET CHARAC
18BA  FE A8              CPI    250Q            ;IS IT (?
18BC  CA CD 18           JZ     ARYES           ;YES,ITS AN ARRAY
18BF  3E 02              MVI    A,2             ;NO-CHEK FOR LEGAL DELI
18C1  C5                 PUSH   B               ;SAVE C
18C2  CD F7 12           CALL   SYMSRT
18C5  C1                 POP    B               ;RESTORE C
18C6  3C                 INR    A               ;DELIMITER FCUND?
18C7  CA 22 1A           JZ     ER8
18CA  C3 76 18           JMP    SC1             ;1 CHAR. SCALAR VAR.
18CD  2B        ARYES:   DCX    H               ;YES-WE HAVE ARRAY
18CE  7E                 MOV    A,M             ;GET VAR.
18CF  23                 INX    H
18D0  F5                 PUSH   PSW             ;SAVE VAR.
18D1  CD FB 15           CALL   ICP8            ;BUMP PNTR'S
18D4  CD C7 19           CALL   EVAL            ;EVALUATE SUBSCRIPT
18D7  E5                 PUSH   H               ;SAVE REG H,L
18D8  21 7C 21           LXI    H,FREG1
18DB  CD A8 17           CALL   COPDH           ;COPY IT
18DE  EB                 XCHG
18DF  E1                 POP    H               ;RESTORE H,L
18E0  CD 15 19           CALL   FIX             ;FIX VALUE
18E3  3E A9              MVI    A,251Q          ;CHECK FOR )
18E5  BE                 CMP    M
18E6  C2 22 1A           JNZ    ER8
18E9  23                 INX    H
18EA  0D                 DCR    C               ;BUMP PNTR'S
18EB  13                 INX    D               ;PNT TO LOWER 2 BYTES
18EC  13                 INX    D
18ED  1A                 LDAX   D
18EE  47                 MOV    B,A             ;H-BYTE TO B
18EF  13                 INX    D               ;PNT TO LOW BYTE
18F0  1A                 LDAX   D               ;LOW BYTE TO A
18F1  B7                 ORA    A               ;KILL CY
18F2  17                 RAL                    ;START MULT OF OFFSET
18F3  5F                 MOV    E,A             ;BY 4(BYTES/FLTPT #)
18F4  78                 MOV    A,B             ;GET H BYTE
18F5  17                 RAL
18F6  57                 MOV    D,A             ;DE IS OFFSET*2
18F7  7B                 MOV    A,E             ;GET LOW
18F8  B7                 ORA    A               ;KILL CARRY
18F9  17                 RAL
18FA  5F                 MOV    E,A
18FB  7A                 MOV    A,D
18FC  17                 RAL
18FD  57                 MOV    D,A
18FE  F1                 POP    PSW             ;DE CONTAIN OFFSET*4
18FF  C5                 PUSH   B               ;GET VAR., SAVE C
1900  4F                 MOV    C,A
1901  06 00              MVI    B,0             ;SETUP TO CALL FSYM
```

```
1903  E5                 PUSH   H               ;SAVE H,L
1904  CD 0D 16           CALL   FSYM            ;GET START ADD.
1907  DA 0F 19           JC     AFOND
190A  3E 12              MVI    A,12H           ;ERROR 12
190C  C3 BE 15           JMP    ERROR           ;ARRAY REF. NOT DIM'ED.
190F  19        AFOND:   DAD    D               ;H,L NOW PNT TO START O
1910  EB                 XCHG                   ;ARRAY, ADD OFFSET, EXC
1911  E1                 POP    H               ;RESTORE PNTR'S AND RET
1912  C1                 POP    B
1913  37                 STC                    ;SET CY
1914  C9                 RET
                ;
                ; ROUTINE TO FIX FLOATING POINT
                ; NUMBERS, ALL REG'S BUT A ARE
                ; MAINTAINED. DE PNT TO 4 BYTES
                ; OF # TO BE FIXED
                ;
1915  C5        FIX:     PUSH   B
1916  E5                 PUSH   H
1917  D5                 PUSH   D               ;SAVE REG'S
1918  13                 INX    D
1919  13                 INX    D
191A  13                 INX    D               ;PNT TO 4TH BYTE
191B  1A                 LDAX   D
191C  F5                 PUSH   PSW             ;SAVE CHAR. (FOR SIGN)
191D  E6 7F              ANI    177Q
191F  17                 RAL                    ;CHEK IF EXP SIGN IS -
1920  17                 RAL
1921  DA 30 19           JC     MINSE
1924  1F                 RAR
1925  1F                 RAR                    ;RESTORE CHAR
1926  FE 18              CPI    30Q             ;IS IT TOO BIG?
1928  DA 32 19           JC     GOOD
192B  3E 13              MVI    A,13H           ;ERROR 13
192D  C3 BE 15           JMP    ERROR           ;FIX # TOO BIG
1930  1F        MINSE:   RAR
1931  1F                 RAR
1932  12        GOOD:    STAX   D               ;ABSOLUTE VALUE
1933  1B                 DCX    D
1934  1B                 DCX    D
1935  1B                 DCX    D               ;MOV PNTR BACK
1936  21 7C 21           LXI    H,FREG1
1939  CD A8 17           CALL   COPDH           ;COPY TO FREG1
193C  21 80 21           LXI    H,FREG2         ;STORE .5*2**24 IN
193F  11 6F 19           LXI    D,FDAT          ;FREG2
1942  CD A8 17           CALL   COPDH           ;COPY IT
1945  21 7C 21           LXI    H,FREG1         ;SET UP TO CALL LADD
1948  06 80              MVI    B,FREG2 AND 377Q
194A  0E 66              MVI    C,SCR AND 377Q
194C  CD D0 0F           CALL   LADD            ;ADD THEM,RESULT IN FRE
194F  21 7C 21           LXI    H,FREG1
1952  F1                 POP    PSW             ;GET SIGN AND ADD.
1953  D1                 POP    D
1954  17                 RAL
1955  3E 00              MVI    A,0             ;GET SIGN ONLY
1957  1F                 RAR
1958  46                 MOV    B,M             ;GET BYTE1
1959  12                 STAX   D               ;STORE BYTE 1 OF FIX
195A  78                 MOV    A,B
195B  E6 7F              ANI    177Q            ;CLEAR HIGH BIT (FROM A
195D  13                 INX    D
195E  23                 INX    H
195F  46                 MOV    B,M             ;GET BYTE 2
1960  12                 STAX   D               ;STORE BYTE 2 OF FIX
1961  13                 INX    D
1962  78                 MOV    A,B
1963  23                 INX    H
1964  46                 MOV    B,M             ;GET BYTE 3
1965  12                 STAX   D               ;STORE BYTE 3 OF FIX
1966  78                 MOV    A,B
1967  13                 INX    D
1968  12                 STAX   D               ;STORE BYTE 4 OF FIX
1969  1B                 DCX    D               ;FIX D PNTR
196A  1B                 DCX    D
196B  1B                 DCX    D
196C  E1                 POP    H
196D  C1                 POP    B
196E  C9                 RET
196F  80 00 00 18  FDAT: DB     200Q,0,0,30Q
                ;
                ; INP SAVES ALL REG'S
                ; SERVES AS A BUFFER ROUTINE BETWEEN FINPT AND
                ; DATA INPUT. IF MODE=0, DATA COMES FROM TTY
                ; IF MODE=1 DATA COMES FROM SOURCE STMTS.
                ; IN ALL CASES HL,C ARE UPDATED FROM HLINP, AND
                ; CREG AND RETURNED TO THOSE LOCATIONS
                ;
1973  E5        INP:     PUSH   H               ;SAVE ALL REG'S
1974  D5                 PUSH   D
1975  C5                 PUSH   B
1976  2A 86 21           LHLD   HLINP           ;GET PNTR'S
1979  3A 84 21           LDA    CREG
197C  4F                 MOV    C,A
197D  B7                 ORA    A               ;CHECK FOR EOL
197E  C2 87 19           JNZ    CHKMD           ;NO CHECK MODE
1981  3E A0     SPACE:   MVI    A,240Q          ;SEND A SPACE
1983  C1        IDONE:   POP    B               ;RESTORE REG'S
1984  D1                 POP    D
1985  E1                 POP    H
1986  C9                 RET                    ;AND RETURN
1987  3A 85 21  CHKMD:   LDA    MODE            ;GET MODE
198A  3D                 DCR    A               ;CHECK IT
198B  CA 97 19           JZ     MODE1           ;MODE IS 1
198E  7E                 MOV    A,M             ;MODE 0, GET CHAR.
198F  FE AC              CPI    ',' OR 200Q     ;IS IT A ','?
1991  CA 81 19           JZ     SPACE           ;YES - SEND A SPACE
1994  C3 BB 19           JMP    BMPTR           ;NO - SEND IT
1997  CD 14 12  MODE1:   CALL   NUMB            ;NUMBER? (ALSO LOADS IT
199A  DA BB 19           JC     BMPTR           ;YES - SEND IT AND BUMP
199D  FE AE              CPI    256Q            ;DEC. PNT.?
199F  CA BB 19           JZ     BMPTR
19A2  FE C5              CPI    305Q            ;E?
19A4  CA BB 19           JZ     BMPTR
19A7  FE AB              CPI    253Q            ;+?
19A9  CA B1 19           JZ     CHEKE
19AC  FE AD              CPI    255Q            ;-?
19AE  C2 81 19           JNZ    SPACE           ;SEND A SPACE
19B1  47        CHEKE:   MOV    B,A             ;CHEK IF E PRECEDES +,-
19B2  2B                 DCX    H               ;BACK UP AND GET PRE-
19B3  7E                 MOV    A,M             ;CEDING CHARACTER
19B4  FE C5              CPI    305Q            ;IS IT E?
19B6  C2 81 19           JNZ    SPACE           ;NO,+OR- WAS DELIMITTER
19B9  78                 MOV    A,B             ;YES,GET + OR -
19BA  23                 INX    H               ;RESTORE H,L
19BB  23        BMPTR:   INX    H               ;BUMP AND STORE PNTR'S
19BC  0D                 DCR    C
19BD  22 86 21           SHLD   HLINP
19C0  21 84 21           LXI    H,CREG
19C3  71                 MOV    M,C
19C4  C3 83 19           JMP    IDONE           ;RESTORE REG'S AND RETJ
                ;
                ; THIS ROUTINE WILL EVALUATE UNARY AND/OR
                ; BINARY EXPRESIONS CALLED WITH H AND L
                ; POINTING TO FIRST CHAR. OF EXP.,C CONTAINS
                ; NUMBER OF CHAR'S LEFT IN LINE. RETURNS
                ; D(HIGH) AND E(LOW) POINTING TO THE ANSWER
                ; THIS ROUTINE CALLS ITSELF RECURSIVELY
                ; IN ORDER TO EVALUATE SUBSCRIPT
                ; EXPRESIONS.  REG A,B DESTROYED
                ; C,H,L ARE UPCATED
                ;
19C7  3E AD     EVAL:    MVI    A,255Q          ;IS IT UNARY -
19C9  BE                 CMP    M               ;Z=1 => YES
19CA  F5                 PUSH   PSW             ;Z=0 => NO
19CB  C2 D1 19           JNZ    ECAV
19CE  CD FB 15           CALL   ICP8            ;BUMP POINTER
19D1  CD C3 17  ECAV:    CALL   VALUE           ;GET PNTR. TO VALUE
19D4  E5                 PUSH   H               ;GET VALUE TO FREG1
19D5  21 7C 21           LXI    H,FREG1
19D8  CD A8 17           CALL   COPDH
19DB  EB                 XCHG
19DC  E1                 POP    H
19DD  F1                 POP    PSW             ;GET SIGN
19DE  C2 EC 19           JNZ    DOL             ;SHALL WE NEGATE?
19E1  13                 INX    D               ;YES, POINT TO CHAR.
19E2  13                 INX    D
19E3  13                 INX    D
19E4  1A                 LDAX   D               ;AND LOAD TO A
19E5  17                 RAL                    ;ROTATE SIGN TO CY
19E6  3F                 CMC                    ;COMPLEMENT IT
19E7  1F                 RAR                    ;ROTATE BACK
19E8  12                 STAX   D               ;STORE AWAY
19E9  1B                 DCX    D               ;AND FIX PNTR.
19EA  1B                 DCX    D
19EB  1B                 DCX    D
19EC  79        DOL:     MOV    A,C             ;IS THIS END OF LINE?
19ED  B7                 ORA    A
19EE  C8                 RZ                     ;YES-RETURN
19EF  C5                 PUSH   B               ;SAVE C
19F0  3E 02              MVI    A,2             ;NO SET UP TO CALL
19F2  CD F7 12           CALL   SYMSRT          ;SYMSRT AND CALL
19F5  C1                 POP    B               ;RESTORE C
19F6  3C                 INR    A               ;DELIMITER FOUND?
19F7  CA 22 1A           JZ     ER8             ;NO, ERROR
19FA  D6 0A     EOK:     SUI    10              ;CHECK FOR EXPRESSION
19FC  D8                 RC                     ;DELIMITER
19FD  F5                 PUSH   PSW             ;SAVE OVERATION
19FE  CD FB 15           CALL   ICP8            ;BUMP PNTR'S
1A01  B7                 ORA    A               ;CLEAR CY
1A02  E5        AGA:     PUSH   H               ;GET BYTES OF NUMBER
```

```
1A03  1A                  LDAX   D                    ;AND PLACE ON STACK
1A04  6F                  MOV    L,A
1A05  13                  INX    D
1A06  1A                  LDAX   D
1A07  13                  INX    D
1A08  67                  MOV    H,A                  ;2 BYTES TO H,L
1A09  E3                  XTHL                        ;XCHANGE, RESTORES H,L
1A0A  3F                  CMC
1A0B  DA 02 1A            JC     AGA                  ;ANOTHER PASS?
1A0E  CD C3 17            CALL   VALUE                ;GET 2ND VALUE
1A11  79                  MOV    A,C                  ;CHECK FOR END OF LINE
1A12  B7                  ORA    A                    ;IF SO => WELL FORMED
1A13  CA 27 1A            JZ     WFOR
1A16  C5                  PUSH   B                    ;SAVE C
1A17  3E 02               MVI    A,2                  ;ELSE CALL SYMSRT TO
1A19  CD F7 12            CALL   SYMSRT               ;CHEK FOR EXP. DEL.
1A1C  C1                  POP    B                    ;RECOVER IT
1A1D  FE 0A               CPI    10
1A1F  DA 27 1A            JC     WFOR                 ;YES, WELL FORMED
1A22  3E 08       ER8:    MVI    A,8                  ;ILL-FORMED EXP.
1A24  C3 BE 15            JMP    ERROR
1A27  C5          WFOR:   PUSH   B                    ;SAVE C, AND H,L
1A28  E5                  PUSH   H
1A29  21 80 21            LXI    H,FREG2              ;COPY 2ND VALUE TO
1A2C  CD A8 17            CALL   COPDH                ;FREG2
1A2F  D1                  POP    D                    ;GET BYTES FROM STACK
1A30  C1                  POP    B
1A31  E1                  POP    H                    ;INTO FREG1+2
1A32  22 7E 21            SHLD   FREG1+2
1A35  E1                  POP    H                    ;AND NEXT 2 BYTES
1A36  22 7C 21            SHLD   FREG1                ;FROM STACK TO FREG1
1A39  EB                  XCHG
1A3A  F1                  POP    PSW                  ;GET OPERATION
                  ;
                  ; THIS ROUTINE PERFORMS BINARY OPERATIONS ON OPERANDS IN
                  ; B,C,H,L ARE LEFT UNDISTURBED. A IS DESTROYED
                  ; D,E PNT TO RESULT
                  ; OPERATIONS ARE SPECIFIED BY A REGISTER AS FOLLOWS:
                  ;
                  ;       A=0     =>            FREG1 * FREG2
                  ;       A=1     =>            FREG1 / FREG2
                  ;       A=2     =>            FREG1 + FREG2
                  ;       A=3     =>            FREG1 - FREG2
                  ;
                  ; IN CASE OF ARITHMETIC ERROR A MESSAGE IS SENT TO USER.
                  ; IF A CONTAINS ILLEGAL OPERATION REQUEST ERROR IS SENT T
                  ; (ERROR 8) AND RUN (THE INTERPRETER) IS ABORTED.
                  ;
1A3B  C5          BINOP:  PUSH   B                    ;SAVE REG'S
1A3C  E5                  PUSH   H
1A3D  21 7C 21            LXI    H,FREG1              ;SET UP PNTR'S TO
1A40  06 80               MVI    B,FREG2 AND 377Q     ;FREG'S AND SCR AREA
1A42  0E 66               MVI    C,SCR AND 377Q       ;AND DO OPERATION
1A44  3D                  DCR    A
1A45  FA 87 1A            JM     FMULT                ;0,1=>* OR /
1A48  CA 8D 1A            JZ     DIV                  ;2,3=>+ OR -
1A4B  3D                  DCR    A
1A4C  CA 56 1A            JZ     ADDD
1A4F  3D                  DCR    A
1A50  CA 81 1A            JZ     SUBB
1A53  C3 22 1A            JMP    ER8                  ;ILLEGAL OPER.
1A56  CD D0 0F    ADDD:   CALL   LADD                 ;DO ADDITION
1A59  54          ASBC:   MOV    D,H                  ;FIX PNTR'S FOR RET.
1A5A  5D                  MOV    E,L
1A5B  B7          FPERR:  ORA    A                    ;SET FLAGS
1A5C  CA 7E 1A            JZ     NFPER                ;NO ERROR
1A5F  D5                  PUSH   D                    ;SAVE DE
1A60  F5                  PUSH   PSW                  ;SAVE A
1A61  CD D0 12            CALL   WRIT                 ;DUMP BUFFER
1A64  F1                  POP    PSW                  ;GET A BACK
1A65  21 77 1A            LXI    H,WFPER              ;RETURN ADDRESS
1A68  E5                  PUSH   H                    ;SAVE ON STACK
1A69  21 5A 14            LXI    H,ODATA              ;MESSAGE TABLE
1A6C  17                  RAL                         ;UNDERFLOW?
1A6D  DA 3F 14            JC     FOR12                ;YES
1A70  17                  RAL                         ;OVERFLOW?
1A71  DA 40 14            JC     FOR11                ;YES
1A74  C3 41 14            JMP    FOR10                ;NO - ITS ZERODIVIDE
1A77  21 5A 14    WFPER:  LXI    H,ODATA              ;MESSAGE TABLE
1A7A  CD DF 15            CALL   ERLN                 ;PRINT 'IN LINE --' (US
1A7D  D1                  POP    D                    ;RESTORE REG'S
1A7E  E1          NFPER:  POP    H
1A7F  C1                  POP    B
1A80  C9                  RET
1A81  CD D9 0F    SUBB:   CALL   LSUB                 ;DO SUBTRACTION
1A84  C3 59 1A            JMP    ASBC
1A87  CD D3 0F    FMULT:  CALL   LMUL                 ;DO MULT.
1A8A  C3 90 1A            JMP    MDBC
1A8D  CD D6 0F    DIV:    CALL   LDIV                 ;DO DIV.
1A90  54          MDBC:   MOV    D,H                  ;AND FIX PNTR'S FOR RET
1A91  59                  MOV    E,C
1A92  C3 5B 1A            JMP    FPERR                ;CHECK FOR ERROR
                  ;
                  ; PRINT PROCESSOR
                  ;
1A95  2A 5B 21    PRI:    LHLD   CPNT
1A98  23                  INX    H                    ;INCR. PAST KEYWORD
1A99  23                  INX    H
1A9A  23                  INX    H
1A9B  CD F6 15            CALL   ICP7
1A9E  23                  INX    H                    ;BUMP PNTRS
1A9F  0D                  DCR    C
1AA0  06 00               MVI    B,0                  ;SET CHAR CNT
1AA2  C2 AE 1A            JNZ    PLOOP                ;CONTINUE IF MORE
1AA5  04                  INR    B                    ;NOTHING MORE, PAD A NU
1AA6  3E 00               MVI    A,0
1AA8  CD AD 12            CALL   PAD
1AAB  C3 46 1B            JMP    PEND                 ;WRITE IT AND CONTINUE
1AAE  7E          PLOOP:  MOV    A,M                  ;GET CHARACTER
1AAF  FE A2               CPI    '"'+200Q             ;IS IT "?
1AB1  C2 D6 1A            JNZ    EXPRE                ;NO
1AB4  CD F6 15    QUOTE:  CALL   ICP7                 ;GET CHARACTER TO A
1AB7  7E                  MOV    A,M
1AB8  FE A2               CPI    '"'+200Q             ;IS IT "?
1ABA  CA C8 1A            JZ     QCHEK
1ABD  04          QOTOK:  INR    B                    ;INCREMENT CNT
1ABE  50                  MOV    D,B                  ;SAVE IN D
1ABF  06 01               MVI    B,1                  ;PAD ONCE
1AC1  CD AD 12            CALL   PAD
1AC4  42                  MOV    B,D                  ;RESTORE CNT
1AC5  C3 B4 1A            JMP    QUOTE                ;AGAIN
1AC8  23          QCHEK:  INX    H                    ;BUMP PNTRS
1AC9  0D                  DCR    C
1ACA  CA 46 1B            JZ     PEND                 ;EOL
1ACD  7E                  MOV    A,M
1ACE  FE A2               CPI    '"'+200Q             ;ANOTHER "?
1AD0  CA BD 1A            JZ     QOTOK
1AD3  C3 07 1B            JMP    SCOLN
1AD6  CD 22 12    EXPRE:  CALL   ALPHA                ;IS IT A LETTER
1AD9  DA ED 1A            JC     PRTIT                ;YES, EVALUATE AND PRIN
1ADC  CD 14 12            CALL   NUMB                 ;IS IT A NUMB?
1ADF  DA ED 1A            JC     PRTIT                ;YES, EVALUATE AND PRIN
1AE2  7E                  MOV    A,M
1AE3  FE AE               CPI    '.'+200Q             ;IS IT A DECIMAL PNT?
1AE5  CA ED 1A            JZ     PRTIT                ;YES EVALUATE, PRINT
1AE8  FE AD               CPI    '-'+200Q             ;IS IT A -?
1AEA  C2 07 1B            JNZ    SCOLN                ;NO, CHECK FOR ;
1AED  C5          PRTIT:  PUSH   B                    ;SAVE CNT
1AEE  CD C7 19            CALL   EVAL                 ;EVALUATE EXPRESION
1AF1  C5                  PUSH   B                    ;SAVE C,H,L
1AF2  E5                  PUSH   H
1AF3  EB                  XCHG                        ;DE TO HL
1AF4  0E 66               MVI    C,SCR AND 377Q       ;SET UP, CONVERT
1AF6  CD E5 0F            CALL   CONV
1AF9  E1                  POP    H                    ;RESTORE REG'S
1AFA  C1                  POP    B
1AFB  79                  MOV    A,C
1AFC  C1                  POP    B
1AFD  4F                  MOV    C,A
1AFE  B7                  ORA    A                    ;CHECK EOL
1AFF  CA 46 1B            JZ     PEND
1B02  3E 0B               MVI    A,11                 ;UPDATE CNTR
1B04  80                  ADD    B
1B05  47                  MOV    B,A
1B06  7E                  MOV    A,M                  ;GET CHAR.
1B07  FE BB       SCOLN:  CPI    ';'+200Q             ;IS IT ;?
1B09  CA 39 1B            JZ     SONWD                ;YES
1B0C  FE AC               CPI    ','+200Q             ;IS IT ,?
1B0E  C2 2C 17            JNZ    ER6                  ;NO-UNEXPECTED CHAR.
1B11  AF                  XRA    A                    ;ZERO A
1B12  C6 0D       ADFLD:  ADI    13                   ;ADD FIELD LENGTH
1B14  B8                  CMP    B                    ;COMPARE TO CNT
1B15  CA 1B 1B            JZ     $+6
1B18  D2 2D 1B            JNC    FLDFD
1B1B  FE 34               CPI    52                   ;LAST FLD?
1B1D  C2 12 1B            JNZ    ADFLD
1B20  CD D0 12            CALL   WRIT                 ;YES-WRITE LINE
1B23  06 00               MVI    B,0                  ;RESET CNT
1B25  23          ONWD:   INX    H                    ;BUMP PNTRS
1B26  0D                  DCR    C
1B27  CA 46 1B            JZ     PEND
1B2A  C3 AE 1A            JMP    PLOOP
1B2D  90          FLDFD:  SUB    B                    ;FOUND FIELD
1B2E  50                  MOV    D,B                  ;DETERMIN  OF SPACES T
1B2F  5F                  MOV    E,A                  ;SET UP TO CALL PAD
1B30  47                  MOV    B,A
1B31  3E A0               MVI    A,240Q
1B33  CD AD 12            CALL   PAD                  ;PAD SPACES
1B36  7A                  MOV    A,D
1B37  83                  ADD    E                    ;NEW CNT
1B38  47                  MOV    B,A                  ;SAVE IN B
1B39  23          SONWD:  INX    H                    ;CHECK EOL
1B3A  0D                  DCR    C
1B3B  C2 AE 1A            JNZ    PLOOP
```

```
1B3E  16 01               MVI    D,1                  ;SUPPRESS CR/LF
1B40  CD D2 12            CALL   WRIT1
1B43  C3 49 1B            JMP    $+6
1B46  CD D0 12    PEND:   CALL   WRIT                 ;DUMP BUFFER, CONTINUE
1B49  C3 8C 1B            JMP    IEND
                  ;
                  ; INPUT PROCESSOR - READS VALUES FROM TTY
                  ; THEY MUST BE DELIMITED BY COMMAS ONLY
                  ;
1B4C  79          INPUT:  MOV    A,C                  ;IN CASE OF ERROR
1B4D  32 5F 21            STA    PL6                  ;SAVE
1B50  2A 5B 21    INPER:  LHLD   CPNT                 ;INPUT LINE (V-STRING)
1B53  23                  INX    H                    ;ADJUST PNTR'S
1B54  23                  INX    H
1B55  23                  INX    H
1B56  CD F6 15            CALL   ICP7
1B59  CD F6 15            CALL   ICP7
1B5C  C5          PRMPT:  PUSH   B                    ;SAVE PNTR'S
1B5D  E5                  PUSH   H
1B5E  06 01               MVI    B,1                  ;SEND PROMPT
1B60  3E 3A               MVI    A,':'
1B62  50                  MOV    D,B                  ;TO SUPPRESS CR/LF
1B63  CD AD 12            CALL   PAD                  ;PAD IT
1B66  CD D2 12            CALL   WRIT1                ;WRITE IT
1B69  21 01 21            LXI    H,IBUF               ;ADD. OF INPUT BUFFER
1B6C  CD C9 13            CALL   TTYIN                ;READ A LINE
1B6F  EB                  XCHG                        ;ADD. OF K-STRING TO DE
1B70  E1                  POP    H                    ;ADD. OF V-STRING
1B71  C1                  POP    B                    ;V-STRING CNT TO C
1B72  47                  MOV    B,A                  ;K-STRING CNT TO B
1B73  CD 92 1B            CALL   STRIN                ;TRANSFER CONSTANTS TO
1B76  CA 89 1B            JZ     INPOK                ;NO ERROR
1B79  21 5A 14            LXI    H,ODATA              ;SEND ERROR MESSAGE
1B7C  CD 42 14            CALL   FORM9
1B7F  CD D0 12            CALL   WRIT
1B82  3A 5F 21            LDA    PL6                  ;GET V-STRING CNT
1B85  4F                  MOV    C,A
1B86  C3 50 1B            JMP    INPER                ;START AGAIN
1B89  DA 5C 1B    INPOK:  JC     PRMPT                ;NEED MORE CONSTANTS
1B8C  2A 56 21    IEND:   LHLD   KFPNT                ;ALL OK - GET NEW PNTR.
1B8F  C3 8C 16            JMP    ILOOP                ;CONTINUE
                  ;
                  ; THIS ROUTINE TRANSFERS THE FLOATING POINT VALUES
                  ; OF AN ASCII STRING OF CONSTANTS TO THE LOCATIONS
                  ; SPECIFIED BY AN ASCII STRING OF VARIBLES
                  ; POINTER AND LINE CNT OF VAR. STRING ARE IN HL,C
                  ; POINTER AND LINE CNT OF CONST. STRING ARE IN DE,B
                  ; ON RETURN:
                  ;
                  ;       Z=0 AND CY=0   ALL OK
                  ;       Z=0 AND CY=1   NEED MORE CONSTANTS
                  ;       Z=1            ERROR IN CONVERSION
                  ;
                  ; ALL POINTERS AND LINE CNT'S ARE RETURNED UPDATED
                  ;
1B92  79          STRIN:  MOV    A,C                  ;GET V-STRING CNT
1B93  B7                  ORA    A                    ;TEST FOR EOL
1B94  C8                  RZ                          ;DONE, CY=0 => ALL OK
1B95  7E                  MOV    A,M                  ;GET CHAR.
1B96  FE AC               CPI    ',' OR 200Q          ;IS IT A ,?
1B98  C2 A0 1B            JNZ    STOKV                ;IT'S NOT A ,
1B9B  23                  INX    H                    ;COMMA, BUMP PNTR'S
1B9C  0D                  DCR    C
1B9D  CA C8 1B            JZ     ERRET                ;POSSIBLE ERROR (IF EOL
1BA0  78          STOKV:  MOV    A,B                  ;GET K-STRING LENGTH
1BA1  B7                  ORA    A                    ;TEST FOR EOL
1BA2  37                  STC                         ;IN CASE IT'S EOL
1BA3  C8                  RZ                          ;RET, CY=1 => NEED MORE
1BA4  1A                  LDAX   D                    ;GET CHAR
1BA5  FE AC               CPI    ',' OR 200Q          ;TEST FOR ,
1BA7  C2 AF 1B            JNZ    STOKK                ;NOT A , - READY TO GO
1BAA  13                  INX    D                    ;BUMP PNTR'S
1BAB  05                  DCR    B
1BAC  CA C8 1B            JZ     ERRET                ;POSSIBLE ERROR (IF EOL
1BAF  C5          STOKK:  PUSH   B                    ;SAVE K-STRING CNT
1BB0  D5                  PUSH   D                    ;SAVE K-STRING PNTR
1BB1  CD 6D 18            CALL   VAR                  ;ADD. TO VARIBLE TO DE
1BB4  EB                  XCHG                        ;VAR. ADD TO H,L
1BB5  22 8A 21            SHLD   VARAD                ;SAVE
1BB8  E1                  POP    H                    ;ADDRESS OF K-STRING
1BB9  79                  MOV    A,C                  ;V-STRING CNT TO A
1BBA  C1                  POP    B                    ;K-STRING CNT TO B
1BBB  48                  MOV    C,B                  ;K-STRING CNT TO C
1BBC  F5                  PUSH   PSW                  ;SAVE V-STRING CNT
1BBD  D5                  PUSH   D                    ;SAVE V-STRING ADD.
1BBE  3E 00               MVI    A,0                  ;A=0 => DATA FROM TTY
1BC0  CD 53 18            CALL   RDKON                ;GET CONSTANT TO GREG
1BC3  D2 CB 1B            JNC    STNER
1BC6  E1                  POP    H                    ;EMPTY STACK
1BC7  E1                  POP    H
1BC8  AF          ERRET:  XRA    A                    ;ERROR
1BC9  3C                  INR    A
1BCA  C9                  RET
1BCB  E5          STNER:  PUSH   H                    ;SAVE K-STRING PNTR.
1BCC  2A 8A 21            LHLD   VARAD                ;GET VAR. ADD
1BCF  11 77 21            LXI    D,GREG               ;ADD. TO CONST.
1BD2  CD A8 17            CALL   COPDH                ;COPY IT TO VARIABLE LO
1BD5  D1                  POP    D                    ;K-STING PNTR. TO DE
1BD6  41                  MOV    B,C                  ;K-STRING LENGTH TO B
1BD7  E1                  POP    H                    ;V-STRING PNTR. TO HL
1BD8  F1                  POP    PSW                  ;V-STRING LENGTH TO C
1BD9  4F                  MOV    C,A
1BDA  C3 92 1B            JMP    STRIN                ;LOOP
                  ;
                  ; LET STMT. PROCESSOR
                  ;
1BDD  2A 5B 21    LET:    LHLD   CPNT                 ;GET PNTR.
1BE0  23                  INX    H                    ;FIX PNTR.
1BE1  23                  INX    H
1BE2  23                  INX    H
1BE3  79                  MOV    A,C                  ;CHECK FOR EOL
1BE4  B7                  ORA    A
1BE5  C2 ED 1B            JNZ    LOK
1BE8  3E 07       ER7:    MVI    A,7
1BEA  C3 BE 15            JMP    ERROR
1BED  CD 6D 18    LOK:    CALL   VAR                  ;GET ADDRESS TO VAR.
1BF0  DA 32 1C            JC     SAVV                 ;IT'S A VARIABLE
1BF3  3E 03               MVI    A,3                  ;NO-CHEK FOR FUNC.
1BF5  CD F7 12            CALL   SYMSRT
1BF8  FE FF               CPI    377Q
1BFA  CA 22 1A            JZ     ER8                  ;DON'T KNOW WHAT IT IS
1BFD  3D                  DCR    A
1BFE  C2 78 1C            JNZ    ER10                 ;ILLEGAL USE OF FUNC.
1C01  23                  INX    H                    ;IT'S PUT,UPDATE H,L
1C02  23                  INX    H
1C03  23                  INX    H
1C04  79                  MOV    A,C                  ;EOL CHK
1C05  B7                  ORA    A
1C06  CA 22 1A            JZ     ER8
1C09  7E                  MOV    A,M                  ;CHEK FOR (
1C0A  FE A8               CPI    250Q
1C0C  C2 22 1A            JNZ    ER8
1C0F  CD FB 15            CALL   ICP8                 ;BUMP PNTRS
1C12  CD C7 19            CALL   EVAL                 ;EVALUATE AND FIX
1C15  E5                  PUSH   H                    ;SAVE H,L
1C16  21 7C 21            LXI    H,FREG1
1C19  CD A8 17            CALL   COPDH                ;COPY IT
1C1C  EB                  XCHG
1C1D  E1                  POP    H
1C1E  CD 15 19            CALL   FIX
1C21  13                  INX    D
1C22  13                  INX    D
1C23  13                  INX    D
1C24  1A                  LDAX   D                    ;GET LOWEST BYTE
1C25  F5                  PUSH   PSW                  ;PORT # IS SAVED
1C26  7E                  MOV    A,M
1C27  FE A9               CPI    251Q                 ;CHECK FOR )
1C29  C2 22 1A            JNZ    ER8
1C2C  CD FB 15            CALL   ICP8                 ;BUMP PNTR'S
1C2F  16 FF               MVI    D,377Q
1C31  5A                  MOV    E,D
1C32  D5          SAVV:   PUSH   D                    ;KEEP ADDRESS
1C33  7E                  MOV    A,M                  ;CHEK FOR =
1C34  FE BD               CPI    275Q
1C36  C2 22 1A            JNZ    ER8
1C39  CD FB 15            CALL   ICP8                 ;BUMP PNTRS
1C3C  CD C7 19            CALL   EVAL                 ;EVALUATE EXPRESSION
1C3F  E1                  POP    H                    ;GET ADDRESS
1C40  CD A3 12            CALL   CHK1
1C43  DA 4C 1C            JC     PTFIN                ;IT WAS A PUT
1C46  CD A8 17            CALL   COPDH                ;COPY TO ADDRESS
1C49  C3 8C 1B            JMP    IEND                 ;CONTINUE
1C4C  21 7C 21    PTFIN:  LXI    H,FREG1              ;COPY VALUE TO FREG1
1C4F  CD A8 17            CALL   COPDH
1C52  EB                  XCHG
1C53  CD 15 19            CALL   FIX                  ;FIX THE VALUE
1C56  13                  INX    D
1C57  13                  INX    D
1C58  13                  INX    D
1C59  1A                  LDAX   D
1C5A  4F                  MOV    C,A                  ;SAVE IN C
1C5B  21 73 1C            LXI    H,PINST              ;ADD OF BYTES TO GO TO
1C5E  11 77 21            LXI    D,GREG               ;RAM AT GREG
1C61  06 05               MVI    B,5                  ;BYTE CNT
1C63  7E          PRI1:   MOV    A,M                  ;STORE PROG. SEG. IN
1C64  12                  STAX   D                    ;RAM
1C65  23                  INX    H
1C66  13                  INX    D
1C67  05                  DCR    B
1C68  C2 63 1C            JNZ    PRI1
1C6B  F1                  POP    PSW                  ;GET PORT #
1C6C  21 78 21            LXI    H,GREG+1
1C6F  77                  MOV    M,A                  ;STORE
1C70  79                  MOV    A,C                  ;GET DATA OUT TO A
1C71  2B                  DCX    H                    ;TRANSFER
```

```
1C72    F9                          PCHL
1C73    D3 00           PINST:      OUT     0
1C75    C3 8C 1B                    JMP     IEND            ;RAM INSTRUCTIONS
1C78    3E 10           ER10:       MVI     A,10H
1C7A    C3 BF 15                    JMP     ERROR
                        ;
                        ; IF STMT. PROCESSOR
                        ;
1C7D    2A 5B 21        IFRT:       LHLD    CPNT
1C80    23                          INX     H               ;GET PNTR., ADJUST
1C81    0C                          INR     C
1C82    CD F6 15                    CALL    ICP7            ;CHECK EOL
1C85    CD C7 19                    CALL    FVAL            ;EVALUATE EXPRESSION
1C88    79                          MOV     A,C
1C89    B7                          ORA     A               ;CHECK EOL
1C8A    CA E8 1B                    JZ      ER7
1C8D    E5              IAGA:       PUSH    H               ;SAVE H,L, PUT VALUE ON
1C8E    1A                          LDAX    D
1C8F    13                          INX     D
1C90    6F                          MOV     L,A
1C91    1A                          LDAX    D
1C92    13                          INX     D
1C93    67                          MOV     H,A
1C94    E3                          XTHL                    ;RESTORE H,L
1C95    3F                          CMC
1C96    DA 8D 1C                    JC      IAGA            ;ANOTHER PASS?
1C99    3E 02                       MVI     A,2
1C9B    CD F7 12                    CALL    SYMSRT          ;CHEK TYPE OF RELATION
1C9E    FE 04                       CPI     4               ;WAS IT LEGAL?
1CA0    DA A8 1C                    JC      II1
1CA3    3E 14           ER14:       MVI     A,14H
1CA5    C3 BE 15                    JMP     ERROR
1CA8    FE 02           II1:        CPI     2               ;WAS IT A ,?
1CAA    CA A3 1C                    JZ      ER14
1CAD    3C                          INR     A               ;ALL OK, INC,SAVE
1CAE    F5                          PUSH    PSW
1CAF    0C                          INR     C
1CB0    CD F6 15                    CALL    ICP7            ;BUMP PNTRS
1CB3    3E 02                       MVI     A,2             ;CALL SYMSRT
1CB5    CD F7 12                    CALL    SYMSRT
1CB8    FE FF                       CPI     377Q            ;FOUND ANYTHING?
1CBA    CA D5 1C                    JZ      RELAT           ;DONE
1CBD    FE 02                       CPI     2
1CBF    CA A3 1C                    JZ      ER14            ;IT WAS A ,
1CC2    FE 04                       CPI     4
1CC4    D2 A3 1C                    JNC     ER14            ;NOT LEGAL
1CC7    3C                          INR     A
1CC8    47                          MOV     B,A
1CC9    0C                          INR     C
1CCA    CD F6 15                    CALL    ICP7
1CCD    F1                          POP     PSW             ;GET SECOND RELATION
1CCE    80                          ADD     B               ;ADD THEM
1CCF    F5                          PUSH    PSW             ;AND SAVE
1CD0    FE 08                       CPI     10Q             ;TEST FOR ==
1CD2    CA A3 1C                    JZ      ER14
                        ;
                        ; AT THIS POINT:
                        ; RELATION IS STORED ON TOP OF STACK (PUSH PSW) ACCORDING
                        ; THE FOLLOWING
                        ;
                        ;           1 =>    <
                        ;           2 =>    >
                        ;           3 =>    <>
                        ;           4 =>    =
                        ;           5 =>    <=
                        ;           6 =>    >=
                        ;
1CD5    CD C7 19        RELAT:      CALL    EVAL            ;EVALUATE
1CD8    E5                          PUSH    H               ;SAVE H,L
1CD9    21 80 21                    LXI     H,FREG2         ;COPY TO FREG2
1CDC    CD A8 17                    CALL    COPDH
1CDF    E1                          POP     H               ;GET H,L
1CE0    F1                          POP     PSW             ;AND RELATION
1CE1    E3                          XTHL                    ;GET 2ND 2 BYTES
1CE2    22 7E 21                    SHLD    FREG1+2         ;STORE
1CE5    E1                          POP     H               ;GET 1ST 2 BYTES,STORE
1CE6    E3                          XTHL
1CE7    22 7C 21                    SHLD    FREG1
1CEA    C5                          PUSH    B
1CEB    F5                          PUSH    PSW             ;SAVE A,B,C
1CEC    CD 23 1D                    CALL    FCOMP           ;COMPARE NUMBERS
1CEF    57                          MOV     D,A             ;SAVE RESULT IN D
1CF0    F1                          POP     PSW             ;GET RELATION,B,C
1CF1    C1                          POP     B
1CF2    BA                          CMP     D               ;SAME?
1CF3    CA 16 1D                    JZ      TRUE            ;YES
1CF6    D6 04                       SUI     4
1CF8    F2 08 1D                    JP      NOT3            ;NOT RELATION 3
1CFB    3C                          INR     A               ;IS IT RELATION 3?
1CFC    C2 12 1D                    JNZ     FALSF           ;NO, ITS FALSE
1CFF    3E 04                       MVI     A,4             ;IT IS, CHECK FOR INEQU
1D01    BA                          CMP     D
1D02    C2 16 1D                    JNZ     TRUE
1D05    C3 12 1D                    JMP     FALSF
1D08    BA              NOT3:       CMP     D               ;RELATION 5,6 TRUE?
1D09    CA 16 1D                    JZ      TRUE            ;YES
1D0C    3E 04                       MVI     A,4             ;IT WAS, CHECK FOR EQUA
1D0E    BA                          CMP     D
1D0F    CA 16 1D                    JZ      TRUE
1D12    E1              FALSE:      POP     H               ;CONTINUE
1D13    C3 8C 1B                    JMP     IEND
1D16    E1              TRUE:       POP     H
1D17    06 04                       MVI     B,4
1D19    CD F6 15        THEN:       CALL    ICP7            ;INCREMENT PAST THEN
1D1C    05                          DCR     B
1D1D    C2 19 1D                    JNZ     THEN
1D20    C3 09 17                    JMP     GTRA            ;TRANSFER TO GOTO
                        ;
                        ; ROUTINE FCOMP COMPARES 2 FLOATING POINT #'S.  THEY ARE
                        ; TO BE IN FREG1 AND FREG2.
                        ; ALL REGISTERS ARE DESTROYED.
                        ; THE VALUE RETURNED IN REG A IS RESULT OF COMPARISON.
                        ; RESULTS ARE AS FOLLOWS:
                        ;
                        ;           A=1     =>      FREG1 < FREG2
                        ;           A=2     =>      FREG1 > FREG2
                        ;           A=4     =>      FREG1 = FREG2
                        ;
1D23    21 7F 21        FCOMP:      LXI     H,FREG1+3       ;PNTS TO CHAR OF 1ST
1D26    11 83 21                    LXI     D,FREG2+3       ;PNTS TO CHAR OF 2ND
1D29    7E                          MOV     A,M             ;GET  1 CHAR
1D2A    06 80                       MVI     B,200Q          ;MASK TO B
1D2C    A0                          ANA     B               ;GET SIGN,  1
1D2D    4F                          MOV     C,A             ;SAVE IN C
1D2E    1A                          LDAX    D               ;GET CHAR  2
1D2F    A0                          ANA     B               ;GET SIGN  2
1D30    A9                          XRA     C
1D31    CA 3B 1D                    JZ      SINEQ           ;SAME SIGNS
1D34    79                          MOV     A,C             ;OPPISITE SIGNS,GET  1
1D35    17                          RAL                     ;ROTATE TO CY
1D36    3E 01                       MVI     A,1
1D38    D8                          RC                      ;FREG1 < FREG2 => A=1
1D39    3C                          INR     A               ;ELSE FREG1 > FREG2
1D3A    C9                          RET                     ;AND A=2
1D3B    C5              SINEQ:      PUSH    B               ;SAVE SIGN
1D3C    2B                          DCX     H               ;PNTR TO  1 IN H,L
1D3D    2B                          DCX     H
1D3E    2B                          DCX     H
1D3F    43                          MOV     B,E             ;PNTR TO  2 IN B
1D40    05                          DCR     B
1D41    05                          DCR     B
1D42    05                          DCR     B
1D43    CD DF 0F                    CALL    LMCM            ;COMPARE MAGNITUDES
                        ; AT THIS POINT Z=1 => =, CY=1 => 1<2
1D46    C1                          POP     B               ;GET SIGN BACK
1D47    C2 4D 1D                    JNZ     $+6
1D4A    3E 04                       MVI     A,4             ;EQUAL => A=4
1D4C    C9                          RET
1D4D    79                          MOV     A,C             ;GET SIGN TO A
1D4E    3C                          INR     A               ;SET SIGN BIT
1D4F    3E 01                       MVI     A,1
1D51    FA 57 1D                    JM      $+6             ;SIGN IS NEGATIVE
1D54    D8                          RC                      ;SIGN=+ AND ABS(FREG1)<
1D55    3C                          INR     A               ;ABS(FREG1)>ABS(FREG2)
1D56    C9                          RET
1D57    D0                          RNC                     ;SIGN=- AND ABS(FREG1)>
1D58    3C                          INR     A               ;ABS(FREG1)<ABS(FREG2)
1D59    C9                          RET
                        ;
                        ; CALL PROCESSOR
                        ;
1D5A    21 8C 1B        CALLP:      LXI     H,IEND          ;INIT RETURN ADDRESS
1D5D    E5                          PUSH    H
1D5E    2A 5B 21                    LHLD    CPNT            ;INIT POINTERS
1D61    23                          INX     H
1D62    23                          INX     H
1D63    23                          INX     H
1D64    CD F6 15                    CALL    ICP7
1D67    7E                          MOV     A,M             ;GET CHAR
1D68    FE A8                       CPI     '('+200Q        ;IS IT A (?
1D6A    C2 E8 1B                    JNZ     ER7             ;BAD
1D6D    CD F6 15                    CALL    ICP7            ;BUMP PNTRS
1D70    CD 2A 12                    CALL    CVB             ;GET SUB
1D73    85                          ADD     L               ;UPDATA H,L
1D74    6F                          MOV     L,A
1D75    3E 00                       MVI     A,0
1D77    8C                          ADC     H
1D78    67                          MOV     H,A             ;D NOW CONTAINS SUB
1D79    E5                          PUSH    H               ;SAVE HL
1D7A    21 AC 21                    LXI     H,SUBS          ;GET START OF SUB TABLE
1D7D    7E              NUSUB:      MOV     A,M             ;GET ENTRY
1D7E    BA                          CMP     D               ;COMPARE
1D7F    CA 8E 1D                    JZ      FNDSB           ;FOUND IT
1D82    23                          INX     H               ;PNT TO NEXT ENTRY
1D83    23                          INX     H
```

```
1D84    23                          INX     H
1D85    3C                          INR     A               ;CHECK TO SEE IF LAST W
1D86    C2 7D 1D                    JNZ     NUSUB
1D89    3E 15                       MVI     A,15H           ;ER 15 - NO SUB BY THIS
1D8B    C3 BE 15                    JMP     ERROR
1D8E    23              FNDSB:      INX     H               ;FOUND IT,GET STARTING
1D8F    5E                          MOV     E,M
1D90    23                          INX     H
1D91    56                          MOV     H,M
1D92    6B                          MOV     L,E             ;AND SAVE IT
1D93    22 5D 21                    SHLD    SBSAV
1D96    2A 59 21                    LHLD    NXTSP           ;INIT MEMORY SCRATCH AR
1D99    22 88 21                    SHLD    MESCR
1D9C    E1                          POP     H               ;GET SOURCE PNTR BACK
1D9D    7E              PARLP:      MOV     A,M             ;GET CHAR
1D9E    FE A9                       CPI     ')'+200Q        ;IS IT )?
1DA0    CA CC 1D                    JZ      CLSUB           ;YES - GO CALL SUB
1DA3    FE AC                       CPI     ','+200Q        ;DO WE HAVE A ,?
1DA5    C2 2C 17                    JNZ     ER6             ;UEXPECTED CHARACTER
1DA8    CD F6 15                    CALL    ICP7            ;BUMP PNTRS
1DAB    CD 6D 18                    CALL    VAR             ;DO WE HAVE A VARIABLE
1DAE    D2 B5 1D                    JNC     PREXP           ;NO
1DB1    D5                          PUSH    D               ;YES - SAVE ADDRESS
1DB2    C3 9D 1D                    JMP     PARLP           ;CONTINUE
1DB5    CD C7 19        PREXP:      CALL    EVAL            ;EVALUATE EXPRESSION
1DB8    E5                          PUSH    H               ;SAVE H,L
1DB9    2A 88 21                    LHLD    MESCR           ;GET SCRATCH AREA
1DBC    CD A8 17                    CALL    COPDH           ;AND COPY TO IT
1DBF    D1                          POP     D               ;HL TO DE
1DC0    E5                          PUSH    H               ;SAVE ADDRESS
1DC1    23                          INX     H               ;UPDATE MESCR
1DC2    23                          INX     H
1DC3    23                          INX     H
1DC4    23                          INX     H
1DC5    22 88 21                    SHLD    MESCR           ;SAVE IT
1DC8    EB                          XCHG                    ;GET H,L BACK
1DC9    C3 9D 1D                    JMP     PARLP           ;CONTINUE
1DCC    2A 5D 21        CLSUB:      LHLD    SBSAV           ;START OF ROUTINE
1DCF    E9                          PCHL                    ;TRANSFER
                        ;
                        ; GOSUB PROCESSOR
                        ;
1DD0    21 8C 1C        GOSUB:      LXI     H,ILOOP         ;FOR RETURN STMT.
1DD3    E5                          PUSH    H               ;TO STACK
1DD4    2A 56 21                    LHLD    KFPNT           ;PNTR. TO NEXT STMT.
1DD7    E5                          PUSH    H               ;SAVE ON STACK
1DD8    2A 59 21                    LHLD    NXTSP           ;CHECK MEMORY
1DDB    CD 14 15                    CALL    MEMFUL
1DDE    2A 5B 21                    LHLD    CPNT            ;GET CHAR. PNTR
1DE1    23                          INX     H
1DE2    C3 03 17                    JMP     GSENT           ;PART OF GOTO TO FINISH
                        ;
                        ; RETURN STMT. PROCESSOR
                        ;
1DE5    E1              RETRN:      POP     H               ;GET RETURN ADD. FROM S
1DE6    C9                          RET                     ;CONTINUE
                        ;
                        ; FOR STATEMENT PROCESSOR
                        ;
1DE7    2A 5B 21        FOR:        LHLD    CPNT            ;FIX PNTRS
1DEA    0C                          INR     C
1DEB    23                          INX     H
1DEC    23                          INX     H
1DED    CD F6 15                    CALL    ICP7
1DF0    CD 22 12                    CALL    ALPHA           ;LETTER?
1DF3    D2 9A 1F                    JNC     ER21            ;NO
1DF6    46                          MOV     B,M             ;GET IT TO B
1DF7    CD F6 15                    CALL    ICP7            ;BUMP PNTR'S
1DFA    51                          MOV     D,C             ;SAVE C
1DFB    0E 00                       MVI     C,0             ;INIT C TO 0
1DFD    CD 14 12                    CALL    NUMB            ;NUMBER?
1E00    D2 09 1E                    JNC     $+9             ;NO
1E03    4E                          MOV     C,M             ;YES, GET IT
1E04    23                          INX     H               ;BUMP PNTR'S
1E05    15                          DCR     D
1E06    CA E8 1B                    JZ      ER7             ;PREMATURE EOL
1E09    E5                          PUSH    H               ;SAVE H,L
1E0A    CD 0D 16                    CALL    FSYM            ;GET VAR. LOCATION
1E0D    E3                          XTHL                    ;PUT ON STACK, GET H,L
1E0E    59                          MOV     E,C             ;VARIABLE TO D,E
1E0F    4A                          MOV     C,D             ;RESTORE C
1E10    50                          MOV     D,B
1E11    EB                          XCHG
1E12    22 8C 21                    SHLD    VNAME           ;SAVE VAR NAME
1E15    EB                          XCHG                    ;RESTORE H,L
1E16    7E                          MOV     A,M             ;LOOK FOR =
1E17    FE BD                       CPI     '=' OR 200Q
1E19    C2 81 1F                    JNZ     ER16
1E1C    CD F6 15                    CALL    ICP7            ;BUMP PNTR'S
1E1F    CD C7 19                    CALL    EVAL            ;EVALUATE EXPRESSION
1E22    E3                          XTHL                    ;VARIABLE LOCATION
1E23    CD A8 17                    CALL    COPDH           ;WRITE VALUE
1E26    22 8E 21                    SHLD    VLOC            ;SAVE PNTR TO VARIABLE
1E29    E1                          POP     H               ;GET H,L BACK
1E2A    79                          MOV     A,C             ;CHECK EOL
1E2B    B7                          ORA     A
1E2C    CA E8 1B                    JZ      ER7
1E2F    3E 02                       MVI     A,2             ;CHECK FOR 'TO'
1E31    CD F7 12                    CALL    SYMSRT
1E34    FE 07                       CPI     7
1E36    C2 86 1F                    JNZ     ER17
1E39    23                          INX     H               ;BUMB PNTR'S
1E3A    23                          INX     H
1E3B    79                          MOV     A,C             ;CHECK EOL
1E3C    B7                          ORA     A
1E3D    CA E8 1B                    JZ      ER7
1E40    CD C7 19                    CALL    EVAL            ;EVALUATE LIMIT
1E43    E5                          PUSH    H               ;SAVE H,L
1E44    21 90 21                    LXI     H,FLIMT         ;SAVE LIMIT VALUE
1E47    CD A8 17                    CALL    COPDH
1E4A    79                          MOV     A,C             ;CHECK EOL
1E4B    B7                          ORA     A
1E4C    C2 56 1E                    JNZ     STP
1E4F    11 D5 1E                    LXI     D,FONE          ;DEFAULT STEP=1
1E52    E1                          POP     H               ;RESTORE H,L
1E53    C3 6B 1E                    JMP     FBILD
1E56    E1              STP:        POP     H               ;GET H,L
1E57    3E 02                       MVI     A,2             ;LOOK FOR 'STEP'
1E59    CD F7 12                    CALL    SYMSRT
1E5C    FE 08                       CPI     8
1E5E    C2 86 1F                    JNZ     ER17
1E61    23                          INX     H               ;FIX H,L
1E62    23                          INX     H
1E63    23                          INX     H
1E64    0C                          INR     C               ;CHECK EOL
1E65    CD F6 15                    CALL    ICP7
1E68    CD C7 19                    CALL    EVAL            ;GET STEP SIZE
                        ;
                        ; AT THIS POINT:
                        ; VARIABLE NAME IS IN LOCATION VNAME
                        ; VARIABLE ADDRESS IS IN LOCATION VLOC
                        ; VARIBLE HAS BEEN INITIALIZED
                        ; LIMIT IS IN 4 BYTE LOCATION FLIMT
                        ; STEP IS POINTED TO BY D,E
                        ; H,L,C ARE POINTER, COUNTER AS USUAL
                        ;
1E6B    D5              FBILD:      PUSH    D               ;SAVE PNTR TO STEP
1E6C    2A 8C 21                    LHLD    VNAME           ;GET VARIABLE NAME
1E6F    3E 3F                       MVI     A,77Q           ;MASK
1E71    A4                          ANA     H               ;MASK OFF TOP 2 BITS
1E72    47                          MOV     B,A             ;SET UP TO CALL FSYM
1E73    4D                          MOV     C,L
1E74    CD 0D 16                    CALL    FSYM            ;FIND ENTRY
1E77    DA 8D 1E                    JC      FEXST           ;IT WAS THERE
1E7A    E5                          PUSH    H               ;IT WASN'T, SAVE H,L
1E7B    2A 59 21                    LHLD    NXTSP           ;UPDATE NXTSP
1E7E    3E 08                       MVI     A,8             ;ADD 8 TO H,L
1E80    85                          ADD     L
1E81    6F                          MOV     L,A
1E82    3E 00                       MVI     A,0
1E84    8C                          ADC     H
1E85    67                          MOV     H,A
1E86    22 59 21                    SHLD    NXTSP           ;NEW VALUE OF NXTSP
1E89    CD 14 15                    CALL    MEMFUL          ;CHECK MEMORY
1E8C    E1                          POP     H               ;GET ADD. IN DATA BLOCK
1E8D    D1              FEXST:      POP     D               ;ADDRESS OF STEP SIZE
1E8E    CD A8 17                    CALL    COPDH           ;STORE IT
1E91    23                          INX     H               ;PNT TO WHERE VAR. PNTR
1E92    23                          INX     H
1E93    23                          INX     H
1E94    23                          INX     H
1E95    3A 8E 21                    LDA     VLOC            ;FIRST BYTE
1E98    77                          MOV     M,A             ;STORE IT
1E99    23                          INX     H
1E9A    3A 8F 21                    LDA     VLOC+1          ;SECOND BYTE
1E9D    77                          MOV     M,A
1E9E    23                          INX     H               ;PNT TO WHERE LIMIT GOE
1E9F    11 90 21                    LXI     D,FLIMT         ;WHERE IT IS NOW
1EA2    CD A8 17                    CALL    COPDH           ;COPY IT
1EA5    23                          INX     H               ;PNT TO WHERE KFPNT GOE
1EA6    23                          INX     H
1EA7    23                          INX     H
1EA8    23                          INX     H
1EA9    3A 56 21                    LDA     KFPNT           ;1ST BYTE
1EAC    77                          MOV     M,A
1EAD    23                          INX     H
1EAE    3A 57 21                    LDA     KFPNT+1         ;2ND BYTE
1EB1    77                          MOV     M,A
                        ; PUT CURRENT VNAME ON NESTING STACK
1EB2    21 00 00                    LXI     H,0             ;GET STACK-POINTER
1EB5    39                          DAD     SP
1EB6    22 8F 21                    SHLD    VLOC            ;SAVE IT
1EB9    2A 94 21                    LHLD    NEST            ;GET NEST SP
1EBC    7D                          MOV     A,L             ;COMPARE WITH STACK LIM
```

```
1EBD    FE 96                  CPI     TOPNS AND 377Q   ;NEED ONLY CCMPARE PAGE
1EBF    CA 8B 1F               JZ      ER18             ;FOR'S NEXTED TOO DEEPL
1EC2    F9           NSTOK:    SPHL                     ;LOAD NEW SP
1EC3    EB                     XCHG                     ;SAVE NEST SP
1EC4    2A 8C 21               LHLD    VNAMF            ;GET INDEX NAME
1EC7    E5                     PUSH    H                ;SAVE IT
1EC8    1B                     DCX     D                ;UPDATE NEST SP
1EC9    1B                     DCX     D
1ECA    EB                     XCHG                     ;SAVE IT
1ECB    22 94 21               SHLD    NEST
1ECE    2A 8E 21               LHLD    VLOC             ;RESTURE OLD SP
1ED1    F9                     SPHL
1ED2    C3 8C 1B               JMP     IEND             ;ALL DONE
1ED5    80 00 00 01  FONE:     DB      200Q,0,0,001Q    ;FLOATING PNT ONE
                     ;
                     ; NEXT STATEMENT PROCESSOR
                     ;
1ED9    2A 5B 21     NEXT:     LHLD    CPNT             ;FIX PNTR'S
1EDC    23                     INX     H
1EDD    23                     INX     H
1EDE    23                     INX     H
1EDF    0C                     INR     C
1EE0    CD F6 15               CALL    ICP7
1EE3    CD 22 12               CALL    ALPHA            ;LETTER?
1EE6    D2 9A 1F               JNC     ER21             ;NO, ERROR
1EE9    46                     MOV     B,M              ;YES, GET IT
1EEA    51                     MOV     D,C              ;SAVE C
1EEB    0E 00                  MVI     C,0              ;INIT C TO 0
1EED    23                     INX     H                ;BUMP PNTR'S
1EEE    15                     DCR     D
1EEF    CA FD 1E               JZ      NEXT1
1EF2    CD 14 12               CALL    NUMB             ;NUMBER?
1EF5    D2 9A 1F               JNC     ER21             ;NO, ERROR
1EF8    4E                     MOV     C,M              ;YES, GET IT
1EF9    15                     DCR     D                ;SHOULD BE ECL
1EFA    C2 9A 1F               JNZ     ER21
1EFD    21 00 00     NEXT1:    LXI     H,0              ;GET SP
1F00    39                     DAD     SP
1F01    22 8E 21               SHLD    VLOC             ;SAVE IT
1F04    2A 94 21               LHLD    NEST             ;GET NEST SP
1F07    7D                     MOV     A,L              ;COMPARE WITH BOTTOM
1F08    FE AA                  CPI     BOTNS AND 377Q
1F0A    CA 90 1F               JZ      ER19             ;NEXT BEFORE FOR
1F0D    F9                     SPHL                     ;LOAD SP
1F0E    F1                     POP     H                ;GET LAST INDEX
1F0F    78                     MOV     A,B              ;CCMPARE TC CURRENT
1F10    BC                     CMP     H
1F11    C2 95 1F               JNZ     ER20             ;NESTING ERROR
1F14    79                     MOV     A,C
1F15    BD                     CMP     L
1F16    C2 95 1F               JNZ     ER20
1F19    2A 8E 21               LHLD    VLOC             ;ALL OK, RESTORE OLD SP
1F1C    F9                     SPHL
1F1D    3E 3F                  MVI     A,77Q            ;MASK
1F1F    A0                     ANA     B                ;MASK OUT TOP 2 BITS
1F20    47                     MOV     B,A
1F21    CD 0D 16               CALL    FSYM             ;FIND SYMBCL
1F24    EB                     XCHG                     ;ADDRESS TO D,E
1F25    21 7C 21               LXI     H,FREG1          ;COPY STEP TO FREG1
1F28    CD A8 17               CALL    COPDH
1F2B    13                     INX     D                ;PNT TO CHARACTERISTIC
1F2C    13                     INX     D
1F2D    13                     INX     D
1F2E    1A                     LDAX    D                ;GET IT
1F2F    E6 80                  ANI     200Q             ;GET SIGN
1F31    17                     RAL                      ;ROTATE IT INTO CARRY
1F32    3F                     CMC                      ;COMPLEMENT IT
1F33    3E 00                  MVI     A,0              ;MAKE SURE A=0
1F35    17                     RAL                      ;ROTATE TO LSB
1F36    3C                     INR     A                ;BUMP BY CNE
1F37    32 8E 21               STA     VLOC             ;SAVE IT, ITS =1 IF - S
1F3A    13                     INX     D                ;PNT TO VARIABLE PNTR
1F3B    EB                     XCHG                     ;GET IT TO DE
1F3C    5E                     MOV     E,M
1F3D    23                     INX     H
1F3E    56                     MOV     D,M
1F3F    23                     INX     H
1F40    E5                     PUSH    H                ;SAVE DATA BLOCK PNTR.
1F41    21 80 21               LXI     H,FREG2          ;COPY VARIBLE VALUE TO
1F44    CD A8 17               CALL    COPDH            ;SAVE VARIABLE LOCATION
1F47    EB                     XCHG
1F48    3E 02                  MVI     A,2              ;SET UP TO ADD
1F4A    CD 3B 1A               CALL    BINOP            ;AND DO IT
1F4D    CD A8 17               CALL    COPDH            ;COPY TO VARIABLE
1F50    21 7C 21               LXI     H,FREG1          ;AND TO FREG1 FOR COMPA
1F53    CD A8 17               CALL    COPDH
1F56    D1                     POP     D                ;PNT TO LIMIT
1F57    21 80 21               LXI     H,FREG2          ;COPY TO FREG2
1F5A    CD A8 17               CALL    COPDH
1F5D    D5                     PUSH    D                ;SAVE DATA BLOCK PNTR
1F5E    CD 23 1D               CALL    FCOMP            ;COMPARE
1F61    21 8E 21               LXI     H,VLOC           ;COMPARE WITH STEP TYPE
1F64    BE                     CMP     M
1F65    E1                     POP     H                ;GET DATA BLCCK PNTR.
1F66    CA 74 1F               JZ      NXTDN            ;YES => LOOP DONE
1F69    23                     INX     H                ;LOOP NOT DONE
1F6A    23                     INX     H                ;PNT TO TRANSFER ADD.
1F6B    23                     INX     H
1F6C    23                     INX     H
1F6D    5E                     MOV     E,M              ;GET IT TO H,L
1F6E    23                     INX     H
1F6F    56                     MOV     D,M
1F70    EB                     XCHG
1F71    C3 8C 16               JMP     ILOOP
1F74    21 94 21     NXTDN:    LXI     H,NEST           ;POP NEST STACK
1F77    34                     INR     M
1F78    34                     INR     M
1F79    C3 8C 1B               JMP     IEND             ;CONTINUE
1F7C    3E 09        ER9:      MVI     A,9
1F7E    C3 BE 15               JMP     ERROR            ;'=' EXPECTED(NOTE: NO
1F81    3E 16        ER16:     MVI     A,16H            ;FOR INDICES)
1F83    C3 BE 15               JMP     ERROR            ;BAD SYNTAX NEAR 'TO' O
1F86    3E 17        ER17:     MVI     A,17H            ;IN FOR STATEMENT
1F88    C3 BE 15               JMP     ERROR            ;FOR'S NESTED TOO DEEPL
1F8B    3E 18        ER18:     MVI     A,18H
1F8D    C3 BE 15               JMP     ERROR            ;'NEXT' EXECUTED BEFORE
1F90    3E 19        ER19:     MVI     A,19H
1F92    C3 BE 15               JMP     ERROR            ;NESTING ERRCR, 'FOR'-'
1F95    3E 20        ER20:     MVI     A,20H
1F97    C3 BE 15               JMP     ERROR            ;BAD INDEX IN FOR-NEXT
1F9A    3E 21        ER21:     MVI     A,21H
1F9C    C3 BE 15               JMP     ERROR
                               END
NO PROGRAM ERRORS

                          SYMBOL TABLE

* 01
A       0007      ADDD    1A56      ADFLD   1812      AFUND   190F
AGA     1A02      ALPHA   1222      AR      1614      ARCK    18B9
ARYES   18CD      ASBC    1A59      B       0000      BAC     1220
BINOP   1A3B      BMPTR   19BB      BND1    156C *    BND2    15A3
BND3    15A6      BUTNS   21AA      BOUND   154D      BUPT    1890 *
C       0001      C1      1219      CALLP   1D5A      CHAR2   0FF7
CHARS   1508      CHEKE   19B1      CHK1    12A3      CHKMD   1987
CLSUB   1DCC      CONT    177B      CONV    0FE5      CUPD1   17AE
COPDH   17A8      CPNT    215B      CREG    2184      CVB     122A
CVB1    124B      CVB2    1268      D       0002      DCOMP   0FF1
DFXL    0FDC      DIM     1720      DIV     1A8D      DLOOP   1726
DOL     19EC      F       0003      ECAV    19D1      ENDD    16F2
ENTRY   1671      EOK     19FA *    ER10    1C78      ER14    1CA3
ER16    1F81      ER17    1F86      ER18    1F8B      ER19    1F90
ER20    1F95      ER21    1F9A      ER6     172C      ER7     1BE8
ER8     1A22      ER9     1F7C      ERLN    15DF      ERRET   1BC8
ERROR   15BE      ERRR1   1502      EVAL    19C7      EXPRE   1AD6
F1      144E      FALSE   1D12      FBAC    1676      FBILD   1E6B
FCOMP   1D23      FDAT    196F      FEXST   1E8D      FINPT   0FE8
FIX     1915      FLDFD   1B2D      FLIMT   2190      FMULT   1A87
FNDSB   1D8E      FONE    1ED5      FOR     1DE7      FOR10   1441
FOR11   1440      FOR12   143F      FORM1   1448      FORM2   1447
FORM3   1446      FORM4   1445      FORM5   1449      FORM6   144F
FORM7   1444      FORM8   1443      FORM9   1442      FOUND   12A0
FPERR   1A5B      FREG1   217C      FREG2   2180      FSYM    160D
FWAM    21AA      GET     17D9      GKEY    16C6      GOOD    1932
GOSUB   1DD0      GOTO    1700      GREG    2177      GSENT   1703
GTRA    1709      H       0004      HLINP   2186      HOME    1824
HSRIN   14D5      IAGA    1C8D      IBUF    2101      ICP2    1605
ICP4    1600      ICP7    15F6      ICP8    15FB      IDONE   1983
IEND    1B8C      IFRT    1C7D      II1     1CA8      ILOOP   168C
INCPT   1607      INP     1973      INPER   1B50      INPOK   1B89
INPUT   1B4C      INSER   108F      ISR10   116B      ISR11   1171
ISR12   117A      ISR1A   10A2      ISRT1   10BD      ISRT2   10C4
ISRT3   10D4      ISRT4   10D7      ISRT5   110F      ISRT6   1121
ISRT7   1144      ISRT8   1148      ISRT9   1152      JTBL    16D6
KASE    2160      KDAT1   134F      KDAT2   1364      KDAT3   139E
KDAT4   13C0      KDATA   134B      KFPNT   2156      KL2     2154
KL4     2156      KL6     2158      KLEN    2158      KLINE   2152
KONT    183B      L       0005      L1      15AE      L2      128B
LADD    0FD0      LDIV    0FD6      LEN     2161      LENGT   1271
LET     1BDD      LIS1    11EA      LIST    11D1      LMCM    0FDF
LMUL    0FD3      LOK     1BED      LPNT    2152      LSUB    0FD9
LUKON   1627      M       0006      M1      1000      M1A     1013
M2      1016      M3      101F      M4      1042      M4A     1047
MCHK    0FF4      MDBC    1A90      MEMEN   27FF      MEMFU   1514
MEMST   2100      MESCR   2188      MINSE   1930      MODE    2185
MODE1   1997      MORE    187F      MULT    0FEB      MULT1   2162
MULT2   2164      NDOU    175F      NEST    2194      NEXT    1ED9
NEXT1   1EFD      NFPER   1A7E      NL2     214D      NL4     214F *
NL6     2151      NLE1    1275      NLE2    1284      NLINE   2148
NOENT   1646      NOLIN   1191      NOMAT   163B      NOSYM   1644
NOT3    1D08      NSRCH   1288      NSTOK   1EC2 *    NUMB    1214
NUSUB   1D7D      NXTDN   1F74      NXTSP   2159      OBUFF   2100
ODA10   14CB      ODAT1   1464      ODAT2   146A      ODAT3   1478
ODAT4   1481      ODAT5   148B      ODAT6   1491      ODAT7   14A0
ODAT8   1487      ODAT9   14C2      ODATA   145A      OKK     1847
OKLET   1731      OKN     1715      QNWD    1B25 *    OUTR    178B
```

```
P1      12B8      P2      12C2      PAD     12AD      PARLP   1D9D
PEND    1B46      PIN1    14DB      PIN1A   14DE      PIN3    1500
PINST   1C73      PK1     1418      PK2     1435      PL2     215B
PL4     215D      PL6     215F      PLINE   2159      PLOOP   1AAE
PREXP   1DB5      PRI     1A95      PRI1    1C63      PRMPT   185C
PRTIT   1AED      PSW     0006      PT1     106D      PTAPE   1065
PTFIN   1C4C      PTVAL   0FEE      QCHEK   1AC8      QOTOK   1ABD
QSTR1   1430      QSTRG   142B      QTCHK   120E *    QUITT   1208
QUOTE   1AB4      RDKON   1853      RELAT   1CD5      RETRN   1DE5
RINST   1836      RUN     1678      S2      1307 *    S3      1308
S4      1323      S4A     1322      S5      133F      SAVV    1C32
SBSAV   215D      SC1     1876      SCALR   18B1      SCMEM   21AD
SCOLN   1B07      SCR     2166      SEND    0006      SFSG    1894
SINEQ   1D3B      SLOAD   18AA      SONWD   1B39      SORCE   169A
SP      0006      SPACE   1981      STAC    2196      STLIN   2149
STNER   1BCB      STOKK   1BAF      STGKV   1BA0      STP     1E56
STPNT   118B      STRIN   1B92      STSIZ   0014      STSPA   214B
SUBB    1A81      SUBS    21AC      SYMSR   12F7      TAPE    11AA
THEN    1D19      TIN1    13CC      TIN2    13F6      TIN4    1408
TIN4A   140B      TIN5    1401      TIN5A   1404      TOPNS   2196
TRUE    1D16      TTYIN   13C9      V1      1815      VALUE   17C3
VAR     186D      VARAD   218A      VEND    21AA      VLOC    218E
VNAME   218C      W1      12DF      W2      12F0      WFOR    1A27
WFPER   1A77      WHAT    119E      WRIT    12D0      WRIT1   12D2
ZROL    0FCD
```

BACK ISSUES

While Supply Lasts!!

APRIL 1976
MAY 1976
AUGUST 1976
OCTOBER 1976
NOVEMBER 1976

Check our distributors first! If not in stock then send $1.75 plus .50 postage and handling for **each** issue you require. All months of issue not listed are no longer available.

SEND TO

McPheters, Wolfe & Jones
"BACK ISSUES"
P.O. Box 1234
Cerritos, CA 90701

INTERFACE AGE IS SEEKING ARTICLES

Here is the opportunity for Computer Clubs to benefit. For each article accepted from a club member and containing an additional cover letter signed by the club's Secretary, INTERFACE AGE will pay a **$25.00 bonus** to the club's treasury over and above the customary author's honorarium and the club will be included in the by line.