IMSAI

8K BASIC

Version 1.4

81-0000036 Rev. B

IMSAI 8K BASIC
Version 1.4

I. Loading

IMSAI 8K BASIC is available on three media: paper tape, Tarbell format cassette, and EPROM.

The paper tape and cassette versions require a minimum of 12K of RAM, installed at contiguous addresses from zero up. More RAM will permit longer BASIC programs and/or more variables.

A. Loading the Paper Tape Version

Use the IMSAI Paper Tape Loader, as supplied with your IMSAI 8080, to load the paper tape. Follow the instructions for loading paper tapes in the User Manual (Chapter 12, System Software). For the convenience of those who do not have the IMSAI Paper Tape Loader, a copy of its documentation is included with each order for BASIC on paper tape.

If your memory has WRITE PROTECT switches, turn the first 8K bytes off.

B. Loading the Cassette Version

Use the IMSAI Tape Cassette Bootstrap Loader (PGM-5A) to load the cassette. Follow the instructions supplied with the loader. A copy of the loader documentation is included with the cassette as an aid to those who do not have the IMSAI loader.

Alternately, IMSAI's Tape Cassette Operating System (PGM-2A) may be used to load the BASIC cassette. In this case, start execution with the TCOS command EXEC 0.

If your memory has WRITE PROTECT switches, turn the first 8K bytes off.

C. Installing the EPROM Version in Your System

Insert the 32 chips in two IMSAI PROM 4 boards, taking care that each chip has pin 1 up and is in the correct socket for its address.

Jumper the boards for addresses 0 and 1000 hex and install them in the machine.

Install RAM starting at address 2000 hex. 4K of RAM is adequate for many programs; 8K BASIC will make use of as much RAM as is installed at contiguous addresses starting at 2000 hex.

II. Starting BASIC

Hit STOP, RESET, RUN. (If the loader started BASIC, skip this step.)

BASIC should type "READY".

Hit the CR key and then type NEW CR. This step is essential when BASIC is first started.

III. Restarting BASIC

If the system gets hung up and will not respond to control–C, try restarting BASIC.

Do this by hitting STOP, RESET, RUN in the front panel.

Your program should still be in memory and may be LISTed, modified, or RUN.

IV. Entering a Program from the Keyboard

Type the command NEW to delete the previous program.

Enter the program one line at a time, typing a CR after each line.

Each statement must be numbered, but they need not be typed in order. The program will be listed or executed in line number order. It is suggested that successive line numbers differ by 10 or more to permit inserting lines later.

A line may be deleted by typing its line number followed by a CR.

A line may be replaced by typing the same line number followed by the updated statement and CR.

A line may be inserted by typing a new, unique line number, followed by the statement and CR.

A line may be retyped if a mistake is noticed before the CR is typed by typing control–U.

The last character may be deleted by typing the RUBOUT key. A back slash, the deleted character, and another back slash will echo. On a CRT terminal, you may delete the previous character by typing control–H which will erase the previous character and back the curser up one position. Now type the correct character and continue.

V. Entering a Program from Paper Tape

Type the command NEW to delete the previous program.

Mount the paper tape in the Teletype reader starting anywhere in the leader.

Type the command TAPE.

Manually start the reader, if necessary.

If the tape was not punched by this BASIC, after it has been read, stop the reader and type KEY CR.

VI. Listing the Program

Type the command LIST CR.

To list a single statement, type LIST line number CR.

To list a portion of the program, type LIST line number, line number CR.

To interrupt a LIST in progress, type control–C.

VII. Punching the Program on Paper Tape

Type SAVE, but do not type the CR.

Turn on the teletype punch.

Hit CR.

The program will be punched and listed.

Turn off the punch.

VIII. Running the Program

Type the command RUN.

The program will run to completion until an error is encountered or until a stop statement is executed.

The program may be terminated by typing control–C.

INPUT statements will prompt with a question mark. Type the value and a CR. Multiple values may be input separated by commas.

IX. Error Messages

BASIC error messages are of the form: XX ERR @ 9999 where XX is the error code

and 9999 is the line number. (Immediate commands print no line number.)

The following errors terminate program execution:

Code	Meaning
DA	Out of data. A READ statement tried to read another DATA value and there were no more DATA statements.
NX	NX with no matching FOR or more than 8 nested FORs.
UL	Unidentified line number in IF...THEN, GOTO, or GOSUB statement.
RT	RETURN encountered and no GOSUB in effect.
OF	Fatal overflow: division by zero, attempt to take log of a number less than or equal to zero, or square root of a negative number (other overflow errors are non-fatal; see below).
SN	Syntax error: unidentified statement or incorrectly formed statement; BASIC can't understand what you're saying.
ST	Execution stack (internal) error. Either: 1) a NEW command was not given after power-up; 2) too few or too many arguments for a function; or 3) some syntax errors give the "ST" error code.

The following errors are non-fatal; i.e., a message is printed and program execution continues.

Code	Meaning
UN	Underflow: number whose absolute value is less than 2.71050 E-20 and greater than zero.

OV	Overflow: number whose absolute value is greater than 5.76461 E18 (slightly higher values may be reached using addition or subtraction).

X. Debugging

After correction, the program may be restarted with the command GOTO 9999 where 9999 is the line number given in the message, or with the CON command.

The program may be stopped by typing control–C.

Variables may be inspected with the PRINT (or ?) command.

Variables may be modified with the LET or INPUT commands.

Output may be suppressed without stopping the program by typing control–O. Another control–O will turn printout on again.

XI. References

This manual is not intended to be an introduction to the BASIC language. Users with no familiarity with BASIC are referred to the following books, all available from IMSAI:

My Computer Likes Me When I Speak BASIC, by Bob Albrecht
An introduction to BASIC for those with no previous computer experience.

BASIC–Plus Manual, Digital Equipment Corp.
User manual for a minicomputer BASIC; 8K BASIC is similar to language described in first few sections.

What To Do After You Hit Return, People's Computer Company
A book of game programs written in BASIC.

XII. IMSAI BASIC Language

Notation: In the following two sections

CAPITAL letters represent words that are used verbatim.

Lower case words are representative, for instance, "variable" means "X", "Y1", "Q9$" or any legal BASIC variable name.

[square brackets] enclose optional items.

"..." means the variable or expression may be repeated any number of times.

Don't forget the commas (or semi-colons) between the repetitions.

Statements:

[LET] variable=expression

DIM variable (dimensions)
for numeric variables only; should appear before variable is used.

PRINT expression, expression ...
prints values at 14 column tab stops

PRINT expression; expression ...
prints without space between

PRINT
alone prints carriage return

All PRINTouts end with a carriage return unless the statement contained a comma or a semi-colon after the last expression.

?
Same as PRINT

INPUT variable, variable ...
prompts with "?" and accepts typed-in numbers or strings. If "text" is used in place of a variable, it is typed out.

REM any text

!
Same as REM

IF expression relop expression THEN line number or statement
"relop" can be <, =, >, <> (not equal), >=, <=.

FOR expression=expression TO expression [STEP expression]

NEXT variable

GOTO line number

GOSUB line number

INPUT LINE string variable
Accepts typed-in text terminated by a carriage return. Any quotes, commas or spaces in the text are transmitted.

RETURN

ON expression GOTO line number, line number ...

ON expression GOSUB line number, line number ...

DATA value, value ...

RESTORE

READ variable, variable ...

DEF FN variable (variable(s))=expression

CHANGE array name=string variable
CHANGE string variable=array name

STOP
stops execution; program can be restarted with CON

END
optional; stops execution, program cannot be restarted with CON

CALL expression
calls user's machine language routine at address given by expression

POKE address, value
stores 8-bit value into memory; e.g., tor transmitting data to machine-language routing.

OUT port, value
outputs value to port for accessing non-standard I/O devices

RANDOMIZE

Expressions

Variables consist of a letter or a letter and a digit, followed by a dollar sign for string-valued variables.

Number range is 2.7105E-20 to 5.7646E18.

Numeric operators available are +, -, *, / and ↑ (raise to a power). On some keyboards ↑ is ^.

Strings may be 0 to 238 characters long. String constants (literal texts) may be enclosed in single or double quotes.

The "+" operator concatenates strings.

Functions Available:

All arguments may be expressions. Those ending in "$" are string-valued.

ABS (x)	Absolute value
SQR (x)	Square root
INT (x)	Largest integer <= n
SGN (x)	Sign of n: -1, 0 or 1
RND[(x)]	Random number between 0 and x. If (x) is omitted, range is 0 and 1.
SIN (x)	
COS (x)	
TAN (x)	
ATN (x)	
LN (x)	Natural logarithm
LOG (x)	Base 10 logarithm
EXP (x)	
PEEK(address)	Contents of memory location
PI	3.14159
LEN(string)	Length
INSTR (x,string1, string2)	Position of string2 in string1 starting at position n; 0 if not found
ASCII (string)	ASCII value of character
CHR$ (x)	String consisting of character with ASCII value x
STRING$(x,y)	String of x characters with ASCII value y
NUM$(x)	Converts number to string of digits
VAL(string)	Numeric value of string (gives CV error if not numeric)
SPACE$(x)	String of x spaces
LEFT$(string,n)	Characters 1 through n of string
RIGHT$(string,n)	Characters n through the end of string
MID$(string,m,n)	n characters of string starting at character m.
POS(x)	Carriage position. First column is 0, x is ignored
TAB(x)	Use only with PRINT. Moves carriage to column x.
INP(port)	Input from I/O port

Multiple statements on one line are allowed if separated by colons.

XIII. Immediate Commands

The following may be typed without a line number for immediate execution.

LIST	
RUN	
XEQ	Like RUN but does not delete data
GOTO	
NEW	
NEW*	Deletes program without deleting data, for chaining
TAPE	
SAVE	
KEY	
FREE	
CON	Continue from STOP, control–C or after correcting an error
IF	Typed without the THEN clause to test variables. For example: IF A=B CR. If you get, UL ERR @ LINE then the relation is true. Otherwise, its false.

In addition, almost all BASIC program statements, if typed without a line number, will be executed immediately.

XIV. Chaining

Some applications involve a program too large to fit the available memory. Techniques described here permit "chaining" multiple programs while transmitting variable values from one to the next.

Load and RUN the first program in the normal manner.

Use "NEW*" rather than "NEW" to delete the program without deleting its data.

Enter or load the next program in the normal manner.

Use XEQ rather than RUN to start the program without clearing the data area.

Repeat the preceding three steps for each additional program.

XV. Compatibility

A. With BASIC-Plus

Line numbers range from 1 to 9999 only.

Subscripts not allowed on string variables.

Number range: 2.7105-E20 to 5.7646E18.

Only the statements, operators, and functions listed above are available.

B. With Previous Release 1.3

Log function gives Base 10 logarithm rather than natural logarithm.

New function LN gives natural logarithm.

PRINT ... TAB(X) ...: If X is less than column position goes to a new line. Old version continued at current column.

XVI. Known Problems

"FRE" returns a number that includes storage actually in use for variables.

If program size exceeds memory available, BASIC bombs out.

If data (variable values) exceed size of memory not used by program, program and possibly interpreter is destroyed.

If program jumps (with a GOTO) out of a FOR loop before the loop has completed then later executes the same FOR statement, BASIC gets confused.

There is no provision for reserving memory for machine language routines other than by addressing an additional board at a non-contiguous address.

Strings over about 238 characters cause a variety of problems including interfering with the current FOR loop and destroying the BASIC interpreter.

```
; BASIC30.ASM   1.4      05/19/77        JRB      8K BASIC
; BASICS2.ASM   1.401    05/11/77        DK       8K BASIC
; BASIC19.ASM   1.401    05/11/77        DH
; BASIC18.ASM   1.401    05/10/77        JRB
; BASIC16.ASM   1.401    05/09/77        DH
; BASIC11.ASM   1.401    05/04/77        DH
; BASIC10.ASM   1.401    05/03/77        DH
; BASIC8.ASM    1.401    05/02/77        DH

; IMSAI 8K-9K BASIC
;
; COPYRIGHT (C) 1977
; IMSAI MANUFACTURING CORPORATION
; 14860 WICKS BLVD, SAN LEANDRO CALIFORNIA  94577

; CORRECTION HISTORY:
;
;   02/25/77 - FIXED BEGPR POINTERS
;            - FIXED LOG(X) FOR 0.5 < X < 1.0
;            - FIXED SQR(X) FOR 0.0 < X < 0.5
;            - FIXED SCI NOTATION INPUT ROUTINE
;            - FIXED EDIT ROUTINE WHEN PROGRAM ENDS ON
;              00 BOUNDARY (SYSTEM USED TO GO AWAY)
;            - ADDED XEQ COMMAND (LIKE RUN BUT KEEPS DATA)
;            - SOFTWARE MEMORY PROTECT OF 1ST 9K IMPLIMENTED
;            - FIXED TAB FOR BACKWARDS MOVEMENT
;            - FIXED OV ERROR FOR SMALL X IN TRIG,LOG & EXP
;            - ADDED PROGRAM CHAINING CAPABILITY.
;            - FIXED EXP(X) ROUTINE FOR LARGE X.
;            - ADDED PEEK(X) COMMAND
;            - ADDED POKE A,X COMMAND
;            - ADDED CALL A COMMAND
;  04/02/77  - ADDED TARBEL CASSETTE SAVE AND LOAD
;            - ADDED FIX LINE EDITOR
;            - RENAMED NATURAL LOG TO LN(X)
;            - ADDED BASE 10 LOG AS LOG(X)
;            - ALLOWED FOR DAZZLER IN OUTPUT ROUTINE
;            - ADDED LINE # SEARCH UTILITY (LOCAT EQU $)
;            - ADDED TABLE SEARCH UTILITY (SEEK EQU $)
;            - ARRAYS CAN NOW HAVE > 256 ELEMENTS PER DIM
; 04/09/77   -ADDED CONDITIONAL ASSY PARMS FOR 8 AND 9K
;            -FIXED POWER ERROR. (X↑B WHEN B=0 GAVE X↑2.)
;            -ADDED CONTROL H AS PHYSICAL RUBOUT OF CHAR
; 04/27/77   -CHANGE RST'S TO RUN UNDER CP/M
;            -ADDED EXPRESSION EVALUATER FIX
;            -LOAD UNDER CPM
; 05/02/77   -ADD DDT, BYE COMMANDS, BIOS I/O
; 05/03/77   -OPTIMIZE FUNCTION ITERATION LOOP (SIN5)
;            -SO UNDERFLOW CAN BE MADE NON-FATAL
; 05/04/77   -OPTIMIZE SIN(X) ROUTINE
;            -ADD NON-FATAL ERRORS
; 05/09/77   -SQUISH TO INCLUDE PEEK,POKE,CALL IN 8K
; 05/11/77   -MAKE RND(X) USE X AS RANGE; X↓0->1,↓X->0
;            -TAB(N) GO TO NEXT LINE IF PAST POSITION
; 5/12/77   - BUG IN NESTED FOR'S AND REENTERED FOR'S FIXED
;
```

```
                ; ASSEMBLY PARAMETERS:
0000 =                  LARGE   EQU     0       ;-1=9K ASSEMBLY, 0=8K
0000 =                  CPM     EQU     0       ;-1=RUN UNDER CPM
0000 =                  HUNTER  EQU     0       ;-1= INCLUDE BAUD COMMAND


                ; CPM EQUATES

0000 =                  BOOT    EQU     0       ;WARM BOOT
0005 =                  BDOS    EQU     5       ;BDOS ENTRY
0100 =                  TBASE   EQU     100H    ;PROGRAM LOAD UNDER CPM
0003 =                  CSTAT   EQU     3       ;OFFSET OF CONSOLE STATUS
                                                ;...QUERY IN BIOS TABLE


                ; BASIC EQUATES

00F7 =                  FATAL   EQU     0F7H    ;CODE FOR FATAL IS RST 6


                BASIC:  IF      NOT CPM
0000                    ORG     0
0000 210024             LXI     H,RAM+1024
0003 3EAE               MVI     A,0AEH  ;START OF INIT SEQUENCE
0005 C38100             JMP     INIT1   ;FINISH INIT
                        ENDIF

                        IF      CPM
                        ORG     TBASE
                        JMP     INITC   ;USE TEMPORARY CODE AT END
                        ENDIF

                ;
                ;       ORG     8
                ;
                ; SKIP CHARS POINTED BY H,L UNTIL NON-BLANK,
                ; LEAVE IN REG A
                ;
0008 7E         RST1:   MOV     A,M     ;LOAD THE BYTE AT (H,L)
0009 FE20               CPI     ' '     ;TEST IF BLANK
000B C0                 RNZ             ;RETURN IF NOT
000C 23                 INX     H       ;POINT NEXT
000D C30800             JMP     RST1    ;LOOP
                ;
                ;
                ;       ORG     16
                ;
                ; COMPARE STRING AT (H,L) TO STRING AT (D,E)
                ; RETURN IF EQUAL (THRU X'00' IN D,E) OR ON FIRST NOT EQUAL
                ; ONLY THE FIRST THREE CHARS NEED BE EQUAL
                ; IGNORE ALL SPACES
                ;
0010 C5         RST2:   PUSH    B       ;SAVE B,C
0011 0600               MVI     B,0     ;INIT COUNT
0013 CF         COMP1:  RST     1       ;SKIP SPACES
0014 1A                 LDAX    D       ;GET CHAR TO MATCH WITH
0015 C3791A             JMP     COMP2   ;CONTINUE ELSEWHERE
                ;
```

```
                        ;
                        ;       ORG     24
                        ;
                        ; STORE THE FLOATING POINT ACCUMULATOR AT (H,L)

0018 115822     RST3:   LXI     D,FACC  ;POINT FLOAT ACC
001B 0604               MVI     B,4     ;BYTE COUNT
001D C34D1C             JMP     COPYD   ;GO MOVE IT


                        ;       ORG     32

                        ; INCREMENT H,L BY BYTE AT (SP), RETURN TO (SP)+1

0020 E3         RST4:   XTHL            ;GET RETURN ADDRESS IN H,L
0021 7E                 MOV     A,M     ;GET THE INCREMENT
0022 23                 INX     H       ;POINT TRUE RETURN
0023 E3                 XTHL            ;PUT BACK TO STACK
0024 D5                 PUSH    D       ;SAVE D,E
0025 C33B00             JMP     RST4A   ;CONTINUE

                        ;
                        ;
                        ;       ORG     40
                        ;
                        ; LOAD THE FLOATING POINT ACCUM WITH THE 4 BYTES AT (H,L)
                        ;
0028 115822     RST5:   LXI     D,FACC  ;POINT FLOAT ACC
 02B 0604               MVI     B,4     ;BYTE COUNT
002D C3581C             JMP     COPYH   ;GO MOVE IT
                        ;
                        ;
                        ;       ORG     48
                        ;
                        ; PRINT:  'XX ERR @ NNN'
                        ; **** IF ERROR MESSAGE CHANGES TO A DIFFERENT RST,
                        ; **** ...CHANGE "FATAL" EQUATE

0030 E3         RST6:   XTHL            ;SAVE HL, GET ERROR CODE PTR
0031 F5                 PUSH    PSW     ;SAVE REGS
0032 D5                 PUSH    D
0033 C5                 PUSH    B
0034 C3311C             JMP     ERROR   ;CONTINUE


                        IF NOT CPM
003B                    ORG     59      ;LEAVE 3 BYTES FOR DDT
                        ENDIF

003B 5F         RST4A:  MOV     E,A     ;PUT IN LOW
003C B7                 ORA     A       ;TEST SIGN
003D 1600               MVI     D,0     ;DEFAULT POSITIVE
003F F24400             JP      RST4B   ;BRIF +
0042 16FF               MVI     D,0FFH  ;ELSE, NEG
0044 19         RST4B:  DAD     D       ;BUMP H,L
0045 D1                 POP     D       ;RESTORE D,E
0046 C9                 RET             ;RETURN
                        ;PAGE
```

```
0047 434F505952          DB      'COPYRIGHT (C) 1977 '
005A 494D534149          DB      'IMSAI MFG CORP '
0069 53414E204C          DB      'SAN LEANDRO CA 94577 USA'


                ; INITIALIZATION ROUTINE
                ; DETERMINE MEMORY SIZE.
                ;    (START AT 9K AND TRY 1K INCREMENTS TILL END)
                ; SETUP POINTERS FOR STACK, DATA, AND PROGRAM
                ; INIT SIO BOARD
                ;
                INIT1:  IF      NOT CPM
0081 D303               OUT     TTY+1   ;INIT   TERMINAL
0083 3E40               MVI     A,40H
0085 D303               OUT     TTY+1
0087 3EBA               MVI     A,0BAH
0089 D303               OUT     TTY+1
008B 3E37               MVI     A,37H
008D D303               OUT     TTY+1
008F 010004             LXI     B,1024  ;1K INCR
0092 7E         INIT2:  MOV     A,M     ;GET A BYTE FROM MEMORY
0093 2F                 CMA             ;COMPLEMENT
0094 77                 MOV     M,A     ;REPLACE
0095 BE                 CMP     M       ;TEST IF RAM/ROM/END
0096 C29F00             JNZ     INIT3   ;BRIF OUT OF RAM
0099 2F                 CMA             ;RE-COMPLEMENT
009A 77                 MOV     M,A     ;PUT ORIG BACK
009B 09                 DAD     B       ;POINT NEXT BLOCK
009C D29200             JNC     INIT2   ;LOOP
                        ENDIF

009F F9         INIT3:  SPHL            ;SET STACK POINTER TO END OF MEMORY
00A0 0100FF             LXI     B,-256  ;ALLOW 256 BYTES FOR STACK
00A3 09                 DAD     B       ;ADD TO ADDRESS
00A4 229122             SHLD    DATAB   ;SAVE ADDR OF START OF DATA
                ;
                ; SOFTWARE WRITE PROTECT OF FIRST 9K OF RAM.
                ;
                ; BUT NO PROTECT UNDER CPM OR FOR 8K (EPROM) VERSION
                        IF      LARGE AND NOT CPM
                        MVI     A,2     ;SET PROTECT OF FIRST 1K BLOCK
                PROTC:  OUT     0FEH    ;SEND IT
                        ADI     4       ;ADDRESS NEXT 1K BLOCK
                        CPI     26H     ;STOP AFTER 9 BLOCKS
                        JNZ     PROTC   ;CONTINUE TO PROTECT
                        ENDIF
00A7 AF                 XRA     A       ;GET A ZERO IN A
00A8 F5                 PUSH    PSW     ;SET STACK 1 LEVEL DEEP WITHOUT A GOSUB
00A9 210000             LXI     H,0     ;CLEAR H,L
00AC 39                 DAD     SP      ;SP TO H,L
00AD 228B22             SHLD    STACK   ;SAVE BEG OF STACK
00B0 CD5101             CALL    IRAM    ;INIT RAM
00B3 11B81D             LXI     D,NRNDX ;POINT TO RANDOM # SERIES
00B6 0608               MVI     B,8     ;LOAD COUNT
00B8 CD4D1C             CALL    COPYD   ;COPY TO TRND<X> IN RAM TABLE
00BB 3602               MVI     M,2     ;SET RANDOM SWITCH
                        IF      CPM
                        CALL    NEW0    ;AUTOMATIC "NEW"
```

```
                        ENDIF
00BD 21781D             LXI     H,VERS  ;POINT VERSION MESSAGE
 1C0 CDBD19     RDYM:   CALL    TERMM   ;WRITE IT
                ;
00C3 =          RDY     EQU     $
                ;
                ; PRINT 'READY'
                ;
00C3 21261E             LXI     H,READY ;POINT READY MSG
00C6 CDBD19             CALL    TERMM   ;GO PRINT IT
                ;
00C9 =          GETCM   EQU     $
                ;
                ;
                ; COMMAND INPUT ROUTINE
                ;
                ; READ A LINE FROM THE TTY
                ; IF STARTS WITH NUMERIC CH, ASSUME IT'S A BASIC STATMENT
                ; IF NOT, IT IS EITHER AN IMMEDIATE STATEMENT, OR A COMMAND
                ;
00C9 3E3A               MVI     A,':'   ;PROMPT & ON SET FOR SW
00CB 327620             STA     EDSW    ;SET MODE=EDIT
00CE 2A8B22             LHLD    STACK   ;GET STACK ADDRESS
00D1 F9                 SPHL            ;SET REG SP
00D2 CD0419             CALL    TERMI   ;GET A LINE
00D5 CD851A             CALL    PACK    ;GO PACK THE NUMBER INTO B,C
00D8 78                 MOV     A,B     ;GET HI BYTE OF LINE NUMBER
00D9 B1                 ORA     C       ;PLUS LOW BYTE
00DA CA6401             JZ      EXEC    ;BRIF EXEC STATEMENT
00DD C5                 PUSH    B       ;SAVE LINE NUMBER
00DE 117D20             LXI     D,IMMED+1       ;POINT SAVE AREA
00E1 EB                 XCHG            ;FLIP/FLOP
00E2 70                 MOV     M,B     ;PUT LO LINE
00E3 23                 INX     H       ;POINT NEXT
00E4 71                 MOV     M,C     ;PUT LO LINE
00E5 23                 INX     H       ;POINT NEXT
00E6 0603               MVI     B,3     ;INIT COUNT
00E8 1A         EDIT1:  LDAX    D       ;GET A BYTE
00E9 77                 MOV     M,A     ;PUT IT DOWN
00EA 04                 INR     B       ;COUNT IT
00EB 23                 INX     H       ;POINT NEXT
00EC 13                 INX     D       ;DITTO
00ED B7                 ORA     A       ;TEST BYTE JUST MOVED
00EE C2E800             JNZ     EDIT1   ;LOOP
00F1 78                 MOV     A,B     ;GET COUNT
00F2 327C20             STA     IMMED   ;STORE THE COUNT
00F5 C1                 POP     B       ;GET LINE NUM
00F6 CD5E1F             CALL    LOCAT   ;GO FIND REQUESTED LINE NUMBER
00F9 E5                 PUSH    H       ;SAVE H,L
00FA DA1401             JC      EDIT5   ;BRIF IF LINE NOT FOUND
00FD 54         EDIT2:  MOV     D,H     ;COPY ADDR
00FE 5D                 MOV     E,L     ;TO D,E
00FF 0600               MVI     B,0     ;GET A ZERO
0101 4E                 MOV     C,M     ;GET LEN
0102 09                 DAD     B       ;POINT NEXT STMT
0103 7E         EDIT3:  MOV     A,M     ;GET LEN NEXT STMT
0104 B7                 ORA     A       ;TEST IT
0105 CA0F01             JZ      EDIT8   ;BRIF END
```

```
0108 47                 MOV     B,A     ;SET LENGTH
0109 CD581C             CALL    COPYH   ;ELSE MOVE LINE
010C C30301             JMP     EDIT3   ;LOOP
010F EB         EDIT8:  XCHG            ;PUT NEW ADDR TO H,L
0110 77                 MOV     M,A     ;MARK END
0111 229322             SHLD    PROGE   ;AND UPDATE ADDRESS
0114 3A7C20     EDIT5:  LDA     IMMED   ;GET LEN OF INSERT
0117 FE04               CPI     4       ;TEST IF DELETE
0119 CAC900             JZ      GETCM   ;BRIF IS
011C 4F                 MOV     C,A     ;SET LO LEN
011D 0600               MVI     B,0     ;ZERO HI LEN
011F 2A9322             LHLD    PROGE   ;GET END OF PROG
0122 54                 MOV     D,H     ;COPY TO
0123 5D                 MOV     E,L     ;D,E
0124 09                 DAD     B       ;DISP LEN OF INSERT
0125 229322             SHLD    PROGE   ;UPDATE END POINT
0128 C1                 POP     B       ;GET ADDR
0129 1A         EDIT6:  LDAX    D       ;GET A BYTE
012A 77                 MOV     M,A     ;COPY IT
012B 1B                 DCX     D       ;POINT PRIOR
012C 2B                 DCX     H       ;DITTO
012D 7A                 MOV     A,D     ;GET HI ADDR
012E B8                 CMP     B       ;COMPARE
012F CA3501             JZ      EDIT7   ;BRIF HI EQUAL
0132 D22901             JNC     EDIT6   ;BRIF NOT LESS
0135 78         EDIT7:  MOV     A,E     ;GET LO ADDR
0136 B9                 CMP     C       ;COMPARE
0137 D23D01             JNC     ED7A    ;MUST TEST FOR 00 BOUNDARY
013A C34601             JMP     ED7B    ;GO AROUND BOUNDARY TEST CODE
013D 2F         ED7A:   CMA             ;COMPLIMENT LOW LINE NUMBER,
013E B9                 CMP     C       ;AND COMPARE TO START
013F C22901             JNZ     EDIT6   ;BRIF NOT =
0142 B7                 ORA     A       ;NOW TEST FOR 00
0143 C22901             JNZ     EDIT6   ;THIS IS USUAL CASE
0146 13         ED7B:   INX     D       ;POINT FORWARD
0147 217C20             LXI     H,IMMED ;POINT INSERT
014A 46                 MOV     B,M     ;GET LENGTH
014B CD581C             CALL    COPYH   ;GO MOVE IT
014E C3C900             JMP     GETCM   ;GO GET ANOTHER COMMAND


                ; IRAM           INITIALIZE RAM
                ;       ZEROES RAM FROM BZERO TO EZERO
                ;       INITS RANDOM # CONSTANTS
                ;       RETURNS H=PTR TO TRND

0151 210020     IRAM:   LXI     H,BZERO ;CLEAR BZERO->EZERO
0154 0677               MVI     B,EZERO-BZERO
0156 CD5E1C             CALL    ZEROM
0159 116B1D             LXI     D,NRNDX ;MOVE RANDOM # SERIES TO RNDX
015C 217722             LXI     H,RNDX
015F 0608               MVI     B,8     ;COUNT
0161 C34D1C             JMP     COPYD   ;MOVE IT & RETURN
                ;PAGE
```

```
0164 =          EXEC    EQU     $
                ;
                ;
                ;
                ; DECODE COMMAND IN IOBUFF
                ; EXECUTE IF POSSIBLE
                ; THEN GOTO GET NEXT COMMAND
                ;
                ;
0164 327422             STA     MULTI   ;RESET MULTI SW
0167 328822             STA     FNMOD   ;RESET FN TYPE
016A 3C                 INR     A       ;GET A ONE
016B 327520             STA     RUNSW   ;SET IMMEDIATE MODE
016E 21CF20             LXI     H,IOBUF+1       ;POINT SMT
0171 117C20             LXI     D,IMMED ;POINT NEW AREA
0174 7E         EXEC1:  MOV     A,M     ;GET A BYTE
0175 12                 STAX    D       ;PUT TO (D,L)
0176 13                 INX     D       ;POINT NEXT
0177 23                 INX     H       ;DITTO
0178 B7                 ORA     A       ;TEST BYTE
0179 C27401             JNZ     EXEC1   ;CONTINUE
017C 21EC1D             LXI     H,NULLI ;POINT NO LINE NUM
017F 228922             SHLD    LINE    ;SAVE ADDR
0182 217C20             LXI     H,IMMED ;POINT START OF CMMD
0185 C33702             JMP     RUN3    ;GO INTO RUN PROCESSOR
                ;
0188 =          NEW     EQU     $
                ;
                ; NEW COMMAND
                ; 'NEW'==>CLEAR PROGRAM AND DATA
                ; 'NEW*'==>CLEAR PROGRAM ONLY
                ;
0188 E5                 PUSH    H       ;SAE PTR
0189 21C900             LXI     H,GETCM ;MAKE SUBROUTINE
018C E3                 XTHL            ;RESTORE H
018D CF                 RST     1       ;GET 1ST NON-BLANK CHAR AFTER 'NEW'
018E DE2A               SBI     '*'     ;TEST
0190 CA9801             JZ      NEW1    ;BRIF PROGRAM CLEAR ONLY
0193 AF         NEW0:   XRA     A       ;GET A ZERO
0194 2A9122             LHLD    DATAB   ;POINT DATA AREA
0197 77                 MOV     M,A     ;CLEAR IT
0198 219622     NEW1:   LXI     H,BEGPR ;POINT START
019B 229322             SHLD    PROGE   ;RESET PROGRAM END
019E 77                 MOV     M,A     ;CLEAR IT
019F C9                 RET
                ;
01A0 =          FREE    EQU     $
                ;
                ; FREE COMMAND
                ; COMPUTE AMOUNT OF AVAILABLE STORAGE (EXCLUDING DATA AREA)
                ;
01A0 2A9122             LHLD    DATAB   ;GET DATA BEG ADDRESS
01A3 EB                 XCHG            ;PUT IN D,E
01A4 2A9322             LHLD    PROGE   ;GET PROGRAM END ADDRESS
01A7 78                 MOV     A,E     ;LO ADDR TO REG A
01A8 95                 SUB     L       ;SUBTRACT
01A9 5F                 MOV     E,A     ;SAVE IT
01AA 7A                 MOV     A,D     ;HI ADDR TO REG A
```

```
01AB 9C                 SBB     H       ;SUBTRACT
01AC 57                 MOV     D,A     ;SAVE IT
01AD CD891C             CALL    BINFL   ;GO FLOAT D,E
01B0 21CE20             LXI     H,IOBUF ;POINT BUFFER
01B3 CDF014             CALL    FOUT    ;GO CONVERT TO OUTPUT
01B6 3600               MVI     M,00H   ;MARK END
01B8 CD8519             CALL    TERMO   ;GO WRITE IT
01BB C3C900             JMP     GETCM   ;CONTINUE
                ;
01BE =          TAPE    EQU     $
                ;
                ; TAPE COMMAND. DON'T ECHO INPUT. CONTINUE UNTIL KEY
                ; COMMAND.
                ;
01BE 3E01               MVI     A,1     ;SET TAPE INPUT SWITCH
01C0 327120             STA     TAPES   ;STORE IT
01C3 3E11               MVI     A,11H   ;GET DC1 (=READER ON)
01C5 CD4F19             CALL    TESTO   ;WRITE IT
01C8 C3C900             JMP     GETCM   ;GO PROCESS INPUT
                ;
01CB =          ENDIT   EQU     $
                ;
                ; END COMMAND. IF TAPE PUNCH SWITCH IS ON, PUNCH 'KEY' THEN
                ; CONTINUE
                ;
01CB 3A7120             LDA     TAPES   ;GET PAPER TAPE SWITCH
01CE FE02               CPI     2       ;TEST FOR SAVE
01D0 C2C300             JNZ     RDY     ;BRIF NOT
01D3 21791E             LXI     H,KEYL  ;POINT 'KEY'
01D6 CD8D19             CALL    TERMM   ;WRITE IT
01D9 CDE601             CALL    HDRTL   ;GO PUT TRAILER


                ; KEY COMMAND. RESET TAPE SWITCH. TURN READER OFF

01DC AF         KEY:    XRA     A       ;RESET TAPE SWITCH
01DD 327120             STA     TAPES
01E0 21621D             LXI     H,PCHOF ;POINT READER/PUNCH OFF
01E3 C3C000             JMP     RDYM    ;PRINT POFF+READY MESSAGE
                ;
01E6 =          HDRTL   EQU     $
                ;
                ; PUNCH HEADER OR TRAILER ON PAPER TAPE.
                ;
01E6 0619               MVI     B,25    ;LOAD COUNT
01E8 3EFF       HDR1:   MVI     A,0FFH  ;LOAD RUBOUT
01EA CD4F19             CALL    TESTO   ;WRITE IT
01ED 05                 DCR     B       ;DECREMENT COUNT
01EE AF                 XRA     A       ;ZERO A
01EF B8                 CMP     B       ;TEST COUNT
01F0 C8                 RZ              ;RETURN ON ZERO
01F1 C3E801             JMP     HDR1    ;CONTINUE
                ;PAGE
```

```
                ; RUN PROCESSOR, GET NEXT STATMENT, AND EXECUTE IT
                ; IF IN IMMEDIATE MODE, THEN RETURN TO GETCMMD

01F4 AF         RUNCM:  XRA     A       ;PUT A ZERO TO A
01F5 2A9122             LHLD    DATAB   ;GET ADDRESS OF DATA POOL
01F8 77                 MOV     M,A     ;INITIALIZE TO 0
01F9 =          XEQ     EQU     $       ;START FOR EXECUTION WITH OLD DATA
01F9 CD5101             CALL    IRAM    ;INITIALIZE START OF RAM
01FC 219522             LXI     H,BEGPR-1       ;POINT 1 PRIOR TO BEGIN
01FF 228F22             SHLD    DATAP   ;RESTORE DATA STMT POINTER
0202 3600               MVI     M,0     ;RESET DATA STMT POINTER
0204 23                 INX     H       ;POINT TO START
0205 227022             SHLD    STMT    ;SAVE IT
0208 C32502             JMP     RUN2    ;GO PROCESS IT

                ; STATEMENTS RETURN HERE TO CONTINUE PROCESSING
020B 217422     RUN:    LXI     H,MULTI ;POINT MULTIPLE SWITCH
020E 7E                 MOV     A,M     ;GET SW
020F B7                 ORA     A       ;TEST IT
0210 CA1802             JZ      RUN1    ;BRIF NOT ON
0213 3600               MVI     M,0     ;ELSE, RESET IT
0215 2A7222             LHLD    ENDLI   ;GET ADDRESS
0218 C33702             JMP     RUN3    ;GO PROCESS REMAIN
021B 2A7022     RUN1:   LHLD    STMT    ;ELSE, GET ADDR OF PREV STMT
021E 5E                 MOV     E,M     ;GET LEN CODE
021F 1600               MVI     D,0     ;CLEAR HIGH BYTE OF ADDR
 221 19                 DAD     D       ;INCR STMT POINTER
0222 227022             SHLD    STMT    ;SAVE IT
0225 3A7520     RUN2:   LDA     RUNSW   ;GET RUN TYPE
0228 B7                 ORA     A       ;TEST IT
0229 C2C900             JNZ     GETCM   ;BRIF IMMEDIATE MODE
022C 7E                 MOV     A,M     ;GET LEN CODE
022D B7                 ORA     A       ;TEST IF END
022E CAC801             JZ      ENDIT   ;BRIF IS
0231 23                 INX     H       ;POINT LINE NUMBER
0232 228922             SHLD    LINE    ;SAVE ADDR
0235 23                 INX     H       ;POINT 2ND BYTE
0236 23                 INX     H       ;POINT 1ST PGM BYTE

                ; ENTER HERE TO DO IMMEDIATE COMMAND
0237 CF         RUN3:   RST     1       ;SKIP BLANKS
0238 225222     RUN4:   SHLD    ADDR1   ;SAVE ADDR
023B CD3A1A             CALL    TSTCC   ;GO SEE IF CONTROL-C OR 0
023E 114C1E             LXI     D,JMPTB ;POINT TO TABLE
0241 CD861F             CALL    SEEK    ;GO SEARCH COMMAND TABLE
0244 CA4F02             JZ      RUN7    ;BRIF COMMAND NOT FOUND
0247 E5                 PUSH    H       ;SAVE H,L
0248 1A                 LDAX    D       ;LOAD LOW BYTE
0249 6F                 MOV     L,A     ;LOW BYTE TO L
024A 13                 INX     D       ;POINT NEXT
024B 1A                 LDAX    D       ;LOAD HIGH BYTE
)24C 67                 MOV     H,A     ;HIGH BYTE TO H
 24D E3                 XTHL            ;COMMAND ADDRESS TO STACK
024E C9                 RET             ;JUMP TO ROUTINE
024F 2A5222     RUN7:   LHLD    ADDR1   ;RESTORE H,L POINTER
0252 C3F105             JMP     LET     ;ASSUME IT'S LET STMT
                ;PAGE
```

```
                ;SAVE COMMAND. TURN THE PUNCH ON THEN LIST PROGRAM

0255 3E02       SAVE:   MVI     A,2     ;SET PUNCH MODE
0257 327120             STA     TAPES
025A 3E12               MVI     A,12H   ;GET DC2 (=PUNCH ON)
025C CD4F19             CALL    TESTO   ;WRITE IT
025F CDE601             CALL    HDRTL   ;GP PUT HEADER
                ;
0262 =          LIST    EQU     $
                ;
                ;
                ; LIST PROCESSOR
                ; DUMP THE SOURCE PROGRAM TO TTY OR PAPER TAPE
                ;
                ;
0262 CF                 RST     1       ;SKIP TO NON BLANK
0263 110000             LXI     D,0     ;GET A ZERO IN D
0266 EB                 XCHG            ;FLIP TO H,L
0267 224B22             SHLD    LINEL   ;SAVE IT
026A 219999             LXI     H,9999H ;GET HIGH NUMBER IN H,L
026D 224D22             SHLD    LINEH   ;SAVE IT
0270 EB                 XCHG            ;FLIP BACK
0271 B7                 ORA     A       ;TEST IF EOL
0272 CA9202             JZ      LIST1   ;BRIF IT IS
0275 CD851A             CALL    PACK    ;GO PACK THE NUMBER, IF ANY
0278 50                 MOV     D,B     ;COPY NUMBER TO D,L
0279 59                 MOV     E,C     ;SAME
027A EB                 XCHG            ;FLIP TO H,L
027B 224B22             SHLD    LINEL   ;SAVE IT
027E 224D22             SHLD    LINEH   ;SAME
0281 EB                 XCHG            ;RESTORE H,L
0282 CF                 RST     1       ;SKIP TO NON BLANK
0283 FE2C               CPI     ','     ;TEST IF COMMA
0285 C29202             JNZ     LIST1   ;BRIF NOT
0288 23                 INX     H       ;POINT NEXT
0289 CF                 RST     1       ;SKIP TO NON-BLANK
028A CD851A             CALL    PACK    ;ELSE, GO GET THE NUMBER
028D 60                 MOV     H,B     ;COPY TO
028E 69                 MOV     L,C     ;D,L
028F 224D22             SHLD    LINEH   ;SAVE IT
0292 219622     LIST1:  LXI     H,BEGPR ;POINT BEGINNING OF PROGRAM
0295 CD3A1A     LIST2:  CALL    TSTCC   ;GO SEE IF CONTROL-C OR CONTROL-O
0298 7E                 MOV     A,M     ;GET LEN CODE
0299 B7                 ORA     A       ;TEST IF END OF PROGRAM
029A CACB01             JZ      ENDIT   ;BRIF END OF PGM
029D D603               SUI     3       ;SUBTRACT THREE
029F 47                 MOV     B,A     ;SAVE LEN
02A0 23                 INX     H       ;POINT HIGH BYTE OF LINE#
02A1 EB                 XCHG            ;FLIP H,L TO D,E
02A2 2A4B22             LHLD    LINEL   ;GET LOW LINE TO TEST
02A5 EB                 XCHG            ;RESTORE H,L
02A6 7E                 MOV     A,M     ;GET LOW BYTE OF LINE NUMBER
02A7 BA                 CMP     D       ;COMP WITH LINEL
02A8 DAE502             JC      LIST8   ;BRIF LESS
02AB C2B502             JNZ     LIST4   ;BRIF NOT EQUAL
02AE 23                 INX     H       ;POINT NEXT
02AF 7E                 MOV     A,M     ;GET NEXT BYTE OF LINE#
02B0 2B                 DCX     H       ;POINT BACK
```

```
02B1 BB                 CMP     E       ;COMP LOW BYTES
02B2 DAE502             JC      LIST8   ;BRIF LESS
02B5 EB         LIST4:  XCHG            ;SAVE H,L IN D,E
02B6 2A4D22             LHLD    LINEH   ;GET HIGH LINE FOR TEST
 289 EB                 XCHG            ;RESTORE H,L
02BA 7E                 MOV     A,M     ;GET LINE BYTE
02BB BA                 CMP     D       ;COMPARE HIGH BYTES
02BC CAC502             JZ      LIST5   ;BRIF EQUAL
02BF D2C801             JNC     ENDIT   ;BRIF HIGHER
02C2 C3CF02             JMP     LIST6   ;GO AROUND
02C5 23         LIST5:  INX     H       ;POINT NEXT
02C6 7E                 MOV     A,M     ;GET NEXT BYTE
02C7 2B                 DCX     H       ;POINT BACK
02C8 BB                 CMP     E       ;COMPARE LOW BYTES
02C9 CACF02             JZ      LIST6   ;BRIF EQUAL
02CC D2C801             JNC     ENDIT   ;BRIF HIGHER
02CF 11CE20     LIST6:  LXI     D,IOBUF ;POINT BUFFER AREA
02D2 CD091A             CALL    LINEO   ;CONVERT LINE NUMBER
02D5 7E         LIST7:  MOV     A,M     ;GET A BYTE
02D6 12                 STAX    D       ;PUT IT TO BUFFER
02D7 13                 INX     D       ;POINT NEXT BUFF
02D8 23                 INX     H       ;POINT NEXT PROG
02D9 05                 DCR     B       ;DECR CTR
02DA C2D502             JNZ     LIST7   ;LOOP
02DD E5                 PUSH    H       ;SAVE HL ADDR
02DE CD8519             CALL    TERMO   ;GO TYPE IT
02E1 E1                 POP     H       ;RETREIVE H ADDR
02E2 C39502             JMP     LIST2   ;CONTINUE
02E5 58         LIST8:  MOV     E,B     ;PUT LEN  IN E
 2E6 1600               MVI     D,0     ;CLEAR D
02E8 19                 DAD     D       ;POINT NEXT STMT
02E9 23                 INX     H       ;POINT NEXT
02EA 23                 INX     H       ;POINT LEN CODE
02EB C39502             JMP     LIST2   ;GO LIST IT
                ;
                ;
02EE =          CONTI   EQU     $
                ;
                ; CONTINUE EXECUTION AT STATEMENT FOLLOWING STOP OR AT
                ; STATEMENT THAT WAS INTERRUPTED WHEN CONTROL-C WAS TYPED
                ;
                ;
02EE 217720             LXI     H,LINEN ;POINT LINE NUMBER OF LAST STOP/ERROR/
02F1 7E                 MOV     A,M     ;GET 1ST CHAR
02F2 B7                 ORA     A       ;TEST IF IMMED CMMD
02F3 CAF105             JZ      LET     ;BRIF IF IMMED CMMD
                ;PAGE
```

```
                        ;
                        ;
                        ; STMT:  GOTO NNNN
                        ;
02F6 AF         GOTO:   XRA     A       ;CLEAR REG A
02F7 327620             STA     EDSW    ;RESET IMMED MODE (IF IT WAS SET)
02FA 327520             STA     RUNSW   ;AND RUN TYPE
02FD CDAD1A             CALL    NOTEO   ;ERROR IF END-OF-LINE
0300 CD851A             CALL    PACK    ;GO GET LINE NUMBER IN B,C
0303 CD941A             CALL    EOL     ;ERROR IF NOT END-OF-LINE
0306 CD5E1F     GOTO2:  CALL    LOCAT   ;GO SEARCH FOR REQUESTED LINE #
0309 DA031C             JC      ULERR   ;BRIF NOT FOUND
030C 227022             SHLD    STMT    ;SAVE ADDR
030F AF                 XRA     A       ;GET A ZERO
0310 327422             STA     MULTI   ;TURN OFF MULTIPLE STMTS
0313 C32502             JMP     RUN2    ;GO PROCESS THE STATEMENT
                        ;
                        ;
                        ; STMT: RESTORE
                        ;
0316 CD941A     RESTO:  CALL    EOL     ;ERROR IF NOT END-OF-LINE
0319 219522             LXI     H,BEGPR-1  ;POINT 1 BEFORE START OF PROGRAM
031C 228F22             SHLD    DATAP   ;FORCE NEXT DATA TO BE AT START
031F C30802             JMP     RUN     ;GO NEXT STMT
                        ;
                        ;
                        ; STMT:  RETURN
                        ;
0322 CD941A     RETUR:  CALL    EOL     ;ERROR IF NOT END-OF-LINE
0325 F1                 POP     PSW     ;POP THE STACK
0326 FEFF               CPI     0FFH    ;TEST IF GOSUB IN EFFECT
0328 C2131C             JNZ     RTERR   ;BRIF ERROR
032B E1                 POP     H       ;GET RETURNED STATMENT ADDRESS
032C 227022             SHLD    STMT    ;RESTORE
032F E1                 POP     H       ;GET ENDLINE VALUE
0330 227222             SHLD    ENDLI   ;RESTORE
0333 F1                 POP     PSW     ;GET MULTI SW VALUE
0334 327422             STA     MULTI   ;RESTORE
0337 C30802             JMP     RUN     ;CONTINUE (AT STMT FOLLOWING GOSUB)
                        ;
                        ;
                        ; STMT:  GOSUB NNNN
                        ;
033A CDAD1A     GOSUB:  CALL    NOTEO   ;ERROR IF END-OF-LINE
033D CD851A             CALL    PACK    ;GET LINE NUMBER
0340 CD941A             CALL    EOL     ;ERROR IF NOT END-OF-LINE
0343 3A7422     GOSU1:  LDA     MULTI   ;GET SW SETTING
0346 F5                 PUSH    PSW     ;SAVE ON STACK
0347 2A7222             LHLD    ENDLI   ;GET ADDR OF END OF STMT
034A E5                 PUSH    H       ;SAVE ONE STACK
034B 2A7022             LHLD    STMT    ;GET STATEMENT ADDRESS
034E E5                 PUSH    H       ;SAVE RETURN ADDRESS IN STACK
034F 3EFF               MVI     A,0FFH  ;MARK AS GOSUB
0351 F5                 PUSH    PSW     ;SAVE STATUS
0352 C30603             JMP     GOTO2   ;GO LOOKUP LINE AND BRANCH
                        ;PAGE
```

```
                        ;
0355 =          PRINT   EQU     $
                        ;
                        ;
                        ; STMT: PRINT ....
                        ;
                        ;
0355 AF                 XRA     A       ;CLEAR REG A
0356 328D22     PRIN1:  STA     PRSW    ;SET SW TO SAY CRLF AT END OF LINE
0359 11CE20             LXI     D,IOBUF ;POINT BUFFER
035C CF                 RST     1       ;SKIP TO NEXT FIELD

035D CDA81A     PRIN4:  CALL    TSTEL   ;TEST IF END OF STMT
0360 CAD303             JZ      PRINC   ;BRIF IT IS
0363 FE2C               CPI     ','     ;TEST IF COMMA
0365 CAAA03             JZ      PRIN8   ;BRIF IT IS
0368 FE3B               CPI     ';'     ;TEST IF SEMI-COLON
036A CAAD03             JZ      PRIN9   ;BRIF IT IS
036D D5                 PUSH    D       ;SAVE D,E
036E E5                 PUSH    H       ;SAVE H,L
036F 11891D             LXI     D,TABLI ;POINT LITERAL
0372 D7                 RST     2       ;GO SEE IF TAB(XX)
0373 CAB303             JZ      PRINA   ;BRIF IS
0376 E1                 POP     H       ;ELSE, RESTORE H,L
0377 CD800F             CALL    EXPR    ;GO EVALUATE EXPRESSION
037A D1                 POP     D       ;RESTORE D,E
037B E5                 PUSH    H       ;SAVE H,L
037C EB                 XCHG            ;FLIP/FLOP
037D 3A8E22             LDA     NS      ;GET TYPE OF RESULT
0380 FEE7               CPI     0E7H    ;TEST IF STRING
0382 CA9603             JZ      PRIN5   ;BRIF IS
0385 CDF014             CALL    FOUT    ;GO CONVERT OUTPUT
0388 23                 INX     H       ;POINT NEXT
0389 EB         PRIN7:  XCHG            ;FLIP/FLOP: END ADDR TO DE
038A E1                 POP     H       ;RESTORE H,L
                ;HERE AFTER SETTING UP VALUE TO PRINT IN BUFFER
038B 3EFE       PRIN2:  MVI A,0FEH      ;SET END CODE=NO CRLF
038D 12                 STAX D          ;PUT TO BUFFER
038E E5                 PUSH H          ;SAVE H,L
038F CDB519             CALL TERMO      ;GO PRINT BUFFER
0392 E1                 POP H           ;RESTORE HL
0393 C35503             JMP PRINT       ;REPEAT FOR NEXT FIELD

0396 112021     PRIN5:  LXI     D,STRIN ;POINT STRING
0399 1A                 LDAX    D       ;GET LEN
039A B7                 ORA     A       ;TEST IT
039B CA8903             JZ      PRIN7   ;BRIF NULL
039E 47                 MOV     B,A     ;SAVE LEN
039F 13         PRIN6:  INX     D       ;POINT NEXT
03A0 1A                 LDAX    D       ;GET A BYTE
03A1 77                 MOV     M,A     ;STORE IT
03A2 23                 INX     H       ;POINT NEXT
03A3 05                 DCR     B       ;DECR CTR
03A4 C29F03             JNZ     PRIN6   ;LOOP
03A7 C38903             JMP PRIN7       ;DIDDLE DE, HL AND CONTINUE

03AA CDDF19     PRIN8:  CALL    TABST   ;GO POSITION NEXT TAB
03AD 23         PRIN9:  INX     H       ;POINT NEXT
```

```
03AE 3E01               MVI     A,1       ;GET SETTING FOR SW
03B0 C35603             JMP     PRIN1     ;GO STORE A IN PRSW & DO NEXT FIELD
03B3 D1         PRINA:  POP     D         ;GET RID OF STACK ENTRY
03B4 CD800F             CALL    EXPR      ;GO EVALUATE
03B7 E5                 PUSH    H         ;SAVE H,L
03B8 CD661C             CALL    FBIN      ;CONVERT TO BINARY
03BB F5                 PUSH    PSW       ;SAVE SPECIFIED COLUMN
03BC 217622             LXI     H,COLUM   ;POINT CURRENT POSITION
03BF 96                 SUB     M         ;SUBTRACT (LEAVES NUMBER OF FILLS)
03C0 FC5A19             CM      CRLF      ;NEXT LINE IF ALREADY PAST
03C3 F1                 POP     PSW       ;RESTORE COL
03C4 96                 SUB     M         ;GET NUMBER FILLS
03C5 E1                 POP     H
03C6 D1                 POP     D
03C7 47                 MOV     B,A       ;SAVE COUNT
03C8 3E20               MVI     A,' '     ;GET FILL
03CA CA8B03     PRINB:  JZ      PRIN2     ;BRIF COUNT ZERO
03CD 12                 STAX    D         ;PUT ONE SPACE
03CE 13                 INX     D         ;POINT NEXT
03CF 05                 DCR     B         ;DECR CTR
03D0 C3CA03             JMP     PRINB     ;LOOP

03D3 CD941A     PRINC:  CALL EOL          ;SAVE EOL POSITION
                ;HERE TO PRINT FINAL CR/LF (OR NOT) AND GO TO NEXT STATEMENT
03D6 3A8D22     PRIN3:  LDA PRSW          ;GET SWITCH
03D9 47                 MOV B,A           ;SAVE ,; SWITCH
03DA 3A7320             LDA OUTSW         ;GET CONTROL-O SWITCH
03DD B7                 ORA A             ;TEST IF bO IN EFFECT
03DE B0                 ORA B             ;AND IF STATEMENT ENDED IN , OR ;
03DF CC5A19             CZ CRLF           ;CRLF IF NEITHER
03E2 C30B02             JMP RUN           ;CONTINUE NEXT STATEMENT
                ;PAGE
```

```
                        ;
03E5 =                  FOR     EQU     $
                        ;
                        ;
                        ;  STMT:  FOR VAR = EXPR TO EXPR [STEP EXPR]
                        ;
                        ;
                        ;  FIRST EVALUATE ARGUMENTS AND STORE POINTERS AND VALUES,
                        ;  BUT DO NOT MAKE TABLE ENTRY YET
03E5 CDC91B                     CALL    VAR     ;NEXT WORD MUST BE VARIABLE
03E8 E8                         XCHG            ;FLIP/FLOP
03E9 222322                     SHLD    INDX    ;SAVE VARIABLE NAME
03EC E8                         XCHG            ;FLIP/FLOP AGAIN
03ED FE3D                       CPI     '='     ;TEST FOR EQUAL SIGN
03EF C20F1C                     JNZ     SNERR   ;BRIF NO EQUAL
03F2 23                         INX     H       ;POINT NEXT
03F3 CD800F                     CALL    EXPR    ;GO EVALUATE EXPR, IF ANY
03F6 E8                         XCHG            ;FLIP/FLOP AGAIN
03F7 2A2322                     LHLD    INDX    ;GET INDEX NAME
03FA E8                         XCHG            ;FLIP/FLOP
03FB E5                         PUSH    H       ;SAVE H,L
03FC CD341B                     CALL    SEARC   ;GO LOCATE NAME
03FF E8                         XCHG            ;PUT ADDR IN H,L
0400 225222                     SHLD    ADDR1   ;SAVE ADDR
0403 DF                         RST     3       ;GO STORE THE VALUE
0404 E1                         POP     H       ;RESTORE POINTER TO STMT
0405 11D21E                     LXI     D,TOLIT ;GET LIT ADDR
0408 D7                         RST     2       ;GO COMPARE
0409 C20F1C                     JNZ     SNERR   ;BRIF ERROR
040C CD800F                     CALL    EXPR    ;GO EVALUATE TO-EXPR
040F E5                         PUSH    H       ;SAVE H,L
0410 212722                     LXI     H,TVAR1 ;POINT 'TO' VALUE
0413 DF                         RST     3       ;SAVE IT
0414 21EA1D                     LXI     H,ONE   ;POINT CONSTANT: 1
0417 EF                         RST     5       ;LOAD IT
0418 E1                         POP     H       ;GET H,L
0419 7E                         MOV     A,M     ;GET THE CHAR
041A B7                         ORA     A       ;TEST FOR END OF STATMENT
041B CA2E04                     JZ      FOR2    ;BRIF NO STEP
041E E5                         PUSH    H       ;RE-SAVE
041F 118D1D                     LXI     D,STEPL ;TEST FOR LIT 'STEP'
0422 D7                         RST     2       ;GO COMPARE
0423 CA2A04                     JZ      FOR1    ;BRIF STEP
0426 E1                         POP     H       ;RESTORE H,L
0427 C32E04                     JMP     FOR2    ;GO NO STEP VALUE
042A D1                 FOR1:   POP     D       ;POP OFF THE STACK
042B CD800F                     CALL    EXPR    ;GO EVALUATE EXPRESSION
042E E5                 FOR2:   PUSH    H       ;SAVE H,L TO END OF STATEMENT
042F 212822                     LXI     H,TVAR2 ;POINT STEP VALUE
0432 DF                         RST     3       ;SAVE IT
0433 E1                         POP     H       ;RESTORE H,L
0434 CD941A                     CALL    EOL     ;ERROR IF NOT END-OF-LINE
                        ; DETERMINE WHETHER LOOP IS TO BE EXECUTED AT ALL
                        ; (IF VALUE > "TO" VALUE AND STEP POSITIVE,
                        ;     JUST SKIP TO NEXT, ETC)
0437 CDCE18                     CALL    FTEST   ;GET STATUS OF FACC
043A F5                         PUSH    PSW     ;SAVE A,STATUS
043B 212722                     LXI     H,TVAR1 ;GET END VALUE
```

```
043E EF              RST     5       ;LOAD IT
043F F1              POP     PSW     ;RESTORE STATUS
0440 F25204          JP      FOR4    ;BRIF FOR IS POSITIVE
0443 2A5222          LHLD    ADDR1   ;GET ADDRESS OF INDEX
0446 CD0C17          CALL    FSUB    ;COMPARE THIS AGAINST END VALUE
0449 CA5E04          JZ      FOR5    ;BRIF START = END
044C FA5E04          JM      FOR5    ;BRIF START > END
044F C3B204          JMP     FOR9    ;GO LOCATE MATCHING NEXT
0452 2A5222  FOR4:   LHLD    ADDR1   ;GET ADDRESS OF INDEX
0455 CD0C17          CALL    FSUB    ;COMPARE
0458 CA5E04          JZ      FOR5    ;BRIF START = END
045B FAB204          JM      FOR9    ;BRIF START > END: SKIP TO "NEXT"
             ; LOOP IS TO BE EXECUTED AT LEAST ONCE:
             ; NEED AN ENTRY IN FOR-NEXT TABLE.
             ; SEE IF THERE IS ALREADY ENTRY FOR THIS VARIABLE
             ; (IE PROGRAM JUMPED OUT OF LOOP EARLIER)
045E 110020  FOR5:   LXI     D,FORNE ;POINT TABLE
0461 2A2322          LHLD    INDX    ;GET INDEX VARIABLE NAME
0464 EB              XCHG            ;FLIP/FLOP
0465 7E              MOV     A,M     ;GET COUNT OF ENTRIES NOW IN TABLE
0466 47              MOV     B,A     ;STORE IT
0467 0E01            MVI     C,1     ;NEW CTR
0469 B7              ORA     A       ;TEST IF ZERO
046A 23              INX     H       ;POINT
046B CA8104          JZ      FOR8    ;BRIF TABLE EMPTY
046E 7E      FOR6:   MOV     A,M     ;GET 1ST BYTE OF TABLE VARIABLE
046F BA              CMP     D       ;TEST IF EQUAL TO THIS FOR'S INDEX
0470 C27A04          JNZ     FOR7    ;BRIF NOT
0473 23              INX     H       ;POINT NEXT
0474 7E              MOV     A,M     ;GET NEXT BYTE
0475 2B              DCX     H       ;POINT BACK
0476 BB              CMP     E       ;TEST IF EQUAL
0477 CA8104          JZ      FOR8    ;BRIF EQUAL
047A E7      FOR7:   RST     4       ;ADJUST H,L
047B 0E              DB      14
047C 0C              INR     C       ;COUNT IT
047D 05              DCR     B       ;DECR CTR
047E C26E04          JNZ     FOR6    ;LOOP
             ; ENTER THIS FOR IN TABLE (WHERE HL POINTS)
0481 79      FOR8:   MOV     A,C     ;GET UPDATED COUNT
0482 FE09            CPI     9       ;TEST IF TBL EXCEEDED
0484 D2181C          JNC     NXERR   ;ERROR IF MORE THAN 8 OPEN FOR/NEXT
0487 320020          STA     FORNE   ;PUT IN TABLE
048A 72              MOV     M,D     ;HI BYTE INDEX VARIABLE NAME
048B 23              INX     H       ;POINT NEXT
048C 73              MOV     M,E     ;STORE LO BYTE
048D 23              INX     H       ;POINT NEXT
048E E5              PUSH    H       ;SAVE H,L
048F 212B22          LXI     H,TVAR2 ;POINT STEP VALUE
0492 EF              RST     5       ;LOAD IT
0493 E1              POP     H       ;RESTORE H,L
0494 DF              RST     3       ;STORE IN STACK
0495 E5              PUSH    H       ;SAVE H,L
0496 212722          LXI     H,TVAR1 ;POINT 'TO' VALUE
0499 EF              RST     5       ;LOAD IT
049A E1              POP     H       ;RESTORE H,L
049B DF              RST     3       ;STORE IN STACK
049C EB              XCHG            ;FLIP/FLOP
```

```
049D 2A7222             LHLD    ENDLI   ;GET END ADDR
04A0 28                 DCX     H       ;POINT ONE PRIOR
04A1 EB                 XCHG            ;FLIP BACK
04A2 72                 MOV     M,D     ;STORE IT
04A3 23                 INX     H       ;POINT NEXT
04A4 73                 MOV     M,E     ;STORE IT
04A5 23                 INX     H       ;POINT NEXT
04A6 3A7122             LDA     STMT+1  ;GET HIGH STMT ADDR
04A9 77                 MOV     M,A     ;PUT IT
04AA 23                 INX     H       ;POINT NEXT
04AB 3A7022             LDA     STMT    ;GET LOW STMT ADDR
04AE 77                 MOV     M,A     ;PUT IT
04AF C30B02             JMP     RUN     ;CONTINUE
                ;
                ; IF HERE, THIS LOOP IS TO BE EXECUTED ZERO TIMES:
                ; SCAN THRU PROGRAM TO FIND MATCHING "NEXT".
                ; THIS CODE WILL FAIL IF USER'S PROGRAM IS TOO
                ; COMPLEX SINCE IT WON'T FOLLOW GOTO'S, IF'S, ETC.
04B2 2A7022     FOR9:   LHLD    STMT    ;GET ADDRESS OF STATEMENT
04B5 5E                 MOV     E,M     ;GET LENGTH CODE
04B6 1600               MVI     D,0     ;INIT INCREMENT
04B8 19                 DAD     D       ;COMPUTE ADDR OF NEXT STATEMENT
04B9 7E                 MOV     A,M     ;GET NEW LEN CODE
04BA B7                 ORA     A       ;SEE IF END OF PGM
04BB CA1B1C             JZ      NXERR   ;BRIF IT IS
04BE 227022             SHLD    STMT    ;SAVE ADDRESS
04C1 E7                 RST     4       ;ADJUST H,L
04C2 03                 DB      3
04C3 CF                 RST     1       ;SKIP SPACES
04C4 11A81E             LXI     D,NEXTL ;POINT 'NEXT'
04C7 D7                 RST     2       ;SEE IF IT IS A NEXT STMT
04C8 C2B204             JNZ     FOR9    ;LOOP IF NOT
04CB CF                 RST     1       ;SKIP SPACES
04CC 3A2422             LDA     INDX+1  ;GET FIRST CHAR
04CF BE                 CMP     M       ;COMPARE
04D0 C2B204             JNZ     FOR9    ;BRIF NOT MATCH NEXT
04D3 3A2322             LDA     INDX    ;GET 2ND CHAR
04D6 23                 INX     H       ;DITTO
04D7 FE20               CPI     ' '     ;SEE IF SINGLE CHAR
04D9 CAE004             JZ      FORA    ;BRIF IT IS
04DC BE                 CMP     M       ;COMPARE THE TWO
04DD C2B204             JNZ     FOR9    ;BRIF NOT EQUAL
04E0 CF         FORA:   RST     1       ;SKIP TO END (HOPEFULLY)
04E1 7E                 MOV     A,M     ;GET THE NON BLANK
04E2 B7                 ORA     A       ;SEE IF END
04E3 C2B204             JNZ     FOR9    ;BRIF NOT END
04E6 C30B02             JMP     RUN     ;ELSE, GO NEXT STMT
                        ;PAGE
```

```
                        ;
04E9 =                  IFSTM   EQU     $
                        ;
                        ;
                        ; STMT: IF EXPR RELATION EXPR THEN STMT#
                        ;
                        ;
04E9 CD800F                     CALL    EXPR    ;GO EVALUATE LEFT EXPR
04EC E5                         PUSH    H       ;SAVE H,L
04ED 3A8E22                     LDA     NS      ;GET TYPE CODE
04F0 322622                     STA     IFTYP   ;SAVE IT
04F3 FEE7                       CPI     0E7H    ;TEST IF STRING
04F5 C20705                     JNZ     IF1     ;BRIF NOT
04F8 21CE20                     LXI     H,IOBUF ;POINT BUFFER
04FB 112021                     LXI     D,STRIN ;POINT RESULT
04FE 1A                         LDAX    D       ;GET LEN
04FF 3C                         INR     A       ;PLUS ONE
0500 47                         MOV     B,A     ;SAVE IT
0501 CD4D1C                     CALL    COPYD   ;GO MOVE IT
0504 C30805                     JMP     IF2     ;GO AROUND
0507 212722             IF1:    LXI     H,TVAR1 ;GET ADDR OF TEMP STORAGE
050A DF                         RST     3       ;SAVE IT
050B E1                 IF2:    POP     H       ;RESTORE H,L
050C AF                         XRA     A       ;CLEAR A
050D 4F                         MOV     C,A     ;SAVE IN REG C
050E 47                         MOV     B,A     ;INIT REG
050F 7E                 IF3:    MOV     A,M     ;GET OPERATOR
0510 04                         INR     B       ;COUNT
0511 FE3D                       CPI     '='     ;TEST FOR EQUAL
0513 C21805                     JNZ     IF4     ;BRIF IT IS
0516 0C                         INR     C       ;ADD 1 TO C
0517 23                         INX     H       ;POINT NEXT
0518 FE3E               IF4:    CPI     '>'     ;TEST FOR GREATER THAN
051A C22005                     JNZ     IF5     ;BRIF IT IS
051D 0C                         INR     C       ;ADD TWO
051E 0C                         INR     C       ;TO REL CODE
051F 23                         INX     H       ;POINT NEXT
0520 FE3C               IF5:    CPI     '<'     ;TEST FOR LESS THAN
0522 C22A05                     JNZ     IF6     ;BRIF IT IS
0525 79                         MOV     A,C     ;GET REL CODE
0526 C604                       ADI     4       ;PLUS FOUR
0528 4F                         MOV     C,A     ;PUT BACK
0529 23                         INX     H       ;POINT NEXT
052A 79                 IF6:    MOV     A,C     ;GET REL CODE
052B B7                         ORA     A       ;TEST IT
052C C5                         PUSH    B       ;SAVE B,C
052D CA0F1C                     JZ      SNERR   ;BRIF SOME ERROR
0530 C1                         POP     B       ;RESTORE B,C
0531 322522                     STA     REL     ;SAVE CODE
0534 78                         MOV     A,B     ;GET COUNT
0535 FE02                       CPI     2       ;TEST FOR TWO
0537 C20F05                     JNZ     IF3     ;SEE IF MULTIPLE RELATION
053A CD800F                     CALL    EXPR    ;GO EVALUATE RIGHT SIDE
053D 225222                     SHLD    ADDR1   ;SAVE LOCATION OF THEN (IF ANY)
0540 3A8E22                     LDA     NS      ;GET TYPE CODE
0543 212622                     LXI     H,IFTYP ;POINT LEFT TYPE
0546 BE                         CMP     M       ;COMPARE
0547 C20F1C                     JNZ     SNERR   ;BRIF MIXED
```

```
054A FEE7              CPI     0E7H    ;TEST IF STRING
054C CAA805            JZ      IFF     ;BRIF IS
054F 212722            LXI     H,TVAR1 ;POINT LEFT
0552 CD0C17            CALL    FSUB    ;SUBTRACT LEFT FROM RIGHT
0555 3A2522            LDA     REL     ;GET RELATION
0558 1F                RAR             ;TEST BIT D0
0559 D26205            JNC     IF8     ;BRIF NO EQUAL TEST
055C CDCE18            CALL    FTEST   ;GET STATUS OF FACC
055F CA8105            JZ      TRUE    ;BRIF LEFT=RIGHT
0562 3A2522    IF8:    LDA     REL     ;LOAD RELATION
0565 E602              ANI     02H     ;MASK IT
0567 CA7005            JZ      IF9     ;BRIF NO >
056A CDCE18            CALL    FTEST   ;GET STATUS OF FACC
056D FA8105            JM      TRUE    ;BRIF GT
0570 3A2522    IF9:    LDA     REL     ;LOAD RELATION
0573 E604              ANI     04H     ;MASK IT
0575 CA0B02            JZ      FALSE   ;BRIF NO <
0578 CDCE18            CALL    FTEST   ;GET STATUS OF FACC
057B FA0B02            JM      FALSE   ;BRIF GT
057E CA0802            JZ      FALSE   ;BRIF ZERO (NOT EQUAL)
0581 2A5222    TRUE:   LHLD    ADDR1   ;GET POINTER TO STATEMENT
0584 11D01E            LXI     D,GOTOL ;POINT 'GO TO'
0587 D7                RST     2       ;GO COMPARE
0588 CAF602            JZ      GOTO    ;BRIF IF ... GOTO NN
058B 2A5222            LHLD    ADDR1   ;GET POINTER TO STATEMENT
058E 11AF1E            LXI     D,GOSBL ;POINT LITERAL
0591 D7                RST     2       ;GO COMPARE
0592 CA3A03            JZ      GOSUB   ;BRIF IF ... GOSUB NN
0595 2A5222            LHLD    ADDR1   ;GET POINTER TO STATEMENT
0598 11921D            LXI     D,THENL ;GET ADDR 'THEN'
059B D7                RST     2       ;GO COMPARE
059C C20F1C            JNZ     SNERR   ;BRIF ERROR
059F CD2A18            CALL    NUMER   ;TEST IF NUMERIC
05A2 CAF602            JZ      GOTO    ;BRIF IT IS
05A5 C33802            JMP     RUN4    ;ELSE, MAY BE ANY STMT
0208 =         FALSE   EQU     RUN
05A8 21CE20    IFF:    LXI     H,IOBUF ;POINT PRIOR
05AB 46                MOV     B,M     ;GET LEN
05AC 112021            LXI     D,STRIN ;POINT THIS
05AF 1A                LDAX    D       ;GET LEN
05B0 4F                MOV     C,A     ;SAVE IT
05B1 13        IFG:    INX     D       ;POINT NEXT
05B2 23                INX     H       ;DITTO
05B3 78                MOV     A,B     ;GET LEFT LEN
05B4 B7                ORA     A       ;TEST IT
05B5 C2BA05            JNZ     IFH     ;BRIF NOT ZERO
05B8 3620              MVI     M,' '   ;EXTEND WITH SPACE
05BA 79        IFH:    MOV     A,C     ;GET RIGHT LEN
05BB B7                ORA     A       ;TEST IT
05BC C2C205            JNZ     IFI     ;BRIF NOT ZERO
05BF 3E20              MVI     A,' '   ;GET SPACE
05C1 12                STAX    D       ;EXTEND
05C2 1A        IFI:    LDAX    D       ;GET RIGHT CHAR
05C3 BE                CMP     M       ;TEST WITH LEFT
05C4 DAE705            JC      IFM     ;BRIF LEFT>RIGHT
05C7 C2EC05            JNZ     IFN     ;BRIF LEFT<RIGHT
05CA 78                MOV     A,B     ;GET LEFT COUNT
05CB 3D                DCR     A       ;SUBT ONE
```

```
05CC FAD005             JM      IFJ     ;BRIF WAS ZERO
05CF 47                 MOV     B,A     ;UPDATE CTR
05D0 79         IFJ:    MOV     A,C     ;GET RIGHT LEN
05D1 3D                 DCR     A       ;SUBT ONE
05D2 FAD605             JM      IFK     ;BRIF WAS ZERO
05D5 4F                 MOV     C,A     ;UPDT CTR
05D6 78         IFK:    MOV     A,B     ;GET LEFT LEN
05D7 B1                 ORA     C       ;COMPARE TO RIGHT
05D8 C2B105             JNZ     IFG     ;BRIF BOTH NOT ZERO
05DB 0601               MVI     B,1     ;SET SW= EQUAL
05DD 3A2522     IFL:    LDA     REL     ;GET RELATION
05E0 A0                 ANA     B       ;AND WITH RESULT
05E1 CA0802             JZ      FALSE   ;BRIF FALSE
05E4 C38105             JMP     TRUE    ;ELSE, TRUE
05E7 0602       IFM:    MVI     B,2     ;SET CODE
05E9 C3DD05             JMP     IFL     ;JUMP
05EC 0604       IFN:    MVI     B,4     ;SET CODE
05EE C3DD05             JMP     IFL     ;JUMP
                        ;PAGE
```

```
                        ;
05F1 =                  LET     EQU     $
                        ;
                        ;
                        ; STMT: [LET] VAR = EXPR
                        ;
                        ;
05F1 CD4F18                     CALL    GETS8   ;GO GET ADDRESS OF VARIABLE
05F4 C5                         PUSH    B       ;SAVE NAME
05F5 D5                         PUSH    D       ;SAVE ADDRESS
05F6 CF                         RST     1       ;GET NEXT CHAR
05F7 FE3D                       CPI     '='     ;TEST FOR EQUAL SIGN
05F9 CA0C06                     JZ      LET1    ;BRIF IS
05FC 3A7620                     LDA     EDSW    ;GET MODE SW
05FF B7                         ORA     A       ;TEST IT
0600 CA0F1C                     JZ      SNERR   ;BRIF LET ERROR
0603 21731D                     LXI     H,WHATL ;POINT LITERAL
0606 CD8D19                     CALL    TERMM   ;GO PRINT IT
0609 C3C900                     JMP     GETCM   ;GO TO COMMAND
060C 23                 LET1:   INX     H       ;POINT NEXT
060D CD800F                     CALL    EXPR    ;GO EVALUATE EXPRESSION
0610 CD941A                     CALL    EOL     ;ERROR IF NOT END-OF-LINE
0613 E1                         POP     H       ;RESTORE ADDRESSS
0614 D1                         POP     D       ;RESTORE NAME
0615 7B                         MOV     A,E     ;GET TYPE
0616 B7                         ORA     A       ;TEST IT
0617 3A8E22                     LDA     NS      ;GET RESULT TYPE
061A FA2606                     JM      LET2    ;BRIF STRING
061D FEE3                       CPI     0E3H    ;TEST IF NUMERIC
061F C20F1C                     JNZ     SNERR   ;BRIF MIXED MODE
0622 DF                         RST     3       ;GO STORE VARIABLE
0623 C30B02                     JMP     RUN     ;CONTINUE
0626 FEE7               LET2:   CPI     0E7H    ;TEST IF STRING
0628 C20F1C                     JNZ     SNERR   ;BRIF MIXED MODE
062B CD3106                     CALL    LET2A   ;GO STORE IT
062E C30B02                     JMP     RUN     ;CONTINUE
                        ;
0631 112021             LET2A:  LXI     D,STRIN ;POINT STRING BUFFER
0634 1A                         LDAX    D       ;GET NEW LEN
0635 96                         SUB     M       ;MINUS OLD LEN
0636 CA8606                     JZ      LET8    ;BRIF SAME LENGTH
0639 54                         MOV     D,H     ;COPY H,L
063A 5D                         MOV     E,L     ;TO D,E
063B 7E                         MOV     A,M     ;GET LEN
063C 3C                         INR     A       ;TRUE LEN
063D 13                 LET3:   INX     D       ;POINT NEXT
063E 3D                         DCR     A       ;DECR CTR
063F C23D06                     JNZ     LET3    ;LOOP
0642 13                         INX     D       ;SKIP
0643 13                         INX     D       ;AGAIN
0644 1A                         LDAX    D       ;GET LO NAM
0645 4F                         MOV     C,A     ;SAVE
0646 13                         INX     D       ;GET HI NAME
0647 1A                         LDAX    D       ;LOAD IT
0648 47                         MOV     B,A     ;SAVE
0649 C5                         PUSH    B       ;SAVE NAME
064A 2B                         DCX     H       ;POINT NEXT ENTRY
0648 7E                 LET4:   MOV     A,M     ;GET NEXT
```

```
064C B7          ORA     A       ;TEST IF END
064D CA6406      JZ      LET6    ;BRIF IS
0650 E5          PUSH    H       ;SAVE H,L
0651 2B          DCX     H       ;SKIP NEXT
0652 2B          DCX     H       ;POINT LEN
0653 46          MOV     B,M     ;GET HI LEN
0654 2B          DCX     H       ;POINT LO
0655 4E          MOV     C,M     ;GET LO LEN
0656 E1          POP     H       ;RESTORE H,L
0657 7E    LET5: MOV     A,M     ;GET A BYTE
0658 12          STAX    D       ;COPY
0659 2B          DCX     H       ;POINT NEXT
065A 1B          DCX     D       ;DITTO
065B 03          INX     B       ;ADD TO CTR
065C 78          MOV     A,B     ;GET HI
065D B1          ORA     C       ;TEST IF ZERO
065E C25706      JNZ     LET5    ;LOOP
0661 C34B06      JMP     LET4    ;CONTINUE
0664 EB    LET6: XCHG            ;PUT NEW ADDR TO H,L
0665 C1          POP     B       ;GET NAME
0666 70          MOV     M,B     ;STORE HI BYTE
0667 2B          DCX     H       ;POINT NEXT
0668 71          MOV     M,C     ;STORE LO
0669 112021      LXI     D,STRIN ;GET NEW LEN
066C 1A          LDAX    D       ;LOAD IT
066D 06FF        MVI     B,0FFH  ;INIT HI COMPLEMENT
066F C605        ADI     5       ;COMPUTE ENTRY LENGTH
0671 CA7906      JZ      LET7    ;BRIF 256 BYTES
0674 D27906      JNC     LET7    ;BRIF LESS 256
0677 06FE        MVI     B,0FEH  ;SET BIT OFF
0679 2F    LET7: CMA             ;1'S COMPLEMENT
067A 3C          INR     A       ;THEN 2'S
067B 4F          MOV     C,A     ;SAVE LO LEN
067C 2B          DCX     H       ;POINT NEXT
067D 70          MOV     M,B     ;STORE HI LEN
067E 2B          DCX     H       ;POINT NEXT
067F 71          MOV     M,C     ;STORE LO LEN
0680 E7          RST     4       ;ADJUST H,L
0681 03          DB      3
0682 09          DAD     B       ;COMPUTE END OF ENTRY
0683 3600        MVI     M,0     ;MARK NEW END
0685 23          INX     H       ;POINT 1ST BYTE
0686 1A    LET8: LDAX    D       ;GET LEN
0687 3C          INR     A       ;TRUE LEN
0688 47          MOV     B,A     ;SAVE LEN
0689 1A    LET9: LDAX    D       ;GET A BYTE
068A 77          MOV     M,A     ;COPY IT
068B 23          INX     H       ;POINT NEXT
068C 13          INX     D       ;DITTO
068D 05          DCR     B       ;SUBT CTR
068E C28906      JNZ     LET9    ;LOOP
0691 C9          RET             ;RETURN
           ;PAGE
```

```
;
0692 =          NEXT    EQU     $
                ;
                ;
                ; STMT:  NEXT VAR
                ;
                ;
0692 CDC91B             CALL    VAR     ;GET VARIABLE NAME
0695 CD941A             CALL    EOL     ;ERROR IF NOT END-OF-LNE
0698 EB                 XCHG            ;FLIP/FLOP
0699 222322             SHLD    INDX    ;SAVE VAR NAME
069C E5                 PUSH    H       ;SAVE VAR NAME
069D 210020             LXI     H,FORNE ;POINT FOR/NEXT TABLE
06A0 46                 MOV     B,M     ;GET SIZE
06A1 78                 MOV     A,B     ;LOAD IT
06A2 B7                 ORA     A       ;TEST IT
06A3 CA1B1C             JZ      NXERR   ;BRIF TABLE EMPTY
06A6 23                 INX     H       ;POINT NEXT
06A7 D1                 POP     D       ;RESTORE VAR NAME
06A8 7E         NEXT1:  MOV     A,M     ;GET 1ST BYTE
06A9 23                 INX     H       ;POINT NEXT
06AA BA                 CMP     D       ;COMPARE
06AB C2B306             JNZ     NEXT2   ;BRIF NOT EQUAL
06AE 7E                 MOV     A,M     ;GET 2ND BYTE
06AF BB                 CMP     E       ;COMPARE
06B0 CABC06             JZ      NEXT3   ;BRIF EQUAL
06B3 E7         NEXT2:  RST     4       ;ADJUST H,L
06B4 0D                 DB      13
06B5 05                 DCR     B       ;DECR COUNT
06B6 C2A806             JNZ     NEXT1   ;LOOP
06B9 C31B1C             JMP     NXERR   ;GO PUT ERROR MSG
06BC 3A0020     NEXT3:  LDA     FORNE   ;GET ORIG COUNT
06BF 90                 SUB     B       ;MINUS REMAIN
06C0 3C                 INR     A       ;PLUS ONE
06C1 320020             STA     FORNE   ;STORE NEW COUNT
06C4 23                 INX     H       ;POINT ADDR
06C5 E5                 PUSH    H       ;SAVE H,L ADDR
06C6 CD3418             CALL    SEARC   ;GO GET ADDR OF INDEX
06C9 EB                 XCHG            ;PUT TO H,L
06CA 225222             SHLD    ADDR1   ;SAVR IT
06CD EF                 RST     5       ;LOAD INDEX
06CE E1                 POP     H       ;GET H,L (TBL)
06CF E5                 PUSH    H       ;RE-SAVE
06D0 CD3716             CALL    FADD    ;ADD STEP VALUE
06D3 212722             LXI     H,TVAR1 ;POINT TEMP AREA
06D6 DF                 RST     3       ;SAVE NEW INDEX
06D7 E1                 POP     H       ;GET H,L (TBL)
06D8 E5                 PUSH    H       ;RE-SAVE
06D9 E7                 RST     4       ;GET LEN TO NEXT
06DA 04                 DB      4
06DB CD0C17             CALL    FSUB    ;SUBTRACT TO VALUE
06DE CAF806             JZ      NEXT6   ;BRIF ZERO
06E1 E1                 POP     H       ;GET H,L (PTR TO STEP)
06E2 E5                 PUSH    H       ;RE-SAVE
06E3 7E                 MOV     A,M     ;GET SIGN&EXPONENT OF STEP
06E4 B7                 ORA     A       ;TEST IT
06E5 3A5822             LDA     FACC    ;GET SIGN & EXPON OF DIFF
```

```
06E8 FAF706             JM      NEXT5   ;BRIF NEGATIVE
06EB B7                 ORA     A       ;TEST SIGN OF DIFF
06EC FAF806             JM      NEXT6   ;BRIF LESS THAN TO-EXPR
06EF 210020     NEXT7:  LXI     H,FORNE ;GET ADDR TABLE
06F2 35                 DCR     M       ;SUBTRACT ONE FROM COUNT
06F3 D1                 POP     D       ;ADJUST STACK
06F4 C30B02             JMP     RUN     ;GO STMT AFTER NEXT
06F7 B7         NEXT5:  ORA     A       ;TEST SIGN OF DIFFERENCE
06F8 FAEF06             JM      NEXT7   ;BRIF END OF LOOP
06FB E1         NEXT6:  POP     H       ;GET PTR TO TBL
06FC E7                 RST     4       ;ADJUST H,L
06FD 08                 DB      8
06FE 56                 MOV     D,M     ;GET HI BYTE
06FF 23                 INX     H       ;POINT NEXT
0700 5E                 MOV     E,M     ;GET LO BYTE
0701 23                 INX     H       ;POINT NEXT
0702 7E                 MOV     A,M     ;GET HI BYTE
0703 327122             STA     STMT+1  ;SAVE
0706 23                 INX     H       ;POINT NEXT
0707 7E                 MOV     A,M     ;GET LO BYTE
0708 327022             STA     STMT    ;SAVE
070B E8                 XCHG            ;H,L = ADDR OF STMT AFTR FOR
070C CD941A             CALL    EOL     ;SETUP MULTI PTR
070F 2A7022             LHLD    STMT    ;GET ADDR OF FOR STMT
0712 23                 INX     H       ;POINT LINE NUM
0713 228922             SHLD    LINE    ;SAVE ADDR LINE
0716 212722             LXI     H,TVAR1 ;POINT UPDTED VALUE
0719 EF                 RST     5       ;GO LOAD IT
071A 2A5222             LHLD    ADDR1   ;GET ADDR OF INDEX
071D DF                 RST     3       ;GO STORE IT
071E C30B02             JMP     RUN     ;CONTINUE WITH STMT AFTER FOR
                ;PAGE
```

```
                        ;
0721 =                  INPUT   EQU     $
                        ;
                        ;
                        ; STMT:  INPUT VAR [, VAR, VAR]
                        ;
                        ;
0721 11841D                     LXI     D,LLINE ;POINT 'LINE'
0724 E5                         PUSH    H       ;SAVE H,L ADDR
0725 D7                         RST     2       ;GO COMPARE
0726 CAA507                     JZ      INPL    ;BRIF EQUAL
0729 D1                         POP     D       ;ELSE, RESTORE H,L ADDR
072A 21CE20                     LXI     H,IOBUF ;GET ADDR OF BUFFER
072D 225222                     SHLD    ADDR1   ;SAVE ADDR
0730 3600                       MVI     M,0     ;MARK BUFFER EMPTY
0732 EB                         XCHG            ;FLIP/BACK
0733 CF                 INPU1:  RST     1       ;SKIP SPACES
0734 FE27                       CPI     27H     ;TEST IF QUOTE
0736 CA3E07                     JZ      INPU2   ;BRIF IS
0739 FE22                       CPI     '"'     ;TEST IF INPUT LITERAL
073B C26107                     JNZ     INPU6   ;BRIF NOT
073E 4F                 INPU2:  MOV     C,A     ;SAVE DELIM
073F 11CE20                     LXI     D,IOBUF ;POINT BUFFER
0742 23                 INPU3:  INX     H       ;POINT NEXT
0743 7E                         MOV     A,M     ;LOAD IT
0744 B9                         CMP     C       ;TEST IF END
0745 CA4D07                     JZ      INPU4   ;BRIF IS
0748 12                         STAX    D       ;PUT TO BUFF
0749 13                         INX     D       ;POINT NEXT
074A C34207                     JMP     INPU3   ;LOOP
074D 23                 INPU4:  INX     H       ;SKIP TRAILING QUOTE
074E EB                         XCHG            ;PUT ADDR TO H,L
074F 36FE                       MVI     M,0FEH  ;MARK END
0751 CD8519                     CALL    TERMO   ;GO PRINT PROMPT
0754 EB                         XCHG            ;GET H,L
0755 CF                         RST     1       ;SKIP TO NON BLANK
0756 FE2C                       CPI     ','     ;TEST IF COMMA
0758 CA6007                     JZ      INPU5   ;BRIF IS
075B FE3B                       CPI     ';'     ;TEST IF COMMA
075D C26107                     JNZ     INPU6   ;BRIF NOT
0760 23                 INPU5:  INX     H       ;SKIP IT
0761 CD4F18             INPU5:  CALL    GETS8   ;GO GET VAR ADDR
0764 E5                         PUSH    H       ;SAVE H ADDR
0765 D5                         PUSH    D       ;SAVE VAR ADDR
0766 2A5222                     LHLD    ADDR1   ;GET ADDR PREV BUFFER
0769 7E                         MOV     A,M     ;LOAD CHAR
076A FE2C                       CPI     ','     ;TEST IF COMMA
076C 23                         INX     H       ;POINT NEXT
076D CA7507                     JZ      INPU7   ;BRIF CONTINUE FROM PREV
0770 3E3F                       MVI     A,'?'   ;LOAD PROMPT
0772 CD0419                     CALL    TERMI   ;GO READ FROM TTY
0775 CF                 INPU7:  RST     1       ;SKIP SPACES
0776 79                         MOV     A,C     ;GET LO NAME
0777 87                         ORA     A       ;TEST IT
0778 FA9C07                     JM      INPUA   ;BRIF STRING
077B CD2E14                     CALL    FIN     ;GO CONVERT TO FLOATING
077E CF                         RST     1       ;SKIP SPACES
077F FE2C                       CPI     ','     ;TEST IF COMMA
```

```
0781 CA8807             JZ      INPU8   ;BRIF IS
0784 B7                 ORA     A       ;TEST IF END OF LINE
0785 C21F1C             JNZ     CVERR   ;BRIF ERROR
0788 225222     INPU8:  SHLD    ADDR1   ;SAVE ADDRESS
078B E1                 POP     H       ;GET VAR ADDR
078C DF                 RST     3       ;GO STORE THE NUMBER
078D E1         INPU9:  POP     H       ;RESTORE STMT POINTER
078E 7E                 MOV     A,M     ;GET CHAR
078F FE2C               CPI     ','     ;TEST FOR COMMA
0791 23                 INX     H       ;POINT NEXT
0792 CA3307             JZ      INPU1   ;RECURSIVE IF COMMA
0795 2B                 DCX     H       ;POINT BACK
0796 CD941A     INPU8:  CALL    EOL     ;ERROR IF NOT END OF LINE
0799 C30802             JMP     RUN     ;CONTINUE NEXT STMT
079C CD0D18     INPUA:  CALL    GETST   ;GO GET THE STRING
079F 225222             SHLD    ADDR1   ;SAVE ADDRESS
07A2 C38D07             JMP     INPU9   ;CONTINUE
                ;
07A5 =          INPL    EQU     $
                ;
                ;
                ; STMT: INPUT LINE A$
                ;
                ;
07A5 D1                 POP     D       ;DUMMY POP TO ADJUST STACK
07A6 CDC91B             CALL    VAR     ;GET STRING NAME
07A9 78                 MOV     A,E     ;LOAD LO BYTE
07AA B7                 ORA     A       ;TEST IT
07AB F20F1C             JP      SNERR   ;BRIF NOT STRING VARIABLE
07AE CD341B             CALL    SEARC   ;ELSE, GET ADDRESS
07B1 D5                 PUSH    D       ;SAVE ON STACK
07B2 CD941A             CALL    EOL     ;ERROR IF NOT END-OF-LINE
07B5 3E01               MVI     A,1     ;GET ON SETTING
07B7 327420             STA     ILSW    ;SET INPUT LINE SWITCH
07BA 3E3F               MVI     A,'?'   ;LOAD PROMPT
07BC CD0419             CALL    TERMI   ;GO READ A LINE
07BF 0600               MVI     B,0     ;INIT COUNT
07C1 112121             LXI     D,STRIN+1       ;POINT STRING BUFFER
07C4 21CF20             LXI     H,IOBUF+1       ;POINT INPUT BUFFER
07C7 7E         INPL1:  MOV     A,M     ;GET NEXT BYTE
07C8 B7                 ORA     A       ;TEST IT
07C9 CAD307             JZ      INPL2   ;BRIF END
07CC 04                 INR     B       ;ADD TO COUNT
07CD 12                 STAX    D       ;PUT TO STRING BUFF
07CE 13                 INX     D       ;POINT NEXT
07CF 23                 INX     H       ;DITTO
07D0 C3C707             JMP     INPL1   ;LOOP
07D3 327420     INPL2:  STA     ILSW    ;RESET SWITCH
07D6 78                 MOV     A,B     ;GET COUNT
07D7 322021             STA     STRIN   ;SET STRING LENGTH
07DA E1                 POP     H       ;GET ADDRESS OF VARIABLE
07DB CD3106             CALL    LET2A   ;GO STORE THE STRING
07DE C30802             JMP     RUN     ;GO NEXT STMT
                        ;PAGE
```

```
                        ;
07E1 =                  READ    EQU     $
                        ;
                        ; STMT: READ VAR [,VAR ...]
                        ;
07E1 CF                         RST     1       ;SKIP BLANKS
07E2 CD4F18                     CALL    GETS8   ;GET VAR ADDR
07E5 E5                         PUSH    H       ;SAVE H,L
07E6 D5                         PUSH    D       ;SAVE D,E
07E7 2A8F22                     LHLD    DATAP   ;GET DATA STMT POINTER
07EA 7E                         MOV     A,M     ;LOAD THE CHAR
07EB B7                         ORA     A       ;TEST IF END OF STMT
07EC C20B08                     JNZ     READ2   ;BRIF NOT END OF STMT
07EF 23                         INX     H       ;POINT START NEXT STMT
07F0 7E                 READ1:  MOV     A,M     ;LOAD LEN
07F1 228F22                     SHLD    DATAP   ;SAVE ADDR
07F4 B7                         ORA     A       ;TEST IF END OF PGM
07F5 CA171C                     JZ      DAERR   ;BRIF OUT OF DATA
07F8 E7                         RST     4       ;ADJUST H,L
07F9 03                         DB      3
07FA 119B1E                     LXI     D,DATAL ;POINT 'DATA'
07FD D7                         RST     2       ;COMPARE
07FE CA0B08                     JZ      READ2   ;BRIF IT IS DATA STMT
0801 2A8F22                     LHLD    DATAP   ;GET ADDR START
0804 5E                         MOV     E,M     ;GET LEN CODE
0805 1600                       MVI     D,0     ;CLEAR D
0807 19                         DAD     D       ;POINT NEXT STMT
0808 C3F007                     JMP     READ1   ;LOOP NEXT STMT
080B CF                 READ2:  RST     1       ;SKIP SPACES
080C 79                         MOV     A,C     ;LOAD LO NAME
080D B7                         ORA     A       ;TEST IT
080E FA3308                     JM      READ6   ;BRIF STRING
0811 CD2E14                     CALL    FIN     ;GO CONVERT VALUE
0814 7E                         MOV     A,M     ;GET CHAR WHICH STOPPED US
0815 FE2C                       CPI     ','     ;TEST IF COMMA
0817 C22C08                     JNZ     READ5   ;BRIF NOT
081A 23                         INX     H       ;POINT NEXT
081B 228F22             READ3:  SHLD    DATAP   ;SAVE ADDRESS
081E E1                         POP     H       ;RESTORE ADDR OF VAR
081F DF                         RST     3       ;STORE THE VALUE
0820 E1                 READ4:  POP     H       ;RESTORE POINTER TO STM
0821 7E                         MOV     A,M     ;GET THE CHAR
0822 FE2C                       CPI     ','     ;TEST IF COMMA
0824 23                         INX     H       ;POINT NEXT
0825 CAE107                     JZ      READ    ;RECURSIVE IF IT IS
0828 2B                         DCX     H       ;RESET
0829 C39607                     JMP     INPUB   ;CONTINUE
082C B7                 READ5:  ORA     A       ;TEST IF END OF STMT
082D CA1B08                     JZ      READ3   ;BRIF OK
0830 C31F1C                     JMP     CVERR   ;GO PROCESS ERROR
0833 CD0D18             READ6:  CALL    GETST   ;GO GET STRING
0836 7E                         MOV     A,M     ;GET CHAR
0837 FE2C                       CPI     ','     ;TEST IF COMMA
0839 CA4308                     JZ      READ7   ;BRIF IS
083C B7                         ORA     A       ;TEST IF END
083D C22C08                     JNZ     READ5   ;BRIF NOT
0840 C34408                     JMP     READ8   ;GO AROUND
0843 23                 READ7:  INX     H       ;POINT PAST
```

```
0844 228F22     READ8:  SHLD    DATAP   ;SAVE ADDRESS
0847 C32008             JMP     READ4   ;CONTINUE
                ;
084A =          OUTP    EQU     $
                ;
                ; STMT: OUT ADDR,VALUE
                ;
                ;
084A CD800F             CALL    EXPR    ;GO EVALUATE ADDRESS
084D 7E                 MOV     A,M     ;GET DELIM
084E FE2C               CPI     ','     ;TEST IF COMMA
0850 C20F1C             JNZ     SNERR   ;BRIF NOT
0853 23                 INX     H       ;SKIP OVER COMMA
0854 CD661C             CALL    FBIN    ;CONVERT TO BINARY IN A-REG
0857 112022             LXI     D,OUTA  ;POINT INSTR
085A EB                 XCHG            ;PUT TO H,L
085B 36D3               MVI     M,0D3H  ;OUT INSTR
085D 23                 INX     H       ;POINT NEXT
085E 77                 MOV     M,A     ;PUT ADDR
085F 23                 INX     H       ;POINT NEXT
0860 36C9               MVI     M,0C9H  ;RET INSTR
0862 EB                 XCHG            ;RESTORE ORIG H,L
0863 CD800F             CALL    EXPR    ;GO EVAL DATA BYTE
0866 CD941A             CALL    EOL     ;ERROR IF NOT END OF STATMENT
0869 CD661C             CALL    FBIN    ;CONVERT TO BINARY
086C CD2022             CALL    OUTA    ;GO PUT THE BYTE
086F C30802             JMP     RUN     ;GO NEXT STMT
                        ;PAGE
```

```
                        ;
0872 =                  STOP    EQU     $
                        ;
                        ; STMT: STOP
                        ;
                        ;
0872 CD941A                     CALL    EOL     ;POINT END OF LINE
0875 212D1E                     LXI     H,STOPM ;POINT MESSAGE: "STOP AT LINE "
0878 CD8D19                     CALL    TERMM   ;GO WRITE IT
087B CDF11B                     CALL    PRLIN   ;GO PRINT LINE NUMBER
087E 3A7520                     LDA     RUNSW   ;GET RUN TYPE
0881 B7                         ORA     A       ;TEST IT
0882 C2C300                     JNZ     RDY     ;BRIF IMMED
0885 327422                     STA     MULTI   ;CLEAR MULTI SW
0888 2A7022                     LHLD    STMT    ;GET ADDR OF PREV STMT
088B 5E                         MOV     E,M     ;GET LEN
088C 1600                       MVI     D,0     ;CLEAR HI BYTE
088E 19                         DAD     D       ;POINT NEXT
088F 23                         INX     H       ;POINT LINE NUMBER
0890 228922                     SHLD    LINE    ;SAVE ADDR
0893 117720                     LXI     D,LINEN ;POINT AREA
0896 CD091A                     CALL    LINEO   ;GO CONVERT LINE NUMBER
0899 EB                         XCHG            ;FLIP TO H,L
089A 3600                       MVI     M,0     ;MARK END
089C C3C300                     JMP     RDY     ;GO TO READY MSG
                        ;
089F =                  RANDO   EQU     $
                        ;
                        ;
                        ; STMT: RANDOMIZE
                        ;
                        ;
089F CD941A                     CALL    EOL     ;ERROR IF NOT END-OF-LINE
08A2 3E01                       MVI     A,1     ;LOAD A ONE
08A4 328722                     STA     RNDSW   ;SET SWITCH = TRUE RANDOM
08A7 117F22                     LXI     D,TRNDX ;POINT 'TRUE' RANDOM NUMBERS
08AA 217722                     LXI     H,RNDX  ;POINT RECEIVE
08AD 0608                       MVI     B,8     ;LOOP CTR
08AF CD4D1C                     CALL    COPYD   ;GO MOVE IT
08B2 C30B02                     JMP     RUN     ;CONTINUE
                        ;
08B5 =                  ON      EQU     $
                        ;
                        ;
                        ; STMT: ON EXPR GOTO NNN NNNN NNNN
                        ;               GOSUB
                        ;
                        ;
08B5 CD800F                     CALL    EXPR    ;GO EVALUATE EXPRESSION
08B8 CD661C                     CALL    FBIN    ;GET BINARY NUMBER IN ACC
08BB B7                         ORA     A       ;TEST RESULT
08BC CA0F1C                     JZ      SNERR   ;BRIF ZERO (ERROR)
08BF 4F                         MOV     C,A     ;SAVE VALUE
08C0 0D                         DCR     C       ;LESS ONE
08C1 AF                         XRA     A       ;GET A ZERO
08C2 322522                     STA     REL     ;TURN OFF SWITCH
08C5 11D01E                     LXI     D,GOTOL ;POINT LITERAL
08C8 E5                         PUSH    H       ;SAVE H,L ADDRESS
```

```
08C9 D7            RST     2         ;GO COMPARE
08CA CAD808        JZ      ON3       ;BRIF ON...GOTO
08CD E1            POP     H         ;ELSE, RESTORE H,L
08CE 11AF1E        LXI     D,GOSBL   ;POINT LITERAL
08D1 D7            RST     2         ;GO COMPARE
08D2 C20F1C        JNZ     SNERR     ;BRIF ERROR
08D5 3E01          MVI     A,1       ;GET ON SETTING
08D7 322522        STA     REL       ;SET SWITCH
08DA E5            PUSH    H         ;DUMMY PUSH
08DB D1      ON3:  POP     D         ;ADJUST STACK
08DC 79      ON3A: MOV     A,C       ;GET COUNT
08DD B7            ORA     A         ;TEST IT
08DE CAFD08        JZ      ON6       ;BRIF VALUE 1
08E1 CF            RST     1         ;ELSE, SKIP BLANKS
08E2 B7            ORA     A         ;TEST IF END OF LINE
08E3 CA0F1C        JZ      SNERR     ;BRIF IS
08E6 FE2C          CPI     ','       ;TEST IS COMMA
08E8 C2EF08        JNZ     ON4       ;BRIF NOT
08EB 23            INX     H         ;SKIP COMMA
08EC C3DC08        JMP     ON3A      ;CONTINUE
08EF CD2A1B  ON4:  CALL    NUMER     ;GO TEST IF NUMERIC
08F2 C2F908        JNZ     ON5       ;BRIF NOT
08F5 23            INX     H         ;POINT NEXT
08F6 C3EF08        JMP     ON4       ;LOOP
08F9 0D      ON5:  DCR     C         ;SUB ONE FROM COUNT
08FA C2DC08        JNZ     ON3A      ;LOOP TILL JUST BEFORE STMT#
08FD CDAD1A  ON6:  CALL    NOTEO     ;ERROR IF NOT END-OF-LINE
0900 FE2C          CPI     ','       ;TEST IF COMMA
0902 C20909        JNZ     ON7       ;BRIF NOT
0905 23            INX     H         ;POINT NEXT
0906 C3FD08        JMP     ON6       ;LOOP
0909 CD2A1B  ON7:  CALL    NUMER     ;TEST IF NUMERIC
090C C20F1C        JNZ     SNERR     ;BRIF NOT
090F CDB51A        CALL    PACK      ;GET THE LINE NUMBER
0912 7E      ON8:  MOV     A,M       ;GET NEXT CHAR
0913 CDA81A        CALL    TSTEL     ;TEST IF END STMT
0916 CA1D09        JZ      ON9       ;BRIF END
0919 23            INX     H         ;POINT NEXT
091A C31209        JMP     ON8       ;LOOP
091D CD941A  ON9:  CALL    EOL       ;SET END OF LINE POINTERS
0920 3A2522        LDA     REL       ;GET TYPE (GOTO OR GOSUB)
0923 B7            ORA     A         ;TEST IT
0924 C24303        JNZ     GOSU1     ;BRIF GOSUB
0927 C30603        JMP     GOTO2     ;BR TO GOTO LOOKUP
                   ;PAGE
```

```
                        ;
092A =                  CHANG   EQU     $
                        ;
                        ; STATEMENT: CHANGE A$ TO X     - OR -
                        ;
                        ;              CHANGE X TO A$
                        ;
092A CDC91B                     CALL    VAR     ;NEXT WORD MUST BE VAR
092D 7B                         MOV     A,E     ;TEST TYPE
092E B7                         ORA     A       ;SET FLAGS
092F F26809                     JP      CHA2    ;BRIF NON-STRING
0932 CD341B                     CALL    SEARC   ;GET ADDR
0935 D5                         PUSH    D       ;SAVE IT
0936 11D21E                     LXI     D,TOLIT ;POINT 'TO'
0939 D7                         RST     2       ;COMPARE
093A C20F1C                     JNZ     SNERR   ;BRIF ERROR
093D CDC91B                     CALL    VAR     ;GET NEXT VARIABLE
0940 7A                         MOV     A,D     ;GET HI NAME
0941 F680                       ORI     80H     ;SET MASK FOR ARRAY
0943 57                         MOV     D,A     ;REPLACE
0944 CD341B                     CALL    SEARC   ;GET ADDRESS
0947 E7                         RST     4       ;POINT START OF ELEMENT 0,0
0948 F5                         DB      -11 AND 0FFH
0949 D1                         POP     D       ;GET PTR TO STMT
094A EB                         XCHG            ;FLIP
094B CD941A                     CALL    EOL     ;NEXT MUST BE E-O-L
094E EB                         XCHG            ;FLIP AGAIN
094F D1                         POP     D       ;GET ADDR STRING
 950 1A                         LDAX    D       ;GET COUNT
 951 47                         MOV     B,A     ;SAVE IT
0952 04                         INR     B       ;BUMP
0953 C5                 CHA1:   PUSH    B       ;SAVE CTR
0954 D5                         PUSH    D       ;SAVE ADDR STRING
0955 E5                         PUSH    H       ;SAVE ADDR NUM
0956 CD1A0D                     CALL    FDEC    ;CONVERT TO F.P.
0959 E1                         POP     H       ;GET ADDR
095A DF                         RST     3       ;STORE IT
095B E7                         RST     4       ;POINT TO NEXT
095C F8                         DB      -8 AND 0FFH
095D D1                         POP     D       ;RESTORE STRING
095E C1                         POP     B       ;AND CTR
095F 13                         INX     D       ;POINT NEXT CHAR
0960 1A                         LDAX    D       ;LOAD IT
0961 05                         DCR     B       ;DECR CTR
0962 C25309                     JNZ     CHA1    ;LOOP
0965 C30B02                     JMP     RUN
                        ;
                        ;
0968 7A                 CHA2:   MOV     A,D     ;GET HI NAME
0969 F680                       ORI     80H     ;MAKE ARRAY NAME
096B 57                         MOV     D,A     ;SAVE
096C CD3418                     CALL    SEARC   ;GET ADDR
096F E7                         RST     4       ;POINT ELEMENT 0,0
 970 F5                         DB      -11 AND 0FFH
 971 E3                         XTHL            ;SAVE ON STACK
 972 11D21E                     LXI     D,TOLIT ;POINT 'TO'
0975 D7                         RST     2       ;COMPARE
0976 C20F1C                     JNZ     SNERR   ;BRIF ERROR
```

```
0979 CDC918             CALL    VAR         ;GET NAME
097C 7B                 MOV     A,E         ;GET TYPE
097D B7                 ORA     A           ;SET FLAGS
097E F20F1C             JP      SNERR       ;BRIF NOT STRING
0981 CD941A             CALL    EOL         ;BRIF NOT E-O-L
0984 CD3418             CALL    SEARC       ;GET ADDR
0987 E1                 POP     H           ;GET ADDR VAR
0988 D5                 PUSH    D           ;SAVE D,E
0989 112021             LXI     D,STRIN     ;POINT STRING BUFFER
098C D5                 PUSH    D           ;SAVE IT
098D EF                 RST     5           ;LOAD IT
098E E7                 RST     4           ;POINT NEXT
098F F8                 DB      -8 AND 0FFH
0990 E5                 PUSH    H           ;SAVE H,L
0991 CD661C             CALL    FBIN        ;CONVERT
0994 E1                 POP     H           ;RESTORE
0995 D1                 POP     D           ;DITTO
0996 47                 MOV     B,A         ;SAVE COUNT
0997 04                 INR     B           ;BUMP IT
0998 12         CHA3:   STAX    D           ;PUT TO STRING
0999 13                 INX     D           ;POINT NEXT STR LOC.
099A C5                 PUSH    B           ;SAVE CTRS
099B D5                 PUSH    D           ;AND ADDR
099C EF                 RST     5           ;LOAD NEXT
099D E7                 RST     4           ;POINT NEXT
099E F8                 DB      -8 AND 0FFH
099F E5                 PUSH    H           ;AND H ADDR
09A0 CD661C             CALL    FBIN        ;CONVERT
09A3 E1                 POP     H           ;RESTORE H,L
09A4 D1                 POP     D           ;AND D,E
09A5 C1                 POP     B           ;AND CTRS
09A6 05                 DCR     B           ;DECR CTR
09A7 C29809             JNZ     CHA3        ;LOOP
09AA E1                 POP     H           ;GET ADDR OF VAR (STRING)
09AB CD3106             CALL    LET2A       ;GO STORE IT
09AE C30802             JMP     RUN         ;CONTINUE
                        ;PAGE
```

```
                        ;
09B1 =                  DIM     EQU     $
                        ;
                        ; STMT: DIM VAR(A,B),...
                        ;
                        ;
09B1 CDC918             CALL    VAR     ;GO GET VAR NAME
09B4 F20F1C             JP      SNERR   ;BRIF NO (
09B7 CD3418             CALL    SEARC   ;GO LOCATE THE VAR
09BA E3                 XTHL            ;PUT ADDR IN STACK, GET PTR TO (
09BB F5                 PUSH    PSW     ;SAVE STATUS
09BC 3EFF               MVI     A,0FFH  ;TURN ON SW
09BE 327220             STA     DIMSW   ;SET IT
09C1 CD800F             CALL    EXPR    ;GO EVALUATE
09C4 F1                 POP     PSW     ;GET STATUS
09C5 E3                 XTHL            ;SWAP PTRS
09C6 D5                 PUSH    D       ;SAVE ROW NUMBER
09C7 C5                 PUSH    B       ;SAVE COL NUMBER
09C8 03                 INX     B       ;INCREMENT COLUMNS
09C9 13                 INX     D       ;AND ROWS
09CA E5                 PUSH    H       ;SAVE H,L
09CB F5                 PUSH    PSW     ;RESAVE STATUS
09CC 210000             LXI     H,0     ;GET A ZERO
09CF 19         DIM1:   DAD     D       ;TIMES ONE
09D0 0B                 DCX     B       ;DCR COLS
09D1 78                 MOV     A,B     ;GET HI
09D2 B1                 ORA     C       ;PLUS LO
09D3 C2CF09             JNZ     DIM1    ;LOOP
09D6 F1                 POP     PSW     ;GET STATUS
09D7 D1                 POP     D       ;GET ADDRESS
09D8 29                 DAD     H       ;TIMES TWO
09D9 29                 DAD     H       ;TIMES FOUR
09DA 010800             LXI     B,8     ;PLUS 2 (NAME AND DISP)
09DD FA1D0A             JM      REDIM   ;GO RE-DIMENSION
09E0 E5                 PUSH    H       ;SAVE PRODUCT
09E1 09                 DAD     B       ;ADD IT
09E2 EB                 XCHG            ;FLIP/FLOP
09E3 2B                 DCX     H       ;POINT LO NAME
09E4 2B                 DCX     H       ;POINT HI DISP
09E5 7B                 MOV     A,E     ;GET LO
09E6 2F                 CMA             ;COMPLEMENT
09E7 C601               ADI     1       ;PLUS ONE
09E9 5F                 MOV     E,A     ;RESTORE
09EA 7A                 MOV     A,D     ;GET HI
09EB 2F                 CMA             ;COMPLEMENT
09EC CE00               ACI     0       ;PLUS CARRY
09EE 77                 MOV     M,A     ;STORE IT
09EF 2B                 DCX     H       ;POINT NEXT
09F0 73                 MOV     M,E     ;STORE LO
09F1 EB                 XCHG            ;SAVE IN D,E
09F2 E1                 POP     H       ;GET PRODUCT
09F3 44                 MOV     B,H     ;COPY H,L
09F4 4D                 MOV     C,L     ;TO B,C
09F5 EB                 XCHG            ;GET LOCAT
09F6 D1                 POP     D       ;GET COLUMNS
09F7 2B                 DCX     H       ;POINT NEXT
09F8 72                 MOV     M,D     ;MOVE LO COL
09F9 2B                 DCX     H       ;POINT NEXT
```

```
09FA 73                 MOV     M,E     ;MOVE HI COL
09FB D1                 POP     D       ;GET ROWS
09FC 2B                 DCX     H       ;POINT NEXT
09FD 72                 MOV     M,D     ;MOVE HI ROW
09FE 2B                 DCX     H       ;POINT NEXT
09FF 73                 MOV     M,E     ;MOVE LO ROW
0A00 2B                 DCX     H       ;POINT NEXT
0A01 3600       DIM2:   MVI     M,0     ;CLEAR ONE BYTE
0A03 2B                 DCX     H       ;POINT NEXT
0A04 0B                 DCX     B       ;DECR CTR
0A05 78                 MOV     A,B     ;GET HI
0A06 B1                 ORA     C       ;PLUS LO
0A07 C2010A             JNZ     DIM2    ;LOOP
0A0A 3600               MVI     M,0     ;MARK END
0A0C E1         DIM3:   POP     H       ;GET PTR TO STMT
0A0D 7E                 MOV     A,M     ;LOAD CHAR
0A0E FE2C               CPI     ','     ;TEST IF COMMA
0A10 C2170A             JNZ     DIM4    ;BRIF NOT
0A13 23                 INX     H       ;SKIP IT
0A14 C3B109             JMP     DIM     ;CONTINUE
0A17 CD941A     DIM4:   CALL    EOL     ;TEST END OF LINE
0A1A C30B02             JMP     RUN     ;CONTINUE WITH PROGRAM
0A1D 09         REDIM:  DAD     B       ;COMPUTE LEN TO NEXT
0A1E 1B                 DCX     D       ;POINT LO NAME
0A1F 1B                 DCX     D       ;POINT HI DISP
0A20 1A                 LDAX    D       ;GET IT
0A21 47                 MOV     B,A     ;SAVE
0A22 1B                 DCX     D       ;POINT LO DISP
0A23 1A                 LDAX    D       ;GET IT
0A24 4F                 MOV     C,A     ;SAVE
0A25 09                 DAD     B       ;COMPUTE DIFF OF PRIOR DIM AND THIS
0A26 7C                 MOV     A,H     ;GET HI DIFF
0A27 B7                 ORA     A       ;TEST IT
0A28 FA330A             JM      REDM1   ;BRIF PREV > NEW
0A2B C20F1C             JNZ     SNERR   ;BRIF PREV < NEW
0A2E 7D                 MOV     A,L     ;GET LO DIFF
0A2F B7                 ORA     A       ;TEST IT
0A30 C20F1C             JNZ     SNERR   ;BRIF PREV < NEW
0A33 EB         REDM1:  XCHG            ;PUT ADDR IN H,L
0A34 2B                 DCX     H       ;POINT HI COL
0A35 D1                 POP     D       ;GET COL
0A36 72                 MOV     M,D     ;MOVE HI
0A37 2B                 DCX     H       ;POINT LO COL
0A38 73                 MOV     M,E     ;MOVE LO
0A39 D1                 POP     D       ;GET ROW
0A3A 2B                 DCX     H       ;POINT HI ROW
0A3B 72                 MOV     M,D     ;MOVE HI
0A3C 2B                 DCX     H       ;POINT LO ROW
0A3D 73                 MOV     M,E     ;MOVE LO
0A3E C30C0A             JMP     DIM3    ;CONTINUE
                        ;PAGE
```

```
                        ;
0A41 =                  SIN     EQU     $
                        ;
                        ; COMPUTE SINE OF X, (X IN RADIANS)
                        ;
                        ; USES 4TH DEGREE POLYNOMIAL APPROXIMATION
                        ;
                        ;
                        ; FIRST, REDUCE ANGLE TO RANGE: (-PI/2,PI/2)
                        ;
0A41 CDCE18                     CALL    FTEST   ;GET STATUS OF ANGLE
0A44 C8                         RZ              ;SIN(0)=0
0A45 F5                         PUSH    PSW     ;SAVE SIGN OF ANGLE
0A46 CDC708                     CALL    ABS
0A49 F1                 SIN1:   POP     PSW     ;COMPLEMENT SIGN FOR EACH PI SUB'D
0A4A 2F                         CMA             ;..
0A4B F5                         PUSH    PSW     ;..
0A4C 21A21D                     LXI     H,PI    ;REDUCE TO -PI<X<0
0A4F CD0C17                     CALL    FSUB
0A52 F2490A                     JP      SIN1
0A55 21D61D                     LXI     H,HALFP ;NOW ADD PI FOR -PI<X<-PI/2
0A58 E5                         PUSH    H
0A59 CD3716                     CALL    FADD
0A5C F47A0C                     CP      NEG     ;AND JUST NEGATE FOR -PI/2<X<0
0A5F E1                         POP     H
0A60 CD3716                     CALL    FADD
0A63 F1                         POP     PSW     ;RESTORE SIGN
  64 B7                         ORA     A
  65 F47A0C                     CP      NEG
                        ;
                        ; INIT REGISTERS
                        ;
0A68 212F22                     LXI     H,TEMP1 ;POINT T1
0A68 DF                         RST     3       ;SAVE IT
0A6C 3A5822                     LDA     FACC    ;GET SIGN&EXPONENT
0A6F CDDC18                     CALL    FEXP    ;EXPAND EXPON.
0A72 F2780A                     JP      SIN3A   ;BRIF POSITIVE
0A75 FEFD                       CPI     0FDH    ;TEST EXPONENT
0A77 D8                         RC              ;RETURN IF VERY SMALL RADIAN
                        ;
                        ; ABOVE ROUTINE WILL APPROX SIN(X) == X FOR X: (-.06,.06)
                        ;
0A78 21D61D             SIN3A:  LXI     H,HALFP ;POINT PI/2
0A78 CD9817                     CALL    FDIV    ;COMPUTE X/PI/2
0A7E 213322                     LXI     H,TEMP2 ;POINT T2
0A81 DF                         RST     3       ;STORE IT
0A82 213322                     LXI     H,TEMP2 ;POINT BACK
0A85 CD1817                     CALL    FMUL    ;COMPUTE SQUARE
0A88 21E61D                     LXI     H,SINCO ;POINT CONSTANTS

                        ; EVALUATE POWER SERIES

                        ; EVALUATE STARTING FROM HIGH ORDER COEFFICIENT:
                        ;  F(X)=(...(CN*FACC+C(N-1))*FACC+...+C1)*FACC*TEMP2+TEMP1

                        ;ON ENTRY:
                        ;       TEMP1=CONSTANT TERM
                        ;       TEMP2=X OR 1
```

```
                ;       FACC=X^2 OR X
                ;       (HL)=COEFFICIENT OF LAST TERM

0A8B E5         EVPS:   PUSH    H       ;SAVE POINTER TO COEFFICIENTS
0A8C 213722             LXI     H,TEMP3 ;SAVE FACC
0A8F DF                 RST     3
0A90 E1                 POP     H       ;RESTORE H
0A91 E5                 PUSH    H
0A92 C39C0A             JMP     EVPS2
0A95 E5         EVPS1:  PUSH    H       ;SAVE PTR TO NEXT COEFFICIENT
0A96 CD3716             CALL    FADD    ;FACC+CN->FACC
0A99 213722             LXI     H,TEMP3 ;POINTER TO X^N
0A9C CD1817     EVPS2:  CALL    FMUL    ;FACC*X^N->FACC
0A9F E1                 POP     H       ;COEFFICIENT PTR
0AA0 E7                 RST     4       ;MOVE TO NEXT COEFFICIENT
0AA1 FC                 DB      -4 AND 0FFH
0AA2 7E                 MOV     A,M     ;GET EXPONENT
0AA3 3D                 DCR     A       ;TEST FOR 1
0AA4 C2950A             JNZ     EVPS1   ;BRIF NOT 1
0AA7 213322             LXI     H,TEMP2 ;MUL BY TEMP2
0AAA CD1817             CALL    FMUL
0AAD 212F22             LXI     H,TEMP1 ;POINT TO CONSTANT TERM
0AB0 C33716             JMP     FADD    ;ADD IT AND RETURN TO CALLER
                ;
0AB3 =          COS     EQU     $
                ;
                ;
                ; COMPUTE COSINE OF ANGLE, X EXPRESSED IN RADIANS
                ; USES THE TRANSFORMATION: Y = PI/2 +- X
                ;      AND THEN COMPUTES SIN(Y).
                ;
                ;
0AB3 21D61D             LXI     H,HALFP ;COMPUTE PI/2 + X
0AB6 CD3716             CALL    FADD    ;GO ADD
0AB9 C3410A             JMP     SIN     ;GO COMPUTE SINE
                ;
0ABC =          TAN     EQU     $
                ;
                ; COMPUTE TANGENT OF X, IN RADIANS
                ; USES THE RELATION:
                ;
                ;           SIN(X)
                ; TAN(X) = ------
                ;           COS(X)
                ;
0ABC 213B22             LXI     H,TEMP4 ;POINT SAVE AREA
0ABF DF                 RST     3       ;SAVE ANGLE
0AC0 CDB30A             CALL    COS     ;COMPUTE COS(X)
0AC3 214722             LXI     H,TEMP7 ;SAVE COS(X)->TEMP7
0AC6 DF                 RST     3
0AC7 213B22             LXI     H,TEMP4 ;MOVE X->FACC
0ACA EF                 RST     5
0ACB CD410A             CALL    SIN     ;COMPUTE SINE
0ACE 214722             LXI     H,TEMP7 ;POINT COS
0AD1 C39B17             JMP     FDIV    ;DIVIDE AND RETURN TO CALLER
                ;
0AD4 =          ATN     EQU     $
                ;
```

```
                ; COMPUTES THE ARCTANGENT OF X
                ; USES A SEVENTH DEGREE POLYNOMIAL APPROXIMATION
                ;
0AD4 CDCE18             CALL    FTEST   ;CHECK SIGN OF ARGUMENT
0AD7 F2E30A             JP      ATN1    ;BRIF POSITIVE
0ADA CD7A0C             CALL    NEG     ;REVERSE SIGN
0ADD CDE30A             CALL    ATN1    ;GET POSITIVE ATN
0AE0 C37A0C             JMP     NEG     ;MAKE NEG & RETURN
                ;
0AE3 21EA1D     ATN1:   LXI     H,ONE   ;POINT: 1
0AE6 CD3716             CALL    FADD    ;GO ADD
0AE9 212F22             LXI     H,TEMP1 ;POINT SAVE
0AEC DF                 RST     3       ;STORE
0AED 219A1D             LXI     H,TWO   ;POINT: 2
0AF0 CD0C17             CALL    FSUB    ;GO SUBTRACT
0AF3 212F22             LXI     H,TEMP1 ;POINT SAVED
0AF6 CD9817             CALL    FDIV    ;DIVIDE
0AF9 213322             LXI     H,TEMP2 ;POINT SAVE
0AFC DF                 RST     3       ;SAVE X'=(X-1)/(X+1)
0AFD 21A61D             LXI     H,QTRPI ;X'+PI/4 -> TEMP1
0B00 CD3716             CALL    FADD
0B03 212F22             LXI     H,TEMP1
0B06 DF                 RST     3
0B07 E5                 PUSH    H       ;SAVE PTR TO TEMP2
0B08 EF                 RST     5       ;LOAD IT
0B09 E1                 POP     H
0B0A CD1817             CALL    FMUL    ;FACC=X'*X'
0B0D 21D21D             LXI     H,ATNCO ;POINT LIST COEFFICIENTS
0B10 C3880A             JMP     EVPS    ;GO COMPUTE & RETURN
                ;
0B13 =          LN      EQU     $
                ;
                ;
                ; COMPUTES THE NATURAL LOGRITHM, LN(X)
                ; USES A 7TH DEGREE POLYNOMIAL APPROXIMATION
                ;
0B13 CDCE18             CALL    FTEST   ;TEST THE ARGUMENT
0B16 FA071C             JM      ZMERR   ;LN(-X)=NO NO
0B19 CA071C             JZ      ZMERR   ;LN(0)=NO NO ALSO
0B1C 213322             LXI     H,TEMP2 ;POINT SAVE AREA
0B1F DF                 RST     3       ;STORE IT
0B20 3A5822             LDA     FACC    ;GET EXPON
0B23 CDDC18             CALL    FEXP    ;EXPAND TO 8 BITS
0B26 CA2C0B             JZ      LN0     ;BRIF 0.5 < X < 1.0
0B29 F2380B             JP      LN1     ;BRIF POSITIVE EXPONENT
0B2C 2F         LN0:    CMA             ;ELSE, COMPLIMENT
0B2D C602               ADI     2       ;PLUS TWO
0B2F CD1A0D             CALL    FDEC    ;CONVERT TO FLOAT POINT
0B32 CD7A0C             CALL    NEG     ;THEN NEGATE
0B35 C33D0B             JMP     LN2     ;GO AROUND
0B38 DE01       LN1:    SBI     1       ;MINUS ONE
0B3A CD1A0D             CALL    FDEC    ;CONVERT TO FLOATING POINT
0B3D 21AE1D     LN2:    LXI     H,LN2C  ;POINT LN(2)
0B40 CD1817             CALL    FMUL    ;MULTIPLY
0B43 212F22             LXI     H,TEMP1 ;POINT SAVE AREA
0B46 DF                 RST     3       ;STORE IT
0B47 EF                 RST     5       ;GET ORIG X
0B48 3E01               MVI     A,1     ;GET EXPONENT: 1
```

```
084A 325822             STA     FACC    ;ADJUST TO RANGE (1,2)
084D 21EA1D             LXI     H,ONE   ;POINT 1
0850 E5                 PUSH    H       ;SAVE PTR TO ONE
0851 CD0C17             CALL    FSUB    ;SUBTRACT ONE
0854 D1                 POP     D       ;SET TEMP2=1
0855 213322             LXI     H,TEMP2
0858 CD481C             CALL    CPY4D
085B 21061E             LXI     H,LNC0  ;POINT COEFFICIENTS
085E C3880A             JMP     EVPS    ;APPROXIMATE & RETURN
                ;
                ; X=LOG(X) --- THIS IS LOG BASE 10.
                ;
0861 =          LOG     EQU     $
0861 CD130B             CALL    LN      ;COMPUTE NATURAL LOG
0864 21221E             LXI     H,LNC   ;POINT LOG(E)
0867 C31817             JMP     FMUL    ;MULTIPLY AND RETURN
                ;
086A =          EXP     EQU     $
                ;
                ;  COMPUTES EXP(X) USING ALGORITHM EXP(X)=(2^I)*(2^FP) WHERE
                ;  2^I=INT(X*LN BASE 2 OF E) AND,
                ;  2^FP=5TH DEGREE POLY. APPROXIMATION
                ;  FP=FRACTIONAL PART OF INT(X*LN2E)
                ;
086A CDCE18             CALL    FTEST   ;CHECK SIGN
086D F2840B             JP      EXP1    ;BRIF POSITIVE
0870 CD7A0C             CALL    NEG     ;ELSE, REVERSE SIGN
0873 CD840B             CALL    EXP1    ;COMPUTE POSITIVE EXP
0876 212F22             LXI     H,TEMP1 ;POINT SAVE AREA
0879 DF                 RST     3       ;STORE IT
087A 21EA1D             LXI     H,ONE   ;POINT 1
087D EF                 RST     5       ;LOAD IT
087E 212F22             LXI     H,TEMP1 ;POINT PREV
0881 C39B17             JMP     FDIV    ;RECRIPRICAL AND RETURN
                ;
0884 210A1E     EXP1:   LXI     H,LN2E  ;POINT LN BASE 2 OF E
0887 CD1817             CALL    FMUL    ;FACC=X*(LN2E)
088A 213722             LXI     H,TEMP3 ;POINT SAVE AREA
088D DF                 RST     3       ;TEMP3=X*LN2E
088E CDE20B             CALL    INT     ;FACC=INT(X*LN2E)
0891 213B22             LXI     H,TEMP4 ;POINT SAVE AREA
0894 DF                 RST     3       ;TEMP4=INT(X*LN2E)
0895 DF                 RST     3       ;DITTO FOR TEMP5
0896 3A5822             LDA     FACC    ;GET THE EXPONENT COUNT
0899 47                 MOV     B,A     ;SAVE COUNT IN B
089A 3A5922             LDA     FACC+1  ;GET MANTISSA
089D 07         ELOOP:  RLC             ;ROTATE LEFT
089E 05                 DCR     B       ;REDUCE COUNT
089F C29D08             JNZ     ELOOP   ;CONTINUE SHIFTING
08A2 3C                 INR     A       ;ADJUST EXPONENT
08A3 323B22             STA     TEMP4   ;STORE EXPONENT
08A6 3E80               MVI     A,80H   ;LOAD CONSTANT
08A8 323C22             STA     TEMP4+1 ;STORE AS MANTISSA
08AB 21EA1D             LXI     H,ONE   ;1 -> TEMP1, TEMP2
08AE EF                 RST     5
08AF 212F22             LXI     H,TEMP1
08B2 DF                 RST     3
08B3 DF                 RST     3
```

```
0BB4 EF                 RST     5       ;LOAD TEMP3=INT(X*LN2E)
0BB5 213F22             LXI     H,TEMP5 ;GET FACC=FP(X*LN2E)
 BB8 CD0C17             CALL    FSUB
0BB8 211E1E             LXI     H,EXPCO ;POINT CONSTANTS
0BBE CD880A             CALL    EVPS    ;COMPUTE POLYNOMIAL
0BC1 213822             LXI     H,TEMP4 ;POINT 26(INT(X*LN2E))
0BC4 C31817             JMP     FMUL    ;MULTIPLY,NORMALIZE AND RETURN
                ;
                ;
0BC7 =          ABS     EQU     $
                ;
                ;
                ; RETURN THE ABSOLUTE VALUE OF THE FLOATING ACCUMULATOR
                ;
                ;
0BC7 3A5822             LDA     FACC    ;GET EXPONENT
0BCA E67F               ANI     7FH     ;STRIP NEGATIVE SIGN
0BCC 325822             STA     FACC    ;REPLACE
0BCF C9                 RET             ;RETURN
                ;
0BD0 =          SGN     EQU     $
                ;
                ;
                ; RETURNS THE SIGN OF THE FLOATING ACCUMULATOR
                ; THAT IS:
                ;  1 IF FACC > 0
                ;  0 IF FACC = 0
                ; -1 IF FACC < 0
                ;
0BD0 CDCE18             CALL    FTEST   ;GET STATUS OF FACC
0BD3 C8                 RZ              ;RETURN IF ZERO
0BD4 E680               ANI     80H     ;ISOLATE SIGN
0BD6 F601       SGN1:   ORI     1       ;CREATE EXPONENT
0BD8 F5                 PUSH    PSW     ;SAVE IT
0BD9 21EA1D             LXI     H,ONE   ;GET ADDRESS OF CONSTANT 1
0BDC EF                 RST     5       ;GO LOAD IT
0BDD F1                 POP     PSW     ;RESTORE SIGN
0BDE 325822             STA     FACC    ;SET THE SIGN
0BE1 C9                 RET             ;RETURN
                ;
0BE2 =          INT     EQU     $
                ;
                ;
                ; RETURNS THE GREATEST INTEGER NOT LARGER THAN VALUE IN FACC
                ; E.G.:
                ;     INT(3.14159) = 3
                ;     INT(0)       = 0
                ;     INT(-3.1415) =-4
                ;
                ;
0BE2 215822             LXI     H,FACC  ;POINT FLOAT ACC
0BE5 7E                 MOV     A,M     ;GET EXPONENT
0BE6 E640               ANI     40H     ;GET SIGN OF CHARACTERISTIC
 BE8 CAF00B             JZ      INT2    ;BRIF GE ZERO
0BEB 0604               MVI     B,4     ;LOOP CTR
0BED C35E1C             JMP     ZEROM   ;GO ZERO THE FACC
0BF0 7E         INT2:   MOV     A,M     ;GET EXPONENT AGAIN
0BF1 B7                 ORA     A       ;TEST SIGN
```

```
0BF2 F2FF0B             JP      INT3    ;BRIF POSITIVE OR ZERO
0BF5 21AA1D             LXI     H,NEGON ;POINT CONSTANT: -.9999999
0BF8 CD3716             CALL    FADD    ;ADD TO FACC
0BFB 215822             LXI     H,FACC  ;POINT EXPONTENT AGAIN
0BFE 7E                 MOV     A,M     ;LOAD IT
0BFF E63F       INT3:   ANI     3FH     ;ISOLATE CHARACTERISTIC
0C01 FE18               CPI     24      ;TEST IF ANY FRACTION
0C03 F0                 RP              ;RETURN IF NOT
0C04 47                 MOV     B,A     ;SAVE EXPONENT
0C05 3E18               MVI     A,24    ;GET CONSTANT
0C07 90                 SUB     B       ;MINUS EXPONENT = LOOP CTR
0C08 4F                 MOV     C,A     ;SAVE IT
0C09 215922     INT4:   LXI     H,FACC+1        ;POINT MSB
0C0C AF                 XRA     A       ;CLEAR CY FLAG
0C0D 0603               MVI     B,3     ;BYTE COUNT
0C0F 7E         INT5:   MOV     A,M     ;LOAD A BYTE
0C10 1F                 RAR             ;SHIFT RIGHT
0C11 77                 MOV     M,A     ;REPLACE
0C12 23                 INX     H       ;POINT NEXT
0C13 05                 DCR     B       ;DECR BYTE CTR
0C14 C20F0C             JNZ     INT5    ;LOOP
0C17 0D                 DCR     C       ;DECR BIT CTR
0C18 C2090C             JNZ     INT4    ;LOOP
0C1B 215822             LXI     H,FACC  ;POINT SIGN & EXP
0C1E 7E                 MOV     A,M     ;LOAD IT
0C1F E680               ANI     80H     ;ISOLATE SIGN
0C21 C618               ADI     24      ;PLUS INTEGER
0C23 77                 MOV     M,A     ;REPLACE IT
0C24 C3DD16             JMP     FNORM   ;GO NORMALIZE & RETURN
                ;
0C27 =          SQR     EQU     $
                ;
                ; COMPUTE SQUARE ROOT OF ARG IN FACC, PUT RESULT IN FACC
                ;
                ; USE HERON'S ITERATIVE PROCESS
                ;
0C27 CDCE18             CALL    FTEST   ;TEST THE ARGUMENT
0C2A C8                 RZ              ;RETURN IF ZERO
0C2B FA071C             JM      ZMERR   ;ERROR IF NEGATIVE
0C2E 327522             STA     DEXP    ;SAVE ORIG EXPONENT
0C31 AF                 XRA     A       ;GET A ZERO
0C32 325822             STA     FACC    ;PUT ARG IN RANGE [.5, 1]
0C35 213322             LXI     H,TEMP2 ;POINT SAVE AREA
0C38 DF                 RST     3       ;STORE IT
                ;
                ; INITIAL APROXIMATION: 0.41730759 + 0.59016206 * MANTISSA
                ;
0C39 21821D             LXI     H,SQC1  ;POINT .59016
0C3C CD1817             CALL    FMUL    ;GO MULTIPLY
0C3F 21861D             LXI     H,SQC2  ;PINT .4173
0C42 CD3716             CALL    FADD    ;GO ADD
0C45 212F22             LXI     H,TEMP1 ;POINT SAVE AREA
0C48 DF                 RST     3       ;GO STORE IT
                ;
                ; NEWTON'S METHOD OF ITERATION TO THE APPROXIMATE
                ; VALUE OF THE SQR OF MANTISSA
                ;
0C49 CD640C             CALL    SQR1    ;FIRST ITERATION
```

```
0C4C 212F22             LXI     H,TEMP1 ;POINT SAVE AREA
0C4F DF                 RST     3       ;STORE IT
0C50 CD640C             CALL    SQR1    ;SECOND ITERATION
                ;
                ; RESTORE RANGE TO OBTAIN THE FINAL RESULT
                ;
0C53 3A7522             LDA     DEXP    ;GET SAVED EXPONENT
0C56 CDDC18             CALL    FEXP    ;EXPAND IT
0C59 1F                 RAR             ;DIVIDE BY 2
0C5A 325822             STA     FACC    ;STORE IT
0C5D D0                 RNC             ;RETURN IF EXPON EVEN
0C5E 21BA1D             LXI     H,SQC3  ;ELSE, POINT SQR(2)
0C61 C31817             JMP     FMUL    ;GO MULTIPLY AND RETURN
                ;
                ; THIS ROUTINE PERFORMS ONE NEWTON ITERATION
                ; TO THE SQUARE ROOT FUNCTION
                ;
0C64 213322     SQR1:   LXI     H,TEMP2 ;POINT MANTISSA
0C67 EF                 RST     5       ;LOAD IT
0C68 212F22             LXI     H,TEMP1 ;POINT PREV GUESS
0C68 CD9B17             CALL    FDIV    ;FORM MANT/TEMP1
0C6E 212F22             LXI     H,TEMP1 ;POINT PREV
0C71 CD3716             CALL    FADD    ;FORM TEMP1 + MANT/TEMP1
0C74 D601               SUI     1       ;DIVIDE BY 2
0C76 325822             STA     FACC    ;FORM (TEMP1 + MANT/TEMP1)/2
0C79 C9                 RET             ;RETURN
                ;
0C7A =          NEG     EQU     $
                ;
                ;
                ; REVERSES THE SIGN OF THE FLOATING ACC
                ;
                ;
0C7A CDCE18             CALL    FTEST   ;GET STATUS OF FACC
0C7D C8                 RZ              ;RETURN IF ZERO
0C7E EE80               XRI     80H     ;REVERSE SIGN
0C80 325822             STA     FACC    ;RESTORE EXPONENT
0C83 C9                 RET             ;CONTINUE EVALUATION
                ;
0C84 =          RND     EQU     $
                ;
                ;
                ; PSEUDO RANDOM NUMBER GENERATOR
                ;
                ;
0C84 214722             LXI     H,TEMP7 ;SAVE ARG
0C87 DF                 RST     3
0C88 0604               MVI     B,4     ;LOOP CTR
0C8A 215822             LXI     H,FACC  ;POINT FLOAT ACCUM
0C8D CD5E1C             CALL    ZEROM   ;GO ZERO THE FACC
0C90 0E03               MVI     C,3     ;OUTTER LOOP CTR
0C92 215922             LXI     H,FACC+1        ;POINT MSB
0C95 E5                 PUSH    H       ;SAVE H,L
0C96 217C22     RND1:   LXI     H,RNDZ+1 -      ;POINT X,Y,Z
0C99 0506               MVI     B,5     ;LOOP CTR
0C9B B7                 ORA     A       ;TURN OFF CY
0C9C 7E         RND2:   MOV     A,M     ;GET A BYTE
0C9D 17                 RAL             ;SHIFT LEFT (MULT BY 2)
```

```
0C9E 77              MOV     M,A       ;REPLACE THE BYTE
0C9F 2B              DCX     H         ;POINT NEXT
0CA0 05              DCR     B         ;DECR CTR
0CA1 C29C0C          JNZ     RND2      ;LOOP
0CA4 23              INX     H         ;POINT MSD X,Y,Z
0CA5 11651D          LXI     D,RNDP    ;POINT TO MODULO
0CA8 0603            MVI     B,3       ;LOOP CTR
0CAA 1A      RND3:   LDAX    D         ;GET BYTE OF P,Q,R
0CAB BE              CMP     M         ;COMPARE WITH X,Y,Z
0CAC 13              INX     D         ;POINT NEXT
0CAD 23              INX     H         ;DITTO
0CAE DAB90C          JC      RND4      ;BRIF P<X
0CB1 C2C50C          JNZ     RND5      ;BRIF P>X
0CB4 1A              LDAX    D         ;GET LOW BYTE
0CB5 BE              CMP     M         ;CMPARE
0CB6 D2C50C          JNC     RND5      ;BRIF P>=X
0CB9 EB      RND4:   XCHG              ;FLIP D,E TO H,L
0CBA 1A              LDAX    D         ;GET LOW X BYTE
0CBB 96              SUB     M         ;SUBTRACT LOW P BYTE
0CBC 12              STAX    D         ;STORE IT
0CBD 1B              DCX     D         ;POINT HIGH
0CBE 2B              DCX     H         ;DITTO
0CBF 1A              LDAX    D         ;GET HIGH X BYTE
0CC0 9E              SBB     M         ;SUB HIGH P BYTE
0CC1 12              STAX    D         ;STORE IT
0CC2 13              INX     D         ;POINT LOW
0CC3 23              INX     H         ;DITTO
0CC4 EB              XCHG              ;RESTORE ADDRS
0CC5 13      RND5:   INX     D         ;POINT NEXT
0CC6 23              INX     H         ;DITTO
0CC7 05              DCR     B         ;DECR CTR
0CC8 C2AA0C          JNZ     RND3      ;LOOP
0CCB 0603            MVI     B,3       ;LOOP CTR
0CCD 117E22  RND6:   LXI     D,RNDS+1          ;POINT LOW S
0CD0 1A              LDAX    D         ;GET LOW S
0CD1 86              ADD     M         ;ADD LOW X,Y,Z
0CD2 12              STAX    D         ;PUT S
0CD3 1B              DCX     D         ;POINT HIGH
0CD4 2B              DCX     H         ;DITTO
0CD5 1A              LDAX    D         ;GET HIGH S
0CD6 8E              ADC     M         ;ADD HIGH X,Y,Z
0CD7 E63F            ANI     3FH       ;TURN OFF HIGH BITS
0CD9 12              STAX    D         ;STORE IT
0CDA 2B              DCX     H         ;POINT NEXT X,Y,Z
0CDB 05              DCR     B         ;DECR CTR
0CDC C2CD0C          JNZ     RND6      ;LOOP
0CDF 3E08            MVI     A,8       ;CONSTANT
0CE1 91              SUB     C         ;LESS CTR
0CE2 1F              RAR               ;DIVIDE BY TWO
0CE3 E1              POP     H         ;GET H,L ADDR
0CE4 3A7E22          LDA     RNDS+1    ;GET LSB OF S
0CE7 77              MOV     M,A       ;STORE IT
0CE8 23              INX     H         ;POINT NEXT
0CE9 E5              PUSH    H         ;SAVE H,L
0CEA 0D              DCR     C         ;DECR CTR
0CEB C2960C          JNZ     RND1      ;LOOP
0CEE E1              POP     H         ;RESTORE SP PTR
0CEF 3A8722          LDA     RNDSW     ;GET SWITCH
```

```
0CF2 B7                 ORA     A       ;TEST IT
0CF3 CA010D             JZ      RND7    ;BRIF NO RANDOMIZE
0CF6 117F22             LXI     D,TRNDX ;POINT SAVED VALUES
0CF9 217722             LXI     H,RNDX  ;POINT NEXT VALUES
0CFC 0608               MVI     B,8     ;LOOP CTR
0CFE CD581C             CALL    COPYH   ;GO COPY
0D01 CDDD16     RND7:   CALL    FNORM
0D04 214722             LXI     H,TEMP7 ;MULTIPLY BY RANGE
0D07 C31817             JMP     FMUL
                ;
0D0A =          INP     EQU     $
                ;
                ;
                ; INPUT A BYTE FROM THE DEVICE IN FACC
                ;
                ; PUT THE RESULT IN THE FACC
                ;
0D0A CD661C             CALL    FBIN    ;CONVERT FACC TO BINARY
0D0D 212022             LXI     H,OUTA  ;POINT INSTR BUFFER
0D10 36D8               MVI     M,0D8H  ;IN INSTR
0D12 23                 INX     H       ;POINT NEXT
0D13 77                 MOV     M,A     ;MOVE ADDR
0D14 23                 INX     H       ;POINT NEXT
0D15 36C9               MVI     M,0C9H  ;RET INSTR
0D17 CD2022             CALL    OUTA    ;GO INPUT A BYTE
0D1A 5F         FDEC:   MOV     E,A     ;MOVE BYTE TO LO D,E
0D1B 1600               MVI     D,0     ;ZERO HI D,E
0D1D C3891C             JMP     BINFL   ;GO CONVERT TO DEC & RET
                ;
0D20 =          POS     EQU     $
                ;
                ;
                ; RETURNS THE CURRENT POSITION OF THE TTY CURSOR
                ;
                ;
0D20 3A7622             LDA     COLUM   ;GET POSITION
0D23 C31A0D             JMP     FDEC    ;CONVERT TO FLOAT AND RETURN
                ;
0D26 =          CONCA   EQU     $
                ;
                ;
                ; CONCATONATE TWO STRINGS TOGETHER
                ; COMBINED LENGTH <= 255
                ;
0D26 D1                 POP     D       ;ADJUST STACK
0D27 112021             LXI     D,STRIN ;POINT STRING BUFFER
0D2A 1A                 LDAX    D       ;GET CURRENT LENGTH
0D2B 4F                 MOV     C,A     ;STORE IT
0D2C 0600               MVI     B,0     ;CLEAR HI
0D2E EB                 XCHG            ;FLIP FLOP
0D2F 09                 DAD     B       ;COMPUTE NEXT
0D30 EB                 XCHG            ;FLIP BACK
0D31 86                 ADD     M       ;COMPUTE COMBINED LENGTH
0D32 46                 MOV     B,M     ;SAVE LEN2
0D33 D23C0D             JNC     CONC2   ;BRIF NO OVFLW
0D36 3EFF               MVI     A,255   ;MAX LEN
0D38 91                 SUB     C       ;MINUS 1ST PART
0D39 47                 MOV     B,A     ;SAVE LEN
```

```
0D3A 3EFF               MVI     A,255   ;UPDATED LENGTH
0D3C 322021     CONC2:  STA     STRIN   ;STORE IT
0D3F 78                 MOV     A,B     ;GET LEN TO MOVE
0D40 B7                 ORA     A       ;TEST IT
0D41 CA4C0D             JZ      CONC4   ;BRIF NULL
0D44 23         CONC3:  INX     H       ;POINT NEXT
0D45 13                 INX     D       ;DITTO
0D46 7E                 MOV     A,M     ;GET NEXT CHAR
0D47 12                 STAX    D       ;PUT IT
0D48 05                 DCR     B       ;DECR COUNT
0D49 C2440D             JNZ     CONC3   ;LOOP
0D4C E1         CONC4:  POP     H       ;GET H,L
0D4D 2B                 DCX     H       ;POINT BACK
0D4E 3A2021             LDA     STRIN   ;GET LEN
0D51 1F                 RAR             ;DIVIDE BY TWO
0D52 3C                 INR     A       ;PLUS ONE
0D53 EB                 XCHG            ;SAVE H,L
0D54 2A6922             LHLD    SPCTR   ;GET CTR
0D57 4F                 MOV     C,A     ;SAVE CTR
0D58 0600               MVI     B,0     ;ZERO HI BYTE
0D5A 09                 DAD     B       ;ADD LEN THIS STRNG
0D5B 226922             SHLD    SPCTR   ;SAVE CTR
0D5E C1                 POP     B
0D5F 210000             LXI     H,0     ;GET ADDR ZERO
0D62 E5         CONC5:  PUSH    H       ;2 BYTE WORD
0D63 3D                 DCR     A       ;DECR CTR
0D64 C2620D             JNZ     CONC5   ;CONTINUE
0D67 39                 DAD     SP      ;GET ADDRESS IN H,L
0D68 EB                 XCHG            ;PUT STACK PTR IN D,E
0D69 72                 MOV     M,D     ;MOVE HI ADDR
0D6A 23                 INX     H       ;POINT NEXT
0D6B 73                 MOV     M,E     ;MOVE LO ADDR
0D6C 23                 INX     H       ;POINT NEXT
0D6D 36E7               MVI     M,0E7H  ;TYPE=STRING
0D6F E5                 PUSH    H       ;SAVE H,L
0D70 212021             LXI     H,STRIN ;GET TEMP STR
0D73 7E                 MOV     A,M     ;GET LENGTH
0D74 3C                 INR     A       ;PLUS ONE
0D75 4F                 MOV     C,A     ;SAVE IT
0D76 7E         CONC6:  MOV     A,M     ;GET A BYTE
0D77 12                 STAX    D       ;PUT IT DOWN
0D78 13                 INX     D       ;POINT NEXT
0D79 23                 INX     H       ;DITTO
0D7A 0D                 DCR     C       ;SUBT CTR
0D7B C2760D             JNZ     CONC6   ;LOOP
0D7E E1                 POP     H       ;RESTORE H,L
0D7F E7                 RST     4       ;ADJUST H,L
0D80 F9                 DB      -7 AND 0FFH
0D81 3E04               MVI     A,4     ;DELETE 4 BYTES
0D83 CDE21A             CALL    SQUIS   ;GO COMPRESS
0D86 C3BA11             JMP     EVAL    ;CONTINUE EVALUATION
                ;
0D89 =          LENFN   EQU     $
                ;
                ; X=LEN(A$)
                ;
                ; RETURN THE LENGTH OF THE STRING
                ;
```

```
0D89 3A2021             LDA     STRIN   ;GET LEN IN ACC
0D8C C31A0D             JMP     FDEC    ;GO CONVERT TO DECIMAL & RETURN
                ;
0D8F =          CHRFN   EQU     $
                ;
                ; A$=CHR$(X)
                ;
                ; RETURNS A ONE CHAR STRING HAVING THE ASCII VALUE - X
                ;
0D8F CD661C             CALL    FBIN    ;CONVERT FACC TO BINARY
0D92 212021             LXI     H,STRIN ;POINT OUT AREA
0D95 3601               MVI     M,1     ;LEN=1
0D97 23                 INX     H       ;POINT NEXT
0D98 77                 MOV     M,A     ;STORE THE CHAR
0D99 C9                 RET             ;RETURN
                ;
0D9A =          ASCII   EQU     $
                ;
                ; X=ASCII(A$)
                ;
                ; RETURNS THE ASCII VALUE OF THE FIRST CHAR IN THE STRING
                ;
0D9A 212021             LXI     H,STRIN ;POINT STRING
0D9D 7E                 MOV     A,M     ;GET LENGTH
0D9E 87                 ORA     A       ;TEST IF > ZERO
0D9F CA1A0D             JZ      FDEC    ;BRIF ZERO & RETURN A ZERO
0DA2 23                 INX     H       ;POINT 1ST CHAR
 DA3 7E                 MOV     A,M     ;LOAD IT
0DA4 C31A0D             JMP     FDEC    ;GO CONVERT TO DECIMAL & RETURN
                ;
0DA7 =          NUMFN   EQU     $
                ;
                ; A$=NUM$(X)
                ;
                ; RETURNS A STRING REPRESENTING X AS IT WOULD HAVE
                ; BEEN PRINTED (INCLUDING TRAILING SPACE)
                ;
0DA7 212021             LXI     H,STRIN ;POINT STRING AREA
0DAA 3600               MVI     M,0     ;INIT COUNT
0DAC 23                 INX     H       ;SKIP TO 1ST POSITION
0DAD CDF014             CALL    FOUT    ;GO CONVERT TO EXTRN DEC
0DB0 AF                 XRA     A       ;GET A ZERO
0DB1 47                 MOV     B,A     ;INIT CTR
0DB2 28         NUM1:   DCX     H       ;POINT PRIOR
0DB3 04                 INR     B       ;COUNT IT
0DB4 8E                 CMP     M       ;TEST IF ZERO
0DB5 C2B20D             JNZ     NUM1    ;LOOP TILL AT START
0DB8 70                 MOV     M,B     ;SET LEN CODE
0DB9 C9                 RET             ;THEN RETURN
                ;
0DBA =          VAL     EQU     $
                ;
                ; X=VAL(A$)
                ;
                ; RETURNS THE VALUE OF THE STRING OF NUMERIC CHARACTERS
                ;
0DBA 212021             LXI     H,STRIN ;POINT STRING AREA
0DBD 7E                 MOV     A,M     ;GET LEN
```

```
0D8E B7                 ORA     A       ;TEST FOR NULL STRING
0D8F 47                 MOV     B,A     ;SAVE LEN
0DC0 CA1A0D             JZ      FDEC    ;BRIF IS (RETURNS A 0.00)
0DC3 112021             LXI     D,STRIN ;POINT BUFFER
0DC6 23         VAL1:   INX     H       ;POINT NEXT
0DC7 7E                 MOV     A,M     ;GET A CHAR
0DC8 FE20               CPI     ' '     ;TEST IF SPACE
0DCA CACF0D             JZ      VAL2    ;BRIF IS
0DCD 12                 STAX    D       ;PUT THE CHAR
0DCE 13                 INX     D       ;INCR ADDR
0DCF 05         VAL2:   DCR     B       ;DECR CTR
0DD0 C2C60D             JNZ     VAL1    ;LOOP
0DD3 AF                 XRA     A       ;GET A ZERO
0DD4 12                 STAX    D       ;PUT IN BUFF
0DD5 212021             LXI     H,STRIN ;POINT START OF BUFFER
0DD8 CD2E14             CALL    FIN     ;GO CONVERT
0DDB 7E                 MOV     A,M     ;GET NON-NUMERIC
0DDC B7                 ORA     A       ;TEST IT
0DDD C21F1C             JNZ     CVERR   ;BRIF ERROR
0DE0 C9                 RET             ;ELSE, RETURN
                ;
0DE1 =          SPACE   EQU     $
                ;
                ; A$=SPACE$(X)
                ;
                ; CREATES A STRING OF SPACES LENGTH = X
                ;
0DE1 CD661C             CALL    FBIN    ;GET BINARY LENGTH
0DE4 212021             LXI     H,STRIN ;POINT TEMP STRING
0DE7 77                 MOV     M,A     ;PUT LEN
0DE8 B7                 ORA     A       ;TEST IT
0DE9 C8         SPAC1:  RZ              ;RETURN IF ZERO
0DEA 23                 INX     H       ;ELSE, POINT NEXT
0DEB 3620               MVI     M,' '   ;MOVE 1 SPACE
0DED 3D                 DCR     A       ;DECR CTR
0DEE C3E90D             JMP     SPAC1   ;LOOP
                ;
0DF1 =          STRFN   EQU     $
                ;
                ; A$=STRING$(X,Y)
                ;
                ; CREATES STRING OF LNGTH X CONTAINING REPETITION OF CHR$(Y)
                ;
0DF1 CD661C             CALL    FBIN    ;GET BINARY LENGTH
0DF4 322021             STA     STRIN   ;PUT TO STRING
0DF7 CD831C             CALL    ARGNU   ;GET NEXT ARGUMENT
0DFA 212021             LXI     H,STRIN ;POINT STRING
0DFD 46                 MOV     B,M     ;GET COUNT
0DFE 23         STRI1:  INX     H       ;POINT NEXT
0DFF 77                 MOV     M,A     ;STORE THE CHAR
0E00 05                 DCR     B       ;DECR CTR
0E01 C2FE0D             JNZ     STRI1   ;LOOP
0E04 C9                 RET             ;RETURN
                ;
0E05 =          LEFT    EQU     $
                ;
                ; B$=LEFT$(A$,X)
                ;
```

```
                ; SUBSTRING FROM THE LEFTMOST X CHARACTERS OF A$
                ;
0E05 CD831C             CALL    ARGNU   ;GET 2ND ARGUMENT
0E08 4F                 MOV     C,A     ;SAVE LEN
0E09 0601               MVI     B,1     ;INIT START
0E0B C3210E             JMP     MID0    ;CONTINUE
                ;
0E0E =          RIGHT   EQU     $
                ;
                ; B$=RIGHT$(A$,X)
                ;
                ; SUBSTRING STARTING AT POSITION X TO END OF STRING
                ;
0E0E CD831C             CALL    ARGNU   ;GET 2ND ARGUMENT
0E11 47                 MOV     B,A     ;SAVE START
0E12 0EFF               MVI     C,255   ;MAX LEN
0E14 C3210E             JMP     MID0    ;CONTINUE
                ;
0E17 =          MIDFN   EQU     $
                ;
                ; B$=MID$(A$,X,Y)
                ;
                ; SUBSTRING OF THE STRING A$ STARTING WITH CHARACTER # X
                ; AND Y CHARACTERS LONG
                ;
0E17 CD831C             CALL    ARGNU   ;LOAD X
0E1A 47                 MOV     B,A     ;SAVE START
0E1B C5                 PUSH    B       ;PUT ON STACK
0E1C CD831C             CALL    ARGNU   ;GET 3RD ARG
0E1F C1                 POP     B       ;RETREIVE
0E20 4F                 MOV     C,A     ;SAVE LEN
0E21 78         MID0:   MOV     A,B     ;LOAD START
0E22 212021             LXI     H,STRIN ;POINT STRING
0E25 BE                 CMP     M       ;TEST IF X>L
0E26 DA2F0E             JC      MID1    ;BRIF X>L
0E29 CA2F0E             JZ      MID1    ;OR EQUAL
0E2C 3600               MVI     M,0     ;ELSE, RESULT IS NULL
0E2E C9                 RET             ;RETURN
0E2F 81         MID1:   ADD     C       ;COMPUTE END POSITION
0E30 DA3C0E             JC      MID2    ;BRIF OVERFLOW
0E33 DE01               SBI     1       ;COMPUTE X+Y-1
0E35 DA3C0E             JC      MID2    ;BRIF OVERFLOW
0E38 BE                 CMP     M       ;COMPARE TO EXISTING LEN
0E39 DA400E             JC      MID3    ;BRIF X+Y-1<LEN(A$)
0E3C 7E         MID2:   MOV     A,M     ;ELSE GET ORIG LEN
0E3D 90                 SUB     B       ;MINUS X
0E3E 3C                 INR     A       ;PLUS ONE
0E3F 4F                 MOV     C,A     ;SAVE (REPLACE Y)
0E40 71         MID3:   MOV     M,C     ;PUT NEW LEN
0E41 58                 MOV     E,B     ;PUT START IN LO
0E42 1600               MVI     D,0     ;ZERO IN HI
0E44 19                 DAD     D       ;COMPUTE START
0E45 112021             LXI     D,STRIN ;GET BEGIN
0E48 7E         MID4:   MOV     A,M     ;GET A CHAR
0E49 13                 INX     D       ;POINT NEXT
0E4A 23                 INX     H       ;DITTO
0E4B 12                 STAX    D       ;PUT DOWN
0E4C 0D                 DCR     C       ;DECR CTR
```

```
0E4D C2480E             JNZ     MID4    ;LOOP
0E50 C9                 RET             ;THEN RETURN
                ;
0E51 =          INSTR   EQU     $
                ;
                ; X=INSTR(Y,A$,B$)
                ;
                ; SEARCH FOR SUBSTRING B$ IN STRING A$ STARTING AT POS Y.
                ; RETURN 0 IF B$ IS NOT IN A$
                ; RETURN 1 IF B$ IS NULL
                ; ELSE RETURN THE CHARACTER POSITION
                ;
0E51 CD831C             CALL    ARGNU   ;GET A$
0E54 212021             LXI     H,STRIN ;POINT A$
0E57 B7                 ORA     A       ;TEST Y
0E58 C2600E             JNZ     INST2   ;BRIF Y NOT ZERO
0E5B 3600       INST1:  MVI     M,0     ;ELSE A$ IS NULL
0E5D C3670E             JMP     INST3   ;GO AROUND
0E60 BE         INST2:  CMP     M       ;TEST Y TO LEN(A$)
0E61 CA670E             JZ      INST3   ;BRIF EQUAL
0E64 D2580E             JNC     INST1   ;BRIF Y > LEN(A$)
0E67 4F         INST3:  MOV     C,A     ;SAVE Y
0E68 0600               MVI     B,0     ;ZERO HI INCR
0E6A 7E                 MOV     A,M     ;GET LEN(A$)
0E6B 91                 SUB     C       ;MINUS Y
0E6C 3C                 INR     A       ;PLUS ONE
0E6D 09                 DAD     B       ;COMPUTE START ADDR
0E6E 47                 MOV     B,A     ;# CHARS REMAIN IN A$
0E6F E5                 PUSH    H       ;SAVE ADDR
0E70 2A5222             LHLD    ADDR1   ;GET ADDR OF ARG
0E73 23                 INX     H       ;POINT NEXT
0E74 56                 MOV     D,M     ;GET HI ADDR
0E75 23                 INX     H       ;POINT NEXT
0E76 5E                 MOV     E,M     ;GET LO ADDR
0E77 23                 INX     H       ;POINT NEXT
0E78 225222             SHLD    ADDR1   ;UPDATED PTR
0E7B E1                 POP     H       ;RESTORE ADDR
0E7C 1A                 LDAX    D       ;GET LEN(B$)
0E7D B7                 ORA     A       ;TEST IF NULL
0E7E C2870E             JNZ     INST6   ;BRIF NOT
0E81 0E01               MVI     C,1     ;SET POSIT = 1
0E83 79         INST5:  MOV     A,C     ;GET POSIT
0E84 C31A0D             JMP     FDEC    ;CONVERT TO DECIMAL & RETURN
0E87 EB         INST6:  XCHG            ;FLIP/FLOP
0E88 78                 MOV     A,B     ;GET LEN OF A$
0E89 BE                 CMP     M       ;COMPARE TO LEN B$
0E8A DAAC0E             JC      INSTA   ;BRIF LEN(B$)< LEN(REM A$)
0E8D C5                 PUSH    B       ;SAVE CTR, POSIT
0E8E D5                 PUSH    D       ;SAVE ADDR A$
0E8F E5                 PUSH    H       ;SAVE ADDR B$
0E90 4E                 MOV     C,M     ;GET LEN B$
0E91 EB                 XCHG            ;FLIP/FLOP
0E92 13         INST8:  INX     D       ;POINT NEXT B$
0E93 1A                 LDAX    D       ;GET B$ CHAR
0E94 BE                 CMP     M       ;COMPARE A$ CHAR
0E95 C2A30E             JNZ     INST9   ;BRIF NOT EQUAL
0E98 23                 INX     H       ;POINT NEXT A$
0E99 0D                 DCR     C       ;DECR CTR (LEN(B$))
```

```
0E9A C2920E             JNZ     INST8   ;LOOP
0E9D E1                 POP     H       ;DUMMY POP
0E9E E1                 POP     H       ;GET DUMMY STACK
0E9F C1                 POP     B       ;GET POSITION
0EA0 C3830E             JMP     INST5   ;WE FOUND A MATCH
0EA3 D1         INST9:  POP     D       ;GET PTR B$
0EA4 E1                 POP     H       ;GET PTR A$
0EA5 C1                 POP     B       ;GET CTRS, POSIT
0EA6 0C                 INR     C       ;UP PTR NUM
0EA7 23                 INX     H       ;POINT NEXT A$
0EA8 05                 DCR     B       ;DECR B
0EA9 C2870E             JNZ     INST6   ;LOOP
0EAC 0E00       INSTA:  MVI     C,0     ;ELSE B$ NOT IN A$
0EAE C3830E             JMP     INST5   ;RETURN
                ;
0EB1 =          FN      EQU     $
                ;
                ; STMT: DEF FNX(A)=EXPR
                ;
                ; NOTE: ENTRY FROM EXPR ANALYZER (RECURSIVE)
                ;
0EB1 C5                 PUSH    B       ;SAVE B,C
0EB2 D5                 PUSH    D       ;SAVE D,E
0EB3 E5                 PUSH    H       ;SAVE H,L
0EB4 EB                 XCHG            ;PUT H,L TO D,E
0EB5 2A5622             LHLD    ADDR3   ;GET ADDR
0EB8 E5                 PUSH    H       ;SAVE IT
0EB9 EB                 XCHG            ;PUT D,E BACK TO H,L
0EBA 225622             SHLD    ADDR3   ;UPDATE PTR
0EBD 2A6922             LHLD    SPCTR   ;GET SP COUNT
0EC0 E5                 PUSH    H       ;SAVE IT
0EC1 3A6322             LDA     PARCT   ;GET PAREN COUNT
0EC4 47                 MOV     B,A     ;PUT TO B
0EC5 3A8822             LDA     FNMOD   ;GET FN MODE
0EC8 4F                 MOV     C,A     ;PUT TO C
0EC9 C5                 PUSH    B       ;SAVE B,C
0ECA 3A7220             LDA     DIMSW   ;GET DIM SW
0ECD F5                 PUSH    PSW     ;SAVE IT
0ECE AF                 XRA     A       ;CLEAR A
0ECF 327220             STA     DIMSW   ;RESET DIM SW
0ED2 2A6C22             LHLD    FNARG   ;GET OLD ARG NAME
0ED5 E5                 PUSH    H       ;SAVE
0ED6 2A6E22             LHLD    FNARG+2 ;GET OLD ARG ADDRESS
0ED9 E5                 PUSH    H       ;SAVE
0EDA 2A9322             LHLD    PROGE   ;GET END OF PROGRAM
0EDD E5                 PUSH    H       ;SAVE IT
0EDE 2A5022             LHLD    EXPRS   ;GET END OF EXPR
0EE1 E5                 PUSH    H       ;SAVE IT
0EE2 229322             SHLD    PROGE   ;SAVE NEW 'END' OF PROGRAM
0EE5 3E01               MVI     A,1     ;GET ON SETTING
0EE7 328822             STA     FNMOD   ;SET IN FUNCTION
0EEA 2A5622             LHLD    ADDR3   ;POINT TO EXPR
0EED 4E                 MOV     C,M     ;GET FN CHAR
0EEE 2B                 DCX     H       ;POINT BACK
0EEF 46                 MOV     B,M     ;GET HI NAME
0EF0 219622             LXI     H,BEGPR ;POINT START OF PROGRAM
0EF3 7E         FN2:    MOV     A,M     ;LOAD LEN TO NEXT STMT
0EF4 B7                 ORA     A       ;TEST IF AT END
```

```
0EF5 CA0F1C          JZ      SNERR    ;BRIF FN NOT FOUND
0EF8 E5              PUSH    H        ;SAVE PTR
0EF9 E7              RST     4        ;ADJUST H,L
0EFA 03              DB      3
0EFB 111E1F          LXI     D,DEFLI  ;LITERAL
0EFE D7              RST     2        ;GO COMPARE
0EFF C2110F          JNZ     FN3      ;BRIF NOT EQUAL
0F02 C5              PUSH    B        ;SAVE TEST NAME
0F03 CDC91B          CALL    VAR      ;GO GET NAME
0F06 C1              POP     B        ;RESTORE NAME
0F07 7A              MOV     A,D      ;GET HI NAME
0F08 B8              CMP     B        ;COMPARE
0F09 C2110F          JNZ     FN3      ;BRIF NOT EQUAL
0F0C 78              MOV     A,E      ;GET LO
0F0D B9              CMP     C        ;COMPARE
0F0E CA190F          JZ      FN4      ;BRIF EQUAL
0F11 E1      FN3:    POP     H        ;GET OLD PTR
0F12 5E              MOV     E,M      ;GET LO LEN
0F13 1600            MVI     D,0      ;ZERO HI LEN
0F15 19              DAD     D        ;POINT NEXT STMT
0F16 C3F30E          JMP     FN2      ;LOOP
0F19 D1      FN4:    POP     D        ;ADJUST STACK
0F1A CF              RST     1        ;SKIP BLANKS
0F1B FE28            CPI     '('      ;TEST IF OPEN PAREN
0F1D C20F1C          JNZ     SNERR    ;BRIF NOT
0F20 23              INX     H        ;SKIP IT
0F21 CDC91B          CALL    VAR      ;GO GET VAR NAME
0F24 E5              PUSH    H        ;SAVE HL ADDR
0F25 216C22          LXI     H,FNARG  ;POINT DUMMY ARG TBL
0F28 72              MOV     M,D      ;STORE LETTER
0F29 23              INX     H        ;POINT NEXT
0F2A 73              MOV     M,E      ;STORE DIGIT
0F2B 23              INX     H        ;POINT NEXT
0F2C EB              XCHG             ;PUT H,L TO D,E
0F2D 2A5622          LHLD    ADDR3    ;POINT TO EXPR STACK
0F30 23              INX     H        ;POINT CODE
0F31 23              INX     H        ;POINT HI ADR
0F32 7E              MOV     A,M      ;GET HI
0F33 12              STAX    D        ;PUT TO TABLE
0F34 13              INX     D        ;POINT NEXT
0F35 23              INX     H        ;DITTO
0F36 7E              MOV     A,M      ;GET LO ADDR
0F37 12              STAX    D        ;PUT TO TABLE
0F38 E1              POP     H        ;RESTORE PTR TO STMT
0F39 CF              RST     1        ;SKIP BLANKS
0F3A FE29            CPI     ')'      ;TEST IF CLOSE PAREN
0F3C C20F1C          JNZ     SNERR    ;BRIF NOT
0F3F 23              INX     H        ;SKIP IT
0F40 CF              RST     1        ;SKIP BLANKS
0F41 FE3D            CPI     '='      ;TEST IF EQUAL SIGN
0F43 C20F1C          JNZ     SNERR    ;BRIF NOT
0F46 23              INX     H        ;SKIP IT
0F47 CD800F          CALL    EXPR     ;GO EVAL FUNCTION
0F4A CD941A          CALL    EOL      ;MUST BE END OF LINE
0F4D E1              POP     H        ;GET H,L
0F4E 225022          SHLD    EXPRS    ;RESTORE START OF EXPR
0F51 E1              POP     H        ;GET H,L
0F52 229322          SHLD    PROGE    ;RESTORE 'END' OF PROGRAM
```

```
0F55 E1          POP     H          ;GET H,L
0F56 226E22      SHLD    FNARG+2    ;STORE ADDR
0F59 E1          POP     H          ;GET H,L
0F5A 226C22      SHLD    FNARG      ;STORE DUMMY ARG
0F5D F1          POP     PSW        ;GET A,STATUS
0F5E 327220      STA     DIMSW      ;RESTORE DIM SW
0F61 C1          POP     B          ;GET B,C
0F62 79          MOV     A,C        ;LOAD C
0F63 328822      STA     FNMOD      ;RESTORE MOE
0F66 78          MOV     A,B        ;LOAD B
0F67 326822      STA     PARCT      ;RESTORE PAREN COUNT
0F6A E1          POP     H          ;GET H,L
0F6B 226922      SHLD    SPCTR      ;RESTORE SP COUNTER
0F6E E1          POP     H          ;GET H,L
0F6F 225622      SHLD    ADDR3      ;RESTORE ADDR OF EVAL
0F72 E1          POP     H          ;GET H,L
0F73 D1          POP     D          ;GET D,E
0F74 2B          DCX     H          ;POINT 2ND BYTE FOLLOWING OP
0F75 225422      SHLD    ADDR2      ;SAVE IT
0F78 E7          RST     4          ;POINT TO ARG TYPE
0F79 05          DB      5
0F7A 225222      SHLD    ADDR1      ;SAVE ADDR
0F7D C30712      JMP     EV3        ;GO WRAPUP
                 ;PAGE
```

```
                        ;
0F80 =                  EXPR    EQU     $
                        ;
                        ;
                        ; EVALUATE EXPRESSION ROUTINE
                        ; LEAVE RESULT IN FACC
                        ; RETURN WHEN EXPRESSION ENDS (TYPICALLY AT END OF LINE)
                        ;
                        ;
0F80 AF                         XRA     A       ;CLEAR REG A
0F81 326822                     STA     PARCT   ;SET PAREN CTR
0F84 EB                         XCHG            ;SAVE H,L
0F85 210000                     LXI     H,0     ;GET A ZERO
0F88 226922                     SHLD    SPCTR   ;INIT CTR
0F8B 2A9322                     LHLD    PROGE   ;POINT END OF PROGRAM AREA
0F8E 23                         INX     H       ;POINT ONE MORE
0F8F 3600                       MVI     M,0     ;INIT START OF STACK
0F91 225022                     SHLD    EXPRS   ;SAVE IT
0F94 EB                         XCHG            ;RESTORE H,L
                        ;
0F95 =                  LOOKD   EQU     $       ;LOOK FOR CON, VAR, OR FUNCTION
0F95 CF                         RST     1       ;SKIP TO NON-BLANK
0F96 CD2A1B                     CALL    NUMER   ;GO TEST IF NUMERIC
0F99 C2AF0F                     JNZ     LDALP   ;BRIF NOT
0F9C CD2E14             LDNUM:  CALL    FIN     ;GO CONVERT NUMERIC (PUT TO FACC)
0F9F 44                 LDF:    MOV     B,H     ;COPY H,L TO B,C
0FA0 4D                         MOV     C,L     ;SAME
0FA1 2A5022                     LHLD    EXPRS   ;GET ADDR OF EXPR AREA
0FA4 CD001B                     CALL    GTEMP   ;GO STORE THE FACC IN TEMP AREA
0FA7 225022                     SHLD    EXPRS   ;SAVE UPDATED ADDRESS
0FAA 60                         MOV     H,B     ;RESTORE H
0FAB 69                         MOV     L,C     ;RESTORE L
0FAC C31D11                     JMP     LOOKO   ;GO GET AN OPERATION CODE
0FAF FE2E               LDALP:  CPI     '.'     ;SEE IF LEADING DECIMAL POINT
0FB1 CA9C0F                     JZ      LDNUM   ;BRIF IS
0FB4 CD211B                     CALL    ALPHA   ;GO SEE IF ALPHA
0FB7 C29110                     JNZ     LDDTN   ;BRIF NOT
0FBA 46                         MOV     B,M     ;SAVE 1ST CHAR
0FBB 23                         INX     H       ;POINT NEXT
0FBC 0E20                       MVI     C,' '   ;DEFAULT FOR 1 CHAR VAR
0FBE CD2A1B                     CALL    NUMER   ;GO SEE IF 2ND IS NUMERIC
0FC1 C2F40F                     JNZ     LDFN    ;BRIF NOT
0FC4 23                         INX     H       ;POINT NEXT
0FC5 4F                         MOV     C,A     ;SAVE THE CHAR
0FC6 CF                 LDV1:   RST     1       ;GET NEXT CHAR
0FC7 FE24                       CPI     '$'     ;TEST IF STRING
0FC9 F5                         PUSH    PSW     ;SAVE STATUS
0FCA C2D30F                     JNZ     LDV2    ;BRIF NOT
0FCD 79                         MOV     A,C     ;GET LOW CHAR
0FCE F680                       ORI     80H     ;SET STRING
0FD0 4F                         MOV     C,A     ;SAVE IT
0FD1 23                         INX     H       ;SKIP $
0FD2 CF                         RST     1       ;SKIP SPACES
0FD3 FE28               LDV2:   CPI     '('     ;TEST IF PAREN
0FD5 CAD713                     JZ      LDV2A   ;BRIF IS
0FD8 E5                         PUSH    H       ;SAVE H,L
0FD9 50                         MOV     D,B     ;COPY B,C
0FDA 59                         MOV     E,C     ;TO D,E
```

```
0FD8 CD3418             CALL    SEARC   ;GO GET VAR ADDR IN D,E.
0FDE 2A5022     LDV:    LHLD    EXPRS   ;GET EXPR ADDR
 FE1 CD1918             CALL    SADR    ;GO STORE ADDRESS
0FE4 225022             SHLD    EXPRS   ;SAVE ADDRESS
0FE7 E8                 XCHG            ;H,L TO D,E
0FE8 E1                 POP     H       ;GET OLD H,L
0FE9 F1                 POP     PSW     ;GET STATUS
0FEA C21D11             JNZ     LOOKO   ;BRIF NOT STRING
0FED E8                 XCHG            ;GET OLD H,L
0FEE 36E7               MVI     M,0E7H  ;MARK AS STRING ADDRESS
0FF0 E8                 XCHG            ;RESTORE H,L
0FF1 C31D11             JMP     LOOKO   ;GO LOOK FOR OPCODE
0FF4 CD2118     LDFN:   CALL    ALPHA   ;GO SEE IF FUNCTION
0FF7 C2C60F             JNZ     LDV1    ;BRIF IT'S NOT
0FFA 28         LDFN1:  DCX     H       ;POINT BACK TO 1ST
0FFB 7E                 MOV     A,M     ;GET THAT CHAR
0FFC FE20               CPI     ' '     ;TEST IF SPACE
0FFE CAFA0F             JZ      LDFN1   ;LOOP IF TRUE
1001 E5                 PUSH    H       ;SAVE H,L
1002 11841C             LXI     D,RNDLI ;POINT LITERAL
1005 D7                 RST     2       ;GO COMPARE
1006 CA6310             JZ      LDRND   ;BRIF RND
1009 E1                 POP     H       ;GET H,L
100A E5                 PUSH    H       ;RESAVE
100B 11211F             LXI     D,FNLIT ;POINT LITERAL
100E D7                 RST     2       ;GO SEE IF FN X
100F CA3E10             JZ      FNL     ;BRIF IS
 012 E1                 POP     H       ;GET H,L
 013 E5                 PUSH    H       ;RESAVE
1014 11971D             LXI     D,PILIT ;POINT LIT
1017 D7                 RST     2       ;GO COMPARE
1018 CA7510             JZ      LDPI    ;BRIF PI
101B E1         FUNC0:  POP     H       ;GET H,L
101C 11981C             LXI     D,FUNCT ;POINT FUNCTION TABLE
101F E5         FUNC1:  PUSH    H       ;SAVE POINTER
1020 CD861F             CALL    SEEK    ;GO SEARCH FUNCTION TABLE
1023 CA3610             JZ      FUNC4   ;BRIF FUNCTION NOT FOUND
1026 1A                 LDAX    D       ;GET A BYTE LOW
1027 4F                 MOV     C,A     ;SAVE IT
1028 13                 INX     D       ;POINT NEXT
1029 1A                 LDAX    D       ;GET HI BYTE
102A 47                 MOV     B,A     ;SAVE IT (B,C = ADDR OF FUNC)
102B CF                 RST     1       ;SKIP BLANKS
102C FE28               CPI     '('     ;TEST FOR OPEN PAREN
102E C20F1C             JNZ     SNERR   ;BRIF MISSING PAREN
1031 13                 INX     D       ;POINT TYPE CODE
1032 1A                 LDAX    D       ;LOAD IT
1033 C37F10             JMP     LDFNC   ;CONTINUE
1036 E1         FUNC4:  POP     H       ;GET H,L
1037 46                 MOV     B,M     ;GET 1ST CHAR
1038 0E20               MVI     C,' '   ;SPACE 2ND CHAR
103A 23                 INX     H       ;POINT TO NEXT
 03B C3C60F             JMP     LDV1    ;BRIF VARIABLE
 03E D1         FNL:    POP     D       ;DUMMY RESET STACK POINTER
 03F CDC918             CALL    VAR     ;GO GET FN NAME
1042 42                 MOV     B,D     ;COPY TO B,C
1043 48                 MOV     C,E     ;SAME
1044 E3                 XCHG            ;SAVE H,L
```

```
1045 2A5022          LHLD    EXPRS   ;POINT EXPR STACK
1048 23              INX     H       ;POINT NEXT
1049 70              MOV     M,B     ;MOVE THE LETTER
104A 23              INX     H       ;POINT NEXT
104B 71              MOV     M,C     ;MOVE DIGIT ($??)
104C 23              INX     H       ;POINT NEXT
104D 36AF            MVI     M,0AFH  ;MOVE CODE
104F 79              MOV     A,C     ;GET LO NAME
1050 B7              ORA     A       ;TEST IT
1051 F25610          JP      FNL3    ;BRIF NOT STRING
1054 36CF            MVI     M,0CFH  ;MOVE CODE
1056 225022  FNL3:   SHLD    EXPRS   ;SAVE POINTER
1059 EB              XCHG            ;GET H,L
105A CF              RST     1       ;GET NEXT CHAR
105B FE28            CPI     '('     ;TEST IF OPEN PAREN
105D C20F1C          JNZ     SNERR   ;BRIF NOT
1060 C3950F          JMP     LOOKD   ;CONTINUE
1063 FE28    LDRND:  CPI     '('     ;TEST IF RND(X)
1065 CA1B10          JZ      FUNC0   ;BRIF IS
1068 E5              PUSH    H       ;ELSE, SAVE H,L
1069 21EA1D          LXI     H,ONE   ;USE RANGE (0,1)
106C EF              RST     5       ;LOAD FACC
106D CD840C          CALL    RND     ;GO GET RANDOM NUMBER
1070 E1              POP     H       ;RESTORE H,L
1071 D1              POP     D       ;RESTORE STACK POINTER
1072 C39F0F          JMP     LDF     ;ACT AS IF CONSTANT
1075 3C      LDPI:   INR     A       ;SET NON ZERO
1076 D1              POP     D       ;DUMMY STACK POP
1077 F5              PUSH    PSW     ;SAVE STATUS
1078 E5              PUSH    H       ;SAVE H,L
1079 11A21D          LXI     D,PI    ;GET ADDRESS OF 3.1415
107C C3DE0F          JMP     LDV     ;GO ACT LIKE VARIABLE
107F D1      LDFNC:  POP     D       ;POP THE STACK
1080 EB              XCHG            ;FLIP/FLOP
1081 2A5022          LHLD    EXPRS   ;GET ADDR
1084 23              INX     H       ;POINT NEXT
1085 70              MOV     M,B     ;HIGH ADDR
1086 23              INX     H       ;POINT NEXT
1087 71              MOV     M,C     ;LOW ADDR
1088 23              INX     H       ;POINT NEXT
1089 77              MOV     M,A     ;CODE
108A 225022          SHLD    EXPRS   ;SAVE ADDR
108D EB              XCHG            ;RESTORE H,L
108E C3950F          JMP     LOOKD   ;NEXT MUST BE DATA TOO
1091 FE2D    LDDTN:  CPI     '-'     ;TEST IF UNARY MINUS
1093 C2A510          JNZ     LDDTP   ;BRIF NOT
1096 EB              XCHG            ;SAVE H,L
1097 2A5022          LHLD    EXPRS   ;GET EXPR END
109A 23              INX     H       ;POINT ONE MORE
109B 3661            MVI     M,61H   ;CODE FOR NEG
109D 225022          SHLD    EXPRS   ;RESTORE PTR
10A0 EB              XCHG            ;RESTORE H,L
10A1 23      SKPP:   INX     H       ;POINT PAST THIS BYTE
10A2 C3950F          JMP     LOOKD   ;NEXT MUST BE DATA
10A5 FE2B    LDDTP:  CPI     '+'     ;TEST IF UNARY PLUS
10A7 CAA110          JZ      SKPP    ;IGNORE IF IS
10AA FE28            CPI     '('     ;ELSE, TEST IF OPEN PAREN
10AC CA0811          JZ      OBRCE   ;BRIF IS
```

```
10AF FE27           CPI     27H       ;TEST IF LITERAL (SINGLE QUOTE)
10B1 CA8910         JZ      LITST     ;BRIF IS
10B4 FE22           CPI     '"'       ;TEST IF LITERAL
10B6 C20F1C         JNZ     SNERR     ;BRIF NOT CON, FUNCTION, OR VAR
10B9 4F       LITST: MOV    C,A       ;SAVE DELIMITER
10BA 112021         LXI     D,STRIN   ;POINT BUFFER
10BD 06FF           MVI     B,0FFH    ;INIT CTR
10BF 23       LIT1:  INX    H         ;POINT NEXT
10C0 7E             MOV     A,M       ;LOAD NEXT
10C1 13             INX     D         ;POINT NEXT
10C2 12             STAX    D         ;STORE IT
10C3 B7             ORA     A         ;TEST IF END
10C4 CA0F1C         JZ      SNERR     ;BRIF ERROR
10C7 04             INR     B         ;COUNT IT
10C8 B9             CMP     C         ;TEST IF END OF STRING
10C9 C2BF10         JNZ     LIT1      ;BRIF NOT
10CC 23             INX     H         ;POINT NEXT
10CD 112021         LXI     D,STRIN   ;POINT BEGIN
10D0 78             MOV     A,B       ;GET COUNT
10D1 12             STAX    D         ;PUT COUNT
10D2 1F             RAR               ;DIVIDE BY TWO
10D3 3C             INR     A         ;PLUS ONE
10D4 4F             MOV     C,A       ;SAVE IT
10D5 0600           MVI     B,0       ;ZERO HIGH
10D7 E5             PUSH    H         ;SAVE PTR
10D8 2A6922         LHLD    SPCTR     ;GET CTR
10DB 09             DAD     B         ;PLUS OLD
10DC 226922         SHLD    SPCTR     ;UPDATE IT
10DF D1             POP     D         ;GET OLD H,L
10E0 210000         LXI     H,0       ;GET A ZERO
10E3 E5       LIT2:  PUSH   H         ;GET 2 WORK BYTES
10E4 0D             DCR     C         ;SUB 1 FROM COUNT
10E5 C2E310         JNZ     LIT2      ;CONTINUE
10E8 39             DAD     SP        ;GET ADDR OF STACK
10E9 D5             PUSH    D         ;SAVE PTR TO STMT
10EA EB             XCHG              ;SAVE H,L IN D,E
10EB 2A5022         LHLD    EXPRS     ;GET START OF EXPR
10EE 23             INX     H         ;PLUS ONE
10EF 72             MOV     M,D       ;HI BYTE
10F0 23             INX     H         ;POINT NEXT
10F1 73             MOV     M,E       ;LO BYTE
10F2 23             INX     H         ;POINT NEXT
10F3 36E7           MVI     M,0E7H    ;TYPE CODE
10F5 225022         SHLD    EXPRS     ;SAVE ADDR
10F8 EB             XCHG              ;D,E BACK TO H,L
10F9 112021         LXI     D,STRIN   ;POINT STRING AREA
10FC 1A             LDAX    D         ;GET COUNT
10FD 3C             INR     A         ;ADD ONE TO COUNT
10FE 47             MOV     B,A       ;SAVE CTR
10FF 1A       LIT3:  LDAX   D         ;GET A BYTE
1100 77             MOV     M,A       ;STORE IT
1101 23             INX     H         ;POINT NEXT
1102 13             INX     D         ;DITTO
1103 05             DCR     B         ;DECR CTR
1104 C2FF10         JNZ     LIT3      ;LOOP
1107 E1             POP     H         ;RESTORE H,L
1108 C31D11         JMP     LOOKO     ;NEXT IS OP
110B EB       OBRCE: XCHG             ;SAVE H,L
```

```
110C 216822             LXI     H,PARCT ;POINT PAREN COUNT
110F 34                 INR     M       ;ADD 1
1110 2A5022             LHLD    EXPRS   ;GET ADDR
1113 23                 INX     H       ;POINT NEXT
1114 3605               MVI     M,05H   ;PUT CODE
1116 225022             SHLD    EXPRS   ;SAVE ADDR
1119 EB                 XCHG            ;RESTORE H,L
111A C3A110             JMP     SKPP    ;GO SKIP CHAR
111D CF         LOOKO:  RST     1       ;SKIP BLANKS
111E FE2B               CPI     '+'     ;TEST IF PLUS
1120 0621               MVI     B,21H   ;CODE
1122 CA5811             JZ      OP1     ;BRIF IS
1125 FE2D               CPI     '-'     ;TEST IF MINUS
1127 0625               MVI     B,25H
1129 CA5811             JZ      OP1     ;BRIF IS
112C FE2F               CPI     '/'     ;TEST IF DIVIDE
112E 0645               MVI     B,45H   ;CODE
1130 CA5811             JZ      OP1     ;BRIF IS
1133 FE5E               CPI     '↑'     ;TEST IF EXPON
1135 0681               MVI     B,81H   ;CODE
1137 CA5811             JZ      OP1     ;BRIF IS
113A FE29               CPI     ')'     ;TEST IF CLOSE PAREN
113C CAAC11             JZ      OP3     ;BRIF IS
113F FE2C               CPI     ','     ;TEST IF COMMA
1141 CA9711             JZ      OP2     ;BRIF IS
1144 FE2A               CPI     '*'     ;TEST IF MULTIPLY
1146 0641               MVI     B,41H   ;CODE
1148 CA5811             JZ      OP1     ;BRIF IS
                ; ELSE MUST BE END OF EXPRESSION
114B 3A6822     ENDXP:  LDA     PARCT   ;GET OPEN PAREN COUNT
114E B7                 ORA     A       ;TEST IT
114F C20F1C             JNZ     SNERR   ;BRIF # OF ('S NOT = # OF )'S
1152 225622             SHLD    ADDR3   ;SAVE ADDR OF STMT
1155 C38A11             JMP     EVAL    ;GO EVALUATE
1158 E5         OP1:    PUSH    H       ;SAVE PLACE IN ASCII EXPRESSION
1159 110501             LXI     D,105H  ;D=BYTE COUNT, E=CODE FOR "("
115C 2A5022             LHLD    EXPRS   ;POINT TO LAST BYTE
115F 78                 MOV     A,B     ;B&E3 -> C
1160 E6E3               ANI     0E3H
1162 4F                 MOV     C,A
                ; INSERT ( AND EVALUATE IF PRECEDENCE REDUCTION,
                ;   ELSE INNSERT OP CODE
1163 7E         OPLP1:  MOV     A,M     ;GET TYPE CODE FROM EXPRESSION
1164 F5                 PUSH    PSW     ;SAVE
1165 E603               ANI     03H     ;GET LENGTH
1167 14         OPLP2:  INR     D       ;BUMP BYTE COUNT
1168 2B                 DCX     H       ;EXPRESSION POINTER
1169 3D                 DCR     A       ;LOOP MOVES TO NEXT ELEMENT
116A C26711             JNZ     OPLP2
116D F1                 POP     PSW     ;RESTORE TYPE CODE
116E E6E3               ANI     0E3H    ;MASK FOR VARIABLE
1170 FEE3               CPI     0E3H    ;WE SKIP OVER VARIABLES
1172 CA6311             JZ      OPLP1   ;BR IF TYPE = E3 OR E7
1175 B9                 CMP     C       ;PRECEDENCE REDUCTION?
1176 D28111             JNC     INS     ;IF NC, YES, INSERT 05
1179 2A5022             LHLD    EXPRS   ;NO, INSERT OP CODE BEFORE VAR AT
117C E7                 RST     4       ;SKIP OVER VARIABLE
117D FD                 DB      -3 AND 0FFH
```

```
117E 1604              MVI     D,4     ;BYTE COUNT
1180 58                MOV     E,B     ;INSERT THIS OP CODE
1181 43        INS:    MOV     B,E     ;SAVE FOR BRANCH AFTER INSERTION
1182 23        INS1:   INX     H       ;BUMP POINTER
1183 4E                MOV     C,M     ;PICK UP BYTE
1184 70                MOV     M,B     ;PUT DOWN REPLACEMENT
1185 41                MOV     B,C     ;SAVE FOR NEXT LOOP
1186 15                DCR     D       ;DONE?
1187 C28211            JNZ     INS1    ;IF NZ, NO
118A 225022            SHLD    EXPRS   ;STORE POINTER
118D E1                POP     H       ;RESTORE ASCII EXPRESSION POINTER
118E 78                MOV     A,E     ;GET FLAG SAVED IN E
118F FE05              CPI     05H     ;STORED A "("?
1191 C2A110            JNZ     SKPP    ;IF NZ, NO, PROCESS NEXT ELEMENT
1194 C38711            JMP     OP4     ;YES, GO EVALUATE
1197 3A6822    OP2:    LDA     PARCT   ;GET OPEN PAREN COUNT
119A B7                ORA     A       ;TEST IT
119B CA4811            JZ      ENDXP   ;BRIF END OF EXPR
119E EB                XCHG            ;ELSE SAVE H,L
119F 2A5022            LHLD    EXPRS   ;GET EXPR BEGIN
11A2 23                INX     H       ;POINT NEXT
11A3 3601              MVI     M,01H   ;MOVE A COMMA
11A5 225022            SHLD    EXPRS   ;UPDATE POINTER
11A8 EB                XCHG            ;FLIP BACK
11A9 C3A110            JMP     SKPP
11AC 3A6822    OP3:    LDA     PARCT   ;GET OPEN PAREN COUNT
11AF 3D                DCR     A       ;SUBTRACT ONE
11B0 326822            STA     PARCT   ;SAVE IT
11B3 FA0F1C            JM      SNERR   ;BRIF TOO MANY )'S
11B6 23                INX     H       ;POINT NEXT SOURCE
               OP4:
11B7 225622            SHLD    ADDR3   ;SAVE ADDR
11BA 2A5022    EVAL:   LHLD    EXPRS   ;GET END OF EXPR
11BD 010000            LXI     B,0     ;INIT B,C TO ZERO
11C0 04        EV1:    INR     B       ;COUNT EACH BYTE
11C1 7E                MOV     A,M     ;GET CODE IN REG A
11C2 2B                DCX     H       ;POINT NEXT
11C3 FEE3              CPI     0E3H    ;TEST IF DATA
11C5 C2D011            JNZ     EV2     ;BRIF NOT DATA
11C8 2B        EV1A:   DCX     H       ;POINT NEXT
11C9 2B                DCX     H       ;DITTO
11CA 04                INR     B       ;BUMP CTR
11CB 04                INR     B       ;BY TWO
11CC 0C                INR     C       ;COUNT THE TERM
11CD C3C011            JMP     EV1     ;LOOP
11D0 FEAF      EV2:    CPI     0AFH    ;TEST IF NUMERIC USER FN
11D2 CAB10E            JZ      FN      ;BRIF IS
11D5 FECF              CPI     0CFH    ;TEST IF STRING USER FN
11D7 CAB10E            JZ      FN      ;BRIF IS
11DA F5                PUSH    PSW     ;ELSE, SAVE STATUS
11DB E6E3              ANI     0E3H    ;MASK IT
11DD FEA3              CPI     0A3H    ;TEST IF NUMERIC FUNCTION
11DF CAF011            JZ      EV2A    ;BRIF IS
11E2 FEC3              CPI     0C3H    ;TEST IF STRING FUNCTION
11E4 CAF011            JZ      EV2A    ;BRIF IS
11E7 F1                POP     PSW     ;RESTORE CODE
11E8 FEE7              CPI     0E7H    ;TEST IF STRING ADDR
11EA CAC811            JZ      EV1A    ;BRIF IS
```

```
11ED C37812             JMP     EV5       ;BR AROUND
11F0 23         EV2A:   INX     H         ;RESET TO TYPE CODE
11F1 225222             SHLD    ADDR1     ;SAVE ADDR
11F4 D1                 POP     D         ;DUMMY POP
11F5 C5                 PUSH    B         ;SAVE CTRS
11F6 2B                 DCX     H         ;POINT TO LOW JMP ADDR
11F7 5E                 MOV     E,M       ;LOW BYTE
11F8 2B                 DCX     H         ;POINT BACK
11F9 56                 MOV     D,M       ;HIGH BYTE
11FA 225422             SHLD    ADDR2     ;SAVE LOCATION
11FD 210712             LXI     H,EV3     ;GET RETURN ADDRESS
1200 E5                 PUSH    H         ;SAVE ON STACK
1201 D5                 PUSH    D         ;SAVE ADDRESS
1202 CD741C             CALL    ARG       ;GO GET 1ST ARG
1205 E1                 POP     H         ;GET H,L ADDRESS
1206 E9                 PCHL              ;GO EXECUTE THE FUNCTION
1207 =          EV3     EQU     $         ;FUNCTIONS RETURN HERE
1207 2A5422             LHLD    ADDR2     ;GET ADDR FUNC
120A 23                 INX     H         ;POINT LO
120B 23                 INX     H         ;POINT TYPE
120C 7E                 MOV     A,M       ;LOAD IT
120D E6E0               ANI     0E0H      ;MASK IT
120F FEC0               CPI     0C0H      ;TEST IF STRING
1211 CA4C12             JZ      EV4       ;BRIF IS
1214 C1                 POP     B         ;GET CTRS
1215 2A6922             LHLD    SPCTR     ;GET COUNTER
1218 23                 INX     H         ;PLUS
1219 23                 INX     H         ;TWO WORDS
121A 226922             SHLD    SPCTR     ;STORE IT
121D 210000             LXI     H,0       ;LOAD ZERO TO H,L
1220 E5                 PUSH    H         ;GET BLOCK OF
1221 E5                 PUSH    H         ;BYTES
1222 39                 DAD     SP        ;GET STACK ADDR
1223 C5                 PUSH    B         ;SAVE CTRS
1224 E5                 PUSH    H         ;SAVE ADDR
1225 DF                 RST     3         ;GO STORE THE VARIABLE
1226 3EE3               MVI     A,0E3H    ;TYPE=NUM
1228 D1         EV3A:   POP     D         ;GET ADDR IN STACK
1229 2A5222             LHLD    ADDR1     ;GET ADDR LST ARG
122C 77                 MOV     M,A       ;STORE TYPE CODE
122D 2B                 DCX     H         ;POINT ONE BACK
122E 73                 MOV     M,E       ;STORE LO ADDR
122F 2B                 DCX     H         ;POINT BACK
1230 72                 MOV     M,D       ;STORE HI ADDR
1231 2A5422             LHLD    ADDR2     ;GET LOCATION FUNCTION
1234 23                 INX     H         ;POINT LO
1235 23                 INX     H         ;POINT TYPE
1236 7E                 MOV     A,M       ;LOAD TYPE
1237 46                 MOV     B,M       ;GET TYPE
1238 E7                 RST     4         ;ADJUST H,L
1239 FD                 DB      -3 AND 0FFH
123A 78                 MOV     A,B       ;LOAD TYPE
123B C1                 POP     B         ;RESTORE CTRS
123C E618               ANI     18H       ;ISOLATE #ARGS
123E 1F                 RAR               ;SHIFT RIGHT
123F 1F                 RAR               ;AGAIN
1240 1F                 RAR               ;ONCE MORE
1241 57                 MOV     D,A       ;SAVE IT
```

```
1242 82                 ADD     D       ;TIMES 2
1243 82                 ADD     D       ;TIMES 3
1244 04                 INR     B       ;POINT
1245 04                 INR     B       ;LST POSIT IN LOC
1246 CDE21A             CALL    SQUIS   ;GO COMPRESS STACK
1249 C3BA11             JMP     EVAL    ;START AT BEGINNING
124C 112021     EV4:    LXI     D,STRIN ;POINT STRING BUFFER
124F 1A                 LDAX    D       ;LOAD IT
1250 1F                 RAR             ;DIVIDE BY TWO
1251 3C                 INR     A       ;ADD 1
1252 2A6922             LHLD    SPCTR   ;GET SP COUNT
1255 4F                 MOV     C,A     ;SAVE LO
1256 0600               MVI     B,0     ;SET HI
1258 09                 DAD     B       ;ADD NUMBER WORDS
1259 226922             SHLD    SPCTR   ;SAVE SP COUNT
125C 210000             LXI     H,0     ;GET SOME ZEROS
125F C1                 POP     B       ;GET CTRS
1260 E5         EV4A:   PUSH    H       ;GET 1 WORD
1261 3D                 DCR     A       ;DECR CTR
1262 C26012             JNZ     EV4A    ;LOOP
1265 39                 DAD     SP      ;GET ADDRESS IN H,L
1266 C5                 PUSH    B       ;RE-SAVE CTRS
1267 E5                 PUSH    H       ;SAVE ADDR
1268 1A                 LDAX    D       ;GET COUNT
1269 3C                 INR     A       ;PLUS ONE
126A 47                 MOV     B,A     ;SAVE IT
126B 1A         EV4B:   LDAX    D       ;GET A BYTE
126C 77                 MOV     M,A     ;STORE IT
126D 13                 INX     D       ;POINT NEXT
126E 23                 INX     H       ;DITTO
126F 05                 DCR     B       ;DECR CTR
1270 C26B12             JNZ     EV4B    ;LOOP
1273 3EE7               MVI     A,0E7H  ;TYPE CODE
1275 C32812             JMP     EV3A    ;CONTINUE
1278 FE05       EV5:    CPI     05H     ;TEST IF OPEN PAREN
127A C29612             JNZ     EV6     ;BRIF NOT
127D 3E01               MVI     A,1     ;DELETE 1 BYTE
127F CDE21A             CALL    SQUIS   ;GO COMPRESS IT
1282 2A5622             LHLD    ADDR3   ;RESTORE STMT POINTER
1285 3A7220             LDA     DIMSW   ;GET SUBSR SWITCH
1288 B7                 ORA     A       ;TEST IT
1289 CA1D11             JZ      LOOKO   ;BRIF NOT IN SUBSCRIPT
128C 3A6822             LDA     PARCT   ;GET OPEN PAREN COUNT
128F B7                 ORA     A       ;TEST
1290 C21D11             JNZ     LOOKO   ;BRIF NOT ZERO
1293 C3BA11             JMP     EVAL    ;ELSE EVALUATE COMPLETE SUBSCR
1296 B7         EV6:    ORA     A       ;TEST IF END OF EXPRESSION
1297 C2C712             JNZ     EV9     ;BRIF NOT
129A 3A7220             LDA     DIMSW   ;GET DIM SW
129D B7                 ORA     A       ;TEST IT
129E C49D13             CNZ     EDM1    ;BRIF NOT OFF
12A1 79                 MOV     A,C     ;GET TERM COUNT
12A2 FE01               CPI     1       ;TEST IF ONE
12A4 C2081C             JNZ     STERR   ;ERROR IF NOT ONE
12A7 23                 INX     H       ;POINT HIGH ADDR
12A8 23                 INX     H       ;SAME
12A9 56                 MOV     D,M     ;HIGH TO D
12AA 23                 INX     H       ;POINT LOW
```

```
12AB 5E              MOV     E,M       ;LOW TO E
12AC CD8313          CALL    EVLD      ;GO LOAD VALUE
12AF 2A6922          LHLD    SPCTR     ;GET STACK CTR
12B2 7D        EV7:  MOV     A,L       ;GET LO BYTE
12B3 B4              ORA     H         ;PLUS HI
12B4 CABC12          JZ      EV8       ;BRIF ZERO
12B7 D1              POP     D         ;RETURN 2 BYTES
12B8 2B              DCX     H         ;DECR CTR
12B9 C3B212          JMP     EV7       ;LOOP
12BC 3A7220    EV8:  LDA     DIMSW     ;GET DIM SW
12BF B7              ORA     A         ;TEST IT
12C0 C4C413          CNZ     EDM4      ;BRIF ON
12C3 2A5622          LHLD    ADDR3     ;RESTORE STMT PTR
12C6 C9              RET               ;RETURN TO STMT PROCESSOR
12C7 FE21      EV9:  CPI     21H       ;TEST IF PLUS
12C9 111B13          LXI     D,FADDJ   ;ADDR
12CC CAF912          JZ      EV10      ;BRIF IS
12CF FE25            CPI     25H       ;TEST IF MINUS
12D1 110C17          LXI     D,FSUB    ;ADDR
12D4 CAF912          JZ      EV10      ;BRIF IS
12D7 FE41            CPI     41H       ;TEST IF MUL
12D9 111817          LXI     D,FMUL    ;ADDR
12DC CAF912          JZ      EV10      ;BRIF IS
12DF FE45            CPI     45H       ;TEST IF DIV
12E1 119B17          LXI     D,FDIV    ;ADDR
12E4 CAF912          JZ      EV10      ;BRIF IS
12E7 FE01            CPI     01H       ;TEST IF COMMA
12E9 CA7713          JZ      EVCOM     ;BRIF IS
12EC FE61            CPI     61H       ;TEST IF UNARY MINUS
12EE CA6313          JZ      EVNEG     ;BRIF IS
12F1 FE81            CPI     81H       ;TEST IF EXPONENTIAL
12F3 112313          LXI     D,POWER   ;ADDR
12F6 C20B1C          JNZ     STERR     ;ERROR IF NOT
12F9 23        EV10: INX     H         ;POINT TO
12FA 23              INX     H         ;1ST DATA
12FB C5              PUSH    B         ;SAVE CTRS
12FC D5              PUSH    D         ;SAVE ROUTINE ADDR
12FD 56              MOV     D,M       ;HIGH TO D
12FE 23              INX     H         ;POINT NEXT
12FF 5E              MOV     E,M       ;LOW TO E
1300 E5              PUSH    H         ;SAVE POINTER
1301 CD8313          CALL    EVLD      ;GO LOAD VALUE
1304 E1              POP     H         ;RESTORE H,L
1305 23              INX     H         ;POINT 2ND DATA
1306 23              INX     H         ;SAME
1307 56              MOV     D,M       ;HIGH TO D
1308 23              INX     H         ;POINT NEXT
1309 5E              MOV     E,M       ;LOW TO E
130A 23              INX     H         ;POINT NEXT
130B 3A8E22          LDA     NS        ;GET PREV TYPE
130E BE              CMP     M         ;TEST THIS TYPE
130F C20F1C          JNZ     SNERR     ;BRIF MIXED MODE
1312 2B              DCX     H         ;POINT BACK
1313 E3              XTHL              ;POP ADDR FROM STACK, PUSH H ONTO
1314 015213          LXI     B,EV11    ;RETURN ADDRESS
1317 C5              PUSH    B         ;SAVE ON STACK
1318 E5              PUSH    H         ;SAVE JUMP ADDR
1319 EB              XCHG              ;PUT VAR ADDR TO H,L
```

```
131A C9              RET             ;FAKE CALL TO ROUTINE
131B FEE7    FADDJ:  CPI     0E7H    ;TEST IF STRINGS
131D CA260D          JZ      CONCA   ;BRIF IS
1320 C33716          JMP     FADD    ;ELSE, GO ADD
1323 E5      POWER:  PUSH    H       ;SAVE ADDR OF VAR
1324 212F22          LXI     H,TEMP1 ;POINT SAVE AREA
1327 DF              RST     3       ;SAVE X
1328 E1              POP     H       ;RESTORE H,L
1329 EF              RST     5       ;LOAD IT
132A CDCE18          CALL    FTEST   ;TEST FOR ZERO
132D CAD608          JZ      SGN1    ;GIVE RESULT = 1 IF POWER = 0
1330 214722          LXI     H,TEMP7 ;POINT SAVE AREA
1333 DF              RST     3       ;SAVE B
1334 212F22          LXI     H,TEMP1 ;POINT X
1337 EF              RST     5       ;GO LOAD IT
1338 CDCE18          CALL    FTEST   ;TEST FOR ZERO
133B C8              RZ              ;0**X = 0
133C CD1308          CALL    LN      ;GET NATURAL LNRITHM
133F 214722          LXI     H,TEMP7 ;POINT B
1342 CD1817          CALL    FMUL    ;GO MULTIPLY
1345 C36A08          JMP     EXP     ;GET EXP FUNC
                ; X**B = EXP(B*LN(X))
1348 212F22  XSQR:   LXI     H,TEMP1 ;POINT X
1348 EF              RST     5       ;LOAD X
134C 212F22          LXI     H,TEMP1 ;POINT X
134F C31817          JMP     FMUL    ;TIMES X
1352 E1      EV11:   POP     H       ;GET H,L
1353 C1              POP     B       ;GET CTRS
1354 2B              DCX     H       ;POINT BACK
1355 2B              DCX     H       ;AND AGAIN
1356 CD0018          CALL    GTEMP   ;GO SAVE FACC
1359 E7              RST     4       ;ADJUST H,L
135A F9              DB      -7 AND 0FFH
135B 3E04            MVI     A,4     ;DELETE 4 BYTES
135D CDE21A          CALL    SQUIS   ;GO COMPRESS
1360 C38A11          JMP     EVAL    ;CONTINUE
1363 23      EVNEG:  INX     H       ;POINT BACK TO OP
1364 C5              PUSH    B       ;SAVE CTRS
1365 E5              PUSH    H       ;SAVE H,L
1366 23              INX     H       ;DITTO
1367 56              MOV     D,M     ;GET HI BYTE
1368 23              INX     H       ;POINT NEXT
1369 5E              MOV     E,M     ;GET LO BYTE
136A CD8313          CALL    EVLD    ;GO LOAD VAR
136D CD7A0C          CALL    NEG     ;GO NEGATE IT
1370 E1              POP     H       ;GET LOCATION
1371 C1              POP     B       ;GET CTRS
1372 CD0018          CALL    GTEMP   ;GO STORE FACC IN STACK
1375 E7              RST     4       ;ADJUST H,L
1376 FC              DB      -4 AND 0FFH
1377 3E01    EVCOM:  MVI     A,1     ;DELETE 1 BYTE
1379 CDE21A          CALL    SQUIS   ;COMPRESS
137C 216B22          LXI     H,CMACT ;GET COUNT
137F 34              INR     M       ;INCR
1380 C38A11          JMP     EVAL    ;CONTINUE
1383 23      EVLD:   INX     H       ;POINT TYPE
1384 7E              MOV     A,M     ;LOAD IT
1385 323E22          STA     NS      ;SAVE IT
```

```
1388 EB                XCHG                ;SAVE H,L IN D,E
1389 FEE7              CPI     0E7H        ;TEST IF STRING
138B C22800            JNZ     RST5        ;LOAD FLOATING POINT
138E 112021            LXI     D,STRIN     ;POINT BUFFER
1391 7E                MOV     A,M         ;GET COUNT
1392 3C                INR     A           ;ADD ONE
1393 47                MOV     B,A         ;SAVE COUNT
1394 7E        EVLD1:  MOV     A,M         ;GET NEXT
1395 12                STAX    D           ;STORE IT
1396 23                INX     H           ;POINT NEXT
1397 13                INX     D           ;DITTO
1398 05                DCR     B           ;DECR COUNT
1399 C29413            JNZ     EVLD1       ;LOOP
139C C9                RET                 ;RETURN
               ;
139D 79        EDM1:   MOV     A,C         ;GET ITEM COUNT
139E E5                PUSH    H           ;SAVE H,L
139F FE01              CPI     1           ;TEST IF 1
13A1 C2B013            JNZ     EDM3        ;BRIF NOT
13A4 0604              MVI     B,4         ;GET COUNT
13A6 212F22            LXI     H,TEMP1     ;POINT AREA
13A9 CD5E1C            CALL    ZEROM       ;GO ZERO IT
13AC E1        EDM2A:  POP     H           ;RESTORE H,L
13AD 0E01              MVI     C,1         ;SET COUNT
13AF C9                RET                 ;RETURN
13B0 FE02      EDM3:   CPI     2           ;TEST IF 2
13B2 C20F1C            JNZ     SNERR       ;ELSE, ERROR
13B5 E7                RST     4           ;POINT 2ND ARG
13B6 05                DB      5
13B7 56                MOV     D,M         ;GET HI ADDR
13B8 23                INX     H           ;POINT NEXT
13B9 5E                MOV     E,M         ;GET LO ADDR
13BA CD8313            CALL    EVLD        ;LOAD THE ARG
13BD 212F22            LXI     H,TEMP1     ;POINT AREA
13C0 DF                RST     3           ;SAVE THE ARG
13C1 C3AC13            JMP     EDM2A       ;CONTINUE
13C4 CD351F    EDM4:   CALL    FACDE       ;CONVERT FACC TO D,E
13C7 D5                PUSH    D           ;PUT D,E TO B,C
13C8 C1                POP     B
13C9 C5                PUSH    B           ;SAVE COL
13CA 212F22            LXI     H,TEMP1     ;POINT 2ND ARGUMENT
13CD EF                RST     5           ;LOAD IT IN FACC
13CE CD351F            CALL    FACDE       ;CONVERT TO D,E
13D1 C1                POP     B           ;GET COL
13D2 AF                XRA     A           ;GET A ZERO
13D3 327220            STA     DIMSW       ;RESET SW
13D6 C9                RET                 ;RETURN
13D7 78        LDV2A:  MOV     A,B         ;GET HI NAME
13D8 F680              ORI     80H         ;SET BIT
13DA 47                MOV     B,A         ;RESTORE
13DB C5                PUSH    B           ;SAVE NAME
13DC EB                XCHG                ;SAVE H,L IN D,E
13DD 3A6822            LDA     PARCT       ;GET PAREN COUNT
13E0 F5                PUSH    PSW         ;SAVE
13E1 AF                XRA     A           ;CLEAR REG A
13E2 326822            STA     PARCT       ;RESET COUNT
13E5 2A6922            LHLD    SPCTR       ;GET STACK COUNTER
13E8 E5                PUSH    H           ;SAVE IT
```

```
13E9 210000            LXI     H,0       ;GET A ZERO
13EC 226922            SHLD    SPCTR     ;RESET CTR
13EF 2A5022            LHLD    EXPRS     ;GET EXPRST
13F2 E5                PUSH    H         ;SAVE IT
13F3 23                INX     H         ;POINT NEXT
13F4 3600              MVI     M,0       ;SET NEW START
13F6 225022            SHLD    EXPRS     ;SAVE IT
13F9 3A7220            LDA     DIMSW     ;GET PREV SE
13FC F5                PUSH    PSW       ;SAVE IT
13FD EB                XCHG              ;RESTORE H,L
13FE 3EFF              MVI     A,0FFH    ;GET ON VALUE
1400 327220            STA     DIMSW     ;SET SW
1403 CD950F            CALL    LOOKD     ;RECURSIVE CALL
1406 F1                POP     PSW       ;GET DIM SW
1407 327220            STA     DIMSW     ;REPLACE IT
140A 225622            SHLD    ADDR3     ;SAVE H,L
140D E1                POP     H         ;GET EXPRST
140E 225022            SHLD    EXPRS     ;SAVE IT
1411 E1                POP     H         ;GET STACK COUNTER
1412 226922            SHLD    SPCTR     ;RESTORE IT
1415 F1                POP     PSW       ;GET PAREN COUNT
1416 326822            STA     PARCT     ;RESTORE IT
1419 E1                POP     H         ;GET NAM
141A D5                PUSH    D         ;SAVE ROW
141B C5                PUSH    B         ;SAVE COL
141C EB                XCHG              ;PUT NAME IN D,E
141D CD3418            CALL    SEARC     ;GO FIND ADDRESS (PUT IN D,E)
1420 D1                POP     D         ;GET ADDR
1421 C1                POP     B         ;RESTORE COL
1422 D1                POP     D         ;RESTORE ROW
1423 CD8518            CALL    SUBSC     ;GET SUBSCRIPT (RETURNS ADDR IN H,L)
1426 EB                XCHG              ;SAVE IN D,E
1427 2A5622            LHLD    ADDR3     ;GET H,L
142A E5                PUSH    H         ;SAVE ON STACK
142B C3DE0F            JMP     LDV       ;CONTINUE
                ;      PAGE
```

```
                        ;
142E =                  FIN     EQU     $
                        ;
                        ; FLOATING POINT INPUT CONVERSION ROUTINE
                        ;
                        ; THIS SUBROUTINE CONVERTS AN ASCII STRING OF CHARACTERS
                        ; TO THE FLOATING POINT ACCUMULATOR.  THE INPUT FIELD
                        ; MAY CONTAIN ANY VALID NUMBER, INCLUDING SCIENTIFIC
                        ; NOTATION (NNN.NNNNE+NN).
                        ; THE INPUT STRING IS TERMINATED BY ANY NON-NUMBERIC CHAR
                        ;
                        ;
142E EB                         XCHG            ;PUT ADDR TO D,E
142F 0E00                       MVI     C,0     ;INITIAL VALUE EXCESS DIGIT COUNT
1431 CD8814                     CALL    FIN8    ;GET INTEGER PORTION
1434 0600                       MVI     B,0     ;CLEAR DIGIT COUNT
1436 FE2E                       CPI     '.'     ;TEST IF DEC-POINT
1438 C23E14                     JNZ     FIN2    ;BRIF NOT
143B CDA214                     CALL    FIN9    ;GET FRACTION
143E F1                 FIN2:   POP     PSW     ;GET SIGN
143F F618                       ORI     24      ;SET UP FOR FLOAT
1441 325822                     STA     FACC
1444 78                         MOV     A,B     ;GET # FRACTION DIGITS
1445 81                         ADD     C       ;+ EXCESS DIGITS
1446 F5                         PUSH    PSW     ;SAVE POWER OF TEN
1447 D5                         PUSH    D       ;SAVE PTR
1448 CDDD16                     CALL    FNORM   ;NORMALIZE NUMBER
1448 1A                         LDAX    D       ;GET NEXT CHARACTER
144C FE45                       CPI     'E'     ;TEST IF EXPONENT
144E C26C14                     JNZ     FIN4    ;BRIF NOT
1451 215C22                     LXI     H,FTEMP ;POINT SAVE AREA
1454 DF                         RST     3       ;SAVE ACC
1455 D1                         POP     D       ;RESTORE PTR
1456 13                         INX     D       ;SKIP 'E'
1457 CD8814                     CALL    FIN8    ;GET NUMERIC EXP
145A 3A5B22                     LDA     FACC+3  ;GET EXPONENT
145D C1                         POP     B       ;EXPONENT SIGN
145E 04                         INR     B       ;TEST
145F F26414                     JP      FIN3    ;BRIF NOT NEG
1462 2F                         CMA             ;NEGATE EXPONENT
1463 3C                         INR     A
1464 C1                 FIN3:   POP     B       ;POWER OF TEN
1465 80                         ADD     B       ;ADD EXPONENT
1466 F5                         PUSH    PSW     ;SAVE COUNT
1467 215C22                     LXI     H,FTEMP ;RESTORE NUMBER
146A D5                         PUSH    D       ;SAVE PTR
146B EF                         RST     5       ;LOAD IT
146C E1                 FIN4:   POP     H       ;RESTORE PTR
146D F1                         POP     PSW     ;RESTORE COUNT
146E C8                 FIN5:   RZ              ;RETURN IF ZERO
146F E5                         PUSH    H       ;SAVE H,L
1470 219E1D                     LXI     H,TEN   ;POINT CONSTANT: 10
1473 FA8014                     JM      FIN7    ;BRIF DIVIDE NEEDED
1476 3D                         DCR     A       ;DECR COUNT
1477 F5                         PUSH    PSW     ;SAVE COUNT
1478 CD1817                     CALL    FMUL    ;GO MULTIPLY BY 10
147B F1                 FIN6:   POP     PSW     ;RESTORE COUNT
147C E1                         POP     H       ;RESTORE H,L
```

```
147D C36E14             JMP     FIN5    ;CONTINUE
 80 3C          FIN7:   INR     A       ;INCR COUNT
 81 F5                  PUSH    PSW     ;SAVE COUNT
1482 CD9B17             CALL    FDIV    ;GO DIVIDE BY 10
1485 C37B14             JMP     FIN6    ;LOOP

                ; FIN8  CONVERT NUMBER STRING TO FACC
                ; ON ENTRY, C=INIT VALUE EXCESS DIGIT COUNT
                ;               DE=INPUT STRING
                ; ON EXIT, SIGN IS ON STACK
                ;       B=DIGIT COUNT
                ;       C=EXCESS DIGIT COUNT

1488 215822     FIN8:   LXI     H,FACC  ;CLEAR FACC
148B 0604               MVI     B,4
148D CD5E1C             CALL    ZEROM
1490 210080             LXI     H,8000H ;ASSUME MINUS
1493 1A                 LDAX    D       ;GET CHAR
1494 FE2D               CPI     '-'
1496 CAA014             JZ      FIN8A
1499 65                 MOV     H,L     ;NOPE, MUST BE PLUS
                                        ;(B IS CLEARED BY ZEROM)
149A FE2B               CPI     '+'
149C CAA014             JZ      FIN8A
149F 1B                 DCX     D       ;NEITHER, BACK UP POINTER
14A0 E3         FIN8A:  XTHL            ;GET RETURN, PUSH SIGN
 4A1 E5                 PUSH    H       ;RESTORE RETURN
 4A2 13         FIN9:   INX     D       ;POINT NEXT
14A3 1A                 LDAX    D       ;GET CHAR
14A4 FE30               CPI     '0'     ;TEST IF LESS ZERO
14A6 D8                 RC              ;RETURN IF IS
14A7 FE3A               CPI     '9'+1   ;TEST IF GT NINE
14A9 D0                 RNC             ;RETURN IF IS
14AA 05                 DCR     B       ;DIGIT COUNT
14AB D5                 PUSH    D       ;SAVE PTR
14AC C5                 PUSH    B       ;SAVE COUNTERS
14AD CDD514             CALL    FMTEN   ;MULTIPLY FACC*TEN
14B0 B7                 ORA     A       ;TEST FOR OVERFLOW
14B1 CABE14             JZ      FIN8    ;BRIF NO OVERFLOW
14B4 216022             LXI     H,FTEMP+4
14B7 EF                 RST     5       ;RESTORE OLD FACC
14B8 C1                 POP     B       ;RESTORE COUNTERS
14B9 0C                 INR     C       ;EXCESS DIGIT
14BA D1                 POP     D
14BB C3A214             JMP     FIN9
14BE C1         FIN8:   POP     B       ;RSTORE COUNTERS
14BF D1                 POP     D       ;& PTR
14C0 1A                 LDAX    D       ;GET THE DIGIT
14C1 E60F               ANI     0FH     ;MASK OFF ZONE
14C3 215B22             LXI     H,FACC+3        ;POINT ACC
14C6 86                 ADD     M       ;ADD
14C7 77                 MOV     M,A     ;STORE
 4C8 2B                 DCX     H       ;POINT NEXT
 4C9 7E                 MOV     A,M     ;LOAD
14CA CE00               ACI     0       ;PLUS CARRY
14CC 77                 MOV     M,A     ;STORE
14CD 2B                 DCX     H       ;POINT NEXT
14CE 7E                 MOV     A,M     ;LOAD
```

```
14CF CE00               ACI     0       ;PLUS CARRY
14D1 77                 MOV     M,A     ;STORE
14D2 C3A214             JMP     FIN9    ;LOOP

                ; MULTIPLY FACC BY TEN

14D5 216022     FMTEN:  LXI     H,FTEMP+4
14D8 DF                 RST     3       ;SAVE FACC
14D9 CDE514             CALL    FIND    ;*2
14DC CDE514             CALL    FIND    ;*4
14DF 216322             LXI     H,FTEMP+7
14E2 CDE814             CALL    FIND0   ;*5
14E5 215B22     FIND:   LXI     H,FACC+3;DOUBLE FACC
14E8 115B22     FIND0:  LXI     D,FACC+3
14EB 0604               MVI     B,4     ;BYTE COUNT
14ED C3F018             JMP     FADDT   ;ADD & RETURN
                        ;PAGE
```

```
                        ;
  F0 =                  FOUT    EQU     $
                        ;
                        ; FLOATING POINT OUTPUT FORMAT ROUTINE
                        ;
                        ; THIS SUBROUTINE CONVERTS A NUMBER IN FACC TO A
                        ; FORMAT SUITABLE FOR PRINTING.  THAT IS, THE
                        ; NUMBER WILL BE IN SCIENTIFIC NOTATION IF EXPONENT
                        ; IS > 5 OR < -2, OTHERWISE IT WILL BE ZERO SUPPRESSED
                        ; ON BOTH SIDES.
                        ;
14F0 115822             FOUT:   LXI     D,FACC+3        ;POINT LSB
14F3 1A                         LDAX    D       ;LOAD IT
14F4 F607                       ORI     07H     ;MASK FOR OUTPUT
14F6 12                         STAX    D       ;REPLACE
14F7 CDCE18                     CALL    FTEST   ;GET SIGN OF NUMBER
14FA 3620                       MVI     M,' '   ;DEFAULT SPACE
14FC F20115                     JP      FOUT0   ;BRIF NOT MINUS
14FF 362D                       MVI     M,'-'   ;MOVE DASH
1501 23                 FOUT0:  INX     H       ;POINT NEXT
1502 C20B15                     JNZ     FOUT2   ;BRIF NOT ZERO
1505 3630                       MVI     M,'0'   ;MOVE THE ZERO
1507 23                         INX     H       ;POINT NEXT
1508 3620                       MVI     M,' '   ;MOVE SPACE FOLLOWING
150A C9                         RET             ;RETURN
150B 3A5822             FOUT2:  LDA     FACC    ;GET SIGN & EXP
150E CDDC18                     CALL    FEXP    ;EXPAND EXPONENT
1511 C21615                     JNZ     FOUTV   ;BRIF NOT ZERO
1514 3E80                       MVI     A,80H   ;SET NEG
1516 E680               FOUTV:  ANI     80H     ;ISOLATE
1518 327522                     STA     DEXP    ;SAVE SIGN
151B E5                         PUSH    H       ;SAVE H,L
151C 3A5822             FOUT3:  LDA     FACC    ;GET SIGN & EXP
151F CDDC18                     CALL    FEXP    ;EXPAND EXP
1522 FE01                       CPI     1       ;TEST RANGE
1524 F23D15                     JP      FOUT6   ;BRIF IN RANGE
1527 217522             FOUT4:  LXI     H,DEXP  ;POINT DEC.EXP
152A 34                         INR     M       ;INCR IT
152B 219E1D                     LXI     H,TEN   ;POINT CONST: 10
152E F23715                     JP      FOUT5   ;BRIF POS.
1531 CD1817                     CALL    FMUL    ;MULTIPLY
1534 C31C15                     JMP     FOUT3   ;LOOP
1537 CD9817             FOUT5:  CALL    FDIV    ;DIVIDE
153A C31C15                     JMP     FOUT3   ;LOOP
153D FE05               FOUT6:  CPI     5       ;TEST HIGH RANGE
153F F22715                     JP      FOUT4   ;BRIF 5 OR GREATER
1542 215C22                     LXI     H,FTEMP ;POINT SAVE AREA
1545 DF                         RST     3       ;STORE IT
1546 3A5822                     LDA     FACC    ;GET EXPONENT
1549 CDDC18                     CALL    FEXP    ;EXPAND
154C 0E06                       MVI     C,6     ;DIGIT COUNT
154E CD8215                     CALL    FOUT8   ;SHIFT LEFT
1551 FE0A                       CPI     10      ;TEST IF DECIMAL DIGIT
1553 FA5D15                     JM      FOUTU   ;BRIF LT
1556 215C22                     LXI     H,FTEMP ;POINT SAVE AREA
1559 EF                         RST     5       ;LOAD IT
155A C32715                     JMP     FOUT4   ;ONCE MORE
155D CD7015             FOUTU:  CALL    FOUT9   ;PUT DIGIT
```

```
1560 AF        FOUT7:  XRA     A       ;CLEAR STATUS
1561 325822            STA     FACC    ;AND OVERFLOW
1564 CDD514            CALL    FMTEN   ;MULTIPLY BY TEN
1567 CD7015            CALL    FOUT9   ;PUT DIGIT
156A C26015            JNZ     FOUT7   ;LOOP
156D C39915            JMP     FOUTH   ;GO AROUND
1570 F630      FOUT9:  ORI     30H     ;DEC. ZONE
1572 E1                POP     H       ;GET RETURN ADDR
1573 E3                XTHL            ;EXCH WITH TOP (PTR)
1574 77                MOV     M,A     ;PUT DIGIT
1575 23                INX     H       ;POINT NEXT
1576 79                MOV     A,C     ;GET COUNT
1577 FE06              CPI     6       ;TEST IF 1ST
1579 C27F15            JNZ     FOUTA   ;BRIF NOT
157C 362E              MVI     M,'.'   ;MOVE DEC. PT.
157E 23                INX     H       ;POINT NEXT
157F E3        FOUTA:  XTHL            ;EXCH WITH RTN
1580 0D                DCR     C       ;DECR COUNT
1581 E9                PCHL            ;RETURN
1582 5F        FOUTB:  MOV     E,A     ;SAVE BIT COUNT
1583 AF                XRA     A       ;CLEAR ACC FLAGS
1584 325822            STA     FACC    ;AND OVERFLOW
1587 215B22    FOUTC:  LXI     H,FACC+3        ;POINT LSB
158A 0604              MVI     B,4     ;BYTE COUNT
158C 7E        FOUTD:  MOV     A,M     ;GET A BYTE
158D 17                RAL             ;SHIFT LEFT
158E 77                MOV     M,A     ;STORE
158F 28                DCX     H       ;POINT NEXT
1590 05                DCR     B       ;DECR CTR
1591 C28C15            JNZ     FOUTD   ;LOOP
1594 1D                DCR     E       ;DECR BIT CTR
1595 C28715            JNZ     FOUTC   ;LOOP
1598 C9                RET             ;RETURN
1599 E1        FOUTH:  POP     H       ;GET PTR
159A 3645              MVI     M,'E'   ;EXPONENT
159C 23                INX     H       ;POINT NEXT
159D 3A7522            LDA     DEXP    ;GET EXPONENT
15A0 3628              MVI     M,'+'   ;DEFAULT
15A2 57                MOV     D,A     ;SAVE NUMBER
15A3 B7                ORA     A       ;TEST IT
15A4 F28015            JP      FOUTI   ;BRIF POS
15A7 362D              MVI     M,'-'   ;ELSE, DASH
15A9 E67F              ANI     7FH     ;STRIP DUMB SIGN
15AB 2F                CMA             ;COMPLEMENT
15AC 3C                INR     A       ;PLUS ONE (TWOS COMP)
15AD 57                MOV     D,A     ;SAVE IT
15AE 2F                CMA             ;RE-COMPLEMENT
15AF 3C                INR     A       ;PLUS ONE
15B0 23        FOUTI:  INX     H       ;POINT NEXT
15B1 E5                PUSH    H       ;SAVE PTR
15B2 1EFF              MVI     E,-1 AND 0FFH   ;INIT CTR (TENS)
15B4 1C        FOUTJ:  INR     E       ;ADD ONE
15B5 D60A              SUI     10      ;LESS 10
15B7 F28415            JP      FOUTJ   ;LOOP
15BA C60A              ADI     10      ;CORRECT UNITS
15BC 47                MOV     B,A     ;SAVE UNITS
15BD 7B                MOV     A,E     ;GET TENS
15BE CD7015            CALL    FOUT9   ;OUTPUT
```

```
15C1 78                 MOV     A,B     ;GET UNITS
15C2 CD7015             CALL    FOUT9   ;OUTPUT
15C5 E1                 POP     H       ;GET PTR
15C6 3620               MVI     M,' '   ;SPACE AFTER
15C8 7A                 MOV     A,D     ;GET DEC EXPON
15C9 B7                 ORA     A       ;SET FLAGS
15CA F2D315             JP      FOUTK   ;BRIF POS.
15CD FEFE               CPI     -2 AND 0FFH     ;TEST FOR MIN
15CF D8                 RC              ;RETURN IF LESS THAN -2
15D0 C3D615             JMP     FOUTL   ;GO AROUND
15D3 FE06       FOUTK:  CPI     6       ;TEST IF TOO BIG
15D5 D0                 RNC             ;RETURN IF 6 OR GREATER
15D6 4F         FOUTL:  MOV     C,A     ;SAVE EXPONENT
15D7 0605               MVI     B,5     ;CTR
15D9 3620       FOUTM:  MVI     M,' '   ;SPACE OUT EXPONENT
15D8 2B                 DCX     H       ;POINT PRIOR
15DC 05                 DCR     B       ;DECR CTR
15DD C2D915             JNZ     FOUTM   ;LOOP
15E0 EB                 XCHG            ;FLIP/FLOP
15E1 78                 MOV     A,E     ;GET LOW BYTE
15E2 D605               SUI     5       ;POINT TO DOT
15E4 6F                 MOV     L,A     ;PUT DOWN
15E5 7A                 MOV     A,D     ;GET HIGH
15E6 DE00               SBI     0       ;IN CASE OF BORROW
15E8 67                 MOV     H,A     ;PUT DOWN
15E9 79                 MOV     A,C     ;GET EXPONENT
15EA B7                 ORA     A       ;TEST SIGN
15EB CAFC15             JZ      FOUTO   ;BRIF ZERO
15EE FA1116             JM      FOUTR   ;BRIF NEGATIVE
15F1 46         FOUTN:  MOV     B,M     ;GET HIGH BYTE
15F2 23                 INX     H       ;POINT NEXT
15F3 7E                 MOV     A,M     ;GET LOW BYTE
15F4 70                 MOV     M,B     ;SHIFT DOT TO RIGHT
15F5 2B                 DCX     H       ;POINT BACK
15F6 77                 MOV     M,A     ;MOVE THE DIGIT LEFT
15F7 23                 INX     H       ;POINT NEXT
15F8 0D                 DCR     C       ;DECR CTR
15F9 C2F115             JNZ     FOUTN   ;LOOP
15FC EB         FOUTO:  XCHG            ;POINT END
15FD 7E         FOUTP:  MOV     A,M     ;GET A DIGIT/DOT
15FE FE30               CPI     '0'     ;TEST FOR TRAILING ZERO
1600 C20916             JNZ     FOUTQ   ;BRIF NOT
1603 3620               MVI     M,' '   ;SPACE FILL
1605 2B                 DCX     H       ;POINT PRIOR
1606 C3FD15             JMP     FOUTP   ;LOOP
1609 FE2E       FOUTQ:  CPI     '.'     ;TEST FOR TRAILING DOT
160B 23                 INX     H       ;JUST IN CASE NOT
160C C0                 RNZ             ;RETURN IF NOT
160D 2B                 DCX     H       ;RESET PTR
160E 3620               MVI     M,' '   ;SPACE IT OUT
1610 C9                 RET             ;RETURN
1611 FEFF       FOUTR:  CPI     0FFH    ;TEST IF -1
1613 C21F16             JNZ     FOUTS   ;ELSE -2
1616 2B                 DCX     H       ;POINT SIGNIFICANT
1617 7E                 MOV     A,M     ;GET THE CHAR
1618 362E               MVI     M,'.'   ;MOVE THE DOT
161A 23                 INX     H       ;POINT NEXT
161B 77                 MOV     M,A     ;SHIFT THE DIGIT
```

```
161C C3FC15             JMP     FOUTO   ;GO ZERO SUPPRESS
161F 28         FOUTS:  DCX     H       ;POINT ONE TO LEFT
1620 7E                 MOV     A,M     ;PICK UP DIGIT
1621 3630               MVI     M,'0'   ;REPLACE
1623 23                 INX     H       ;POINT RIGHT
1624 77                 MOV     M,A     ;PUT THE DIGIT
1625 62                 MOV     H,D     ;GET LOW ADDR
1626 6B                 MOV     L,E     ;POINT LAST DIGIT
1627 0606               MVI     B,6     ;CTR
1629 2B         FOUTT:  DCX     H       ;POINT PRITO
162A 7E                 MOV     A,M     ;GET A DIGIT
162B 23                 INX     H       ;POINT
162C 77                 MOV     M,A     ;PUT IT ONE TO RIGHT
162D 2B                 DCX     H       ;POINT
162E 05                 DCR     B       ;DECR CTR
162F C22916             JNZ     FOUTT   ;LOOP
1632 362E               MVI     M,'.'   ;MOVE THE DOT
1634 C3FC15             JMP     FOUTO   ;CONTINUE
                ;
1637 =          FADD    EQU     $
                ;
                ;
                ; FLOATING POINT ADD THE NUMBER AT (H,L) TO THE FACC
                ;
                ;
1637 23                 INX     H       ;POINT FIRST DIGIT
1638 7E                 MOV     A,M     ;LOAD IT
1639 B7                 ORA     A       ;TEST IT
163A CACE18             JZ      FTEST   ;BRIF ZERO
163D 2B                 DCX     H       ;POINT BACK
163E CDCE18             CALL    FTEST   ;GO TEST SIGN OF FACC
1641 CA2800             JZ      RST5    ;JUST LOAD IF FACC = 0
1644 CDDC18             CALL    FEXP    ;GO GET EXPONENT
1647 47                 MOV     B,A     ;SAVE EXPONENT
1648 7E                 MOV     A,M     ;GET EXPONENT OF ADDR
1649 CDDC18             CALL    FEXP    ;GO GET EXPONENT
164C 4F                 MOV     C,A     ;SAVE THE EXPONENT
164D 90                 SUB     B       ;GET DIFFERENCE OF TWO EXPONENTS
164E CA6316             JZ      FADD4   ;BRIF THEY'RE EQ
1651 F25616             JP      FADD3   ;BRIF DIFFERENCE IS POSITIVE
1654 2F                 CMA             ;COMPLEMENT ACC
1655 3C                 INR     A       ;PLUS ONE (TWO'S COMPLEMENT)
1656 FE18       FADD3:  CPI     24      ;COMPARE DIFFERENCE TO MAX
1658 DA6316             JC      FADD4   ;BRIF LESS
165B 78                 MOV     A,B     ;GET EXPON OF ADDUEND
165C 91                 SUB     C       ;GET TRUE DIFFERENCE AGAIN
165D F2CE18             JP      FTEST   ;BRIF FACC > ADDER
1660 C32800             JMP     RST5    ;ELSE, ADDER > FACC
1663 F5         FADD4:  PUSH    PSW     ;SAVE DIFFERENCE
1664 C5                 PUSH    B       ;SAVE EXPONENTS
1665 115C22             LXI     D,FTEMP ;GET ADDR OF TEMP ACC
1668 CD561C             CALL    CPY4H
166B C1                 POP     B       ;GET EXPONENTS
166C F1                 POP     PSW     ;GET DIFFERENCE
166D CA9416             JZ      FADD9   ;JUST ADD IF ZERO
1670 215D22             LXI     H,FTEMP+1       ;DEFAULT
1673 F5                 PUSH    PSW     ;SAVE DIFFERENCE
1674 78                 MOV     A,B     ;GET FACC EXPON
```

```
1675 91                 SUB     C       ;MINUS FTEMP EXPON
1676 F23616             JP      FADD6   ;BRIF TEMP MUST BE SHIFTED
 679 215822             LXI     H,FACC  ;POINT FLOAT ACC
167C 79                 MOV     A,C     ;GET EXPONENT, SIGN
167D E67F               ANI     7FH     ;STRIP EXP SIGN
167F 4F                 MOV     C,A     ;PUT BACK
1680 7E                 MOV     A,M     ;GET THE EXP
1681 E680               ANI     80H     ;STRIP OFF OLD EXPON
1683 B1                 ORA     C       ;MOVE ADDER EXPON TO IT
1684 77                 MOV     M,A     ;REPLACE
1685 23                 INX     H       ;POINT FIRST DATA BYTE
1686 F1         FADD6:  POP     PSW     ;GET DIFFER
1687 4F                 MOV     C,A     ;SAVE IT
1688 0603       FADD7:  MVI     B,3     ;LOOP CTR (INNER)
168A AF                 XRA     A       ;INIT CARRY TO Z
168B E5                 PUSH    H       ;SAVE ADDR
168C CDFB18             CALL    FSHFT   ;GO SHIFT
168F E1                 POP     H       ;GET ADDR
1690 0D                 DCR     C       ;DECR CTR
1691 C28816             JNZ     FADD7   ;LOOP
1694 =          FADD9   EQU     $
1694 215C22             LXI     H,FTEMP
1697 3A5822             LDA     FACC    ;GET EXPONENT
169A AE                 XRA     M       ;SEE IF SIGNS THE SAME
169B 115B22             LXI     D,FACC+3;POINT LEAST SIGN BYTE
169E 215F22             LXI     H,FTEMP+3
16A1 FABC16             JM      FADDA   ;BRIF SIGNS DIFFERENT
 6A4 CDEE18             CALL    FADT3   ;ADD 3 BYTES
 6A7 D2CE18             JNC     FTEST   ;BRIF NO OVERFLOW
16AA EB                 XCHG            ;POINT HL TO FACC
16AB CD8917             CALL    SVSGN   ;SAVE SIGN, RETURN EXPONENT
16AE 3C                 INR     A       ;INCREMENT EXPONENT
16AF CD9117             CALL    RSSGN   ;RESTORE SIGN TO EXPONENT
16B2 23                 INX     H       ;POINT DATA
16B3 37                 STC             ;SET CY
16B4 0603               MVI     B,3     ;CTR
16B6 CDFB18             CALL    FSHFT   ;GO SHIFT IT
16B9 C3CE18             JMP     FTEST   ;RETURN
16BC =          FADDA   EQU     $
16BC 0603               MVI     B,3
16BE CDE318             CALL    FSUBT   ;SUBTRACT
16C1 D2DD16             JNC     FNORM   ;BRIF NO BORROW
16C4 215B22             LXI     H,FACC+3;MUST NEGATE
16C7 0603               MVI     B,3
16C9 37         FNEG:   STC
16CA 7E         FNEG1:  MOV     A,M     ;GET BYTE
16CB 2F                 CMA
16CC D2D116             JNC     FNEG2
16CF C601               ADI     1       ;INCREMENT + COMPLEMENT=NEGATE
16D1 77         FNEG2:  MOV     M,A
16D2 2B                 DCX     H
16D3 05                 DCR     B
16D4 C2CA16             JNZ     FNEG1
 6D7 CDDD16             CALL    FNORM
 6DA C37A0C             JMP     NEG     ;REVERSE SIGN
                        ;PAGE
```

```
                        ;
16DD =                  FNORM   EQU     $
                        ;
                        ;
                        ; NORMALIZE THE FLOATING ACCUMULATOR
                        ; THAT IS, THE FIRST BIT MUST BE SIGNIFICANT
                        ;
                        ;
16DD 215822                     LXI     H,FACC+3        ;POINT LSB
16E0 7E                         MOV     A,M     ;LOAD IT
16E1 2B                         DCX     H       ;POINT PRIOR
16E2 B6                         ORA     M       ;MERGE
16E3 2B                         DCX     H       ;POINT PRIOR
16E4 B6                         ORA     M       ;MERGE
16E5 2B                         DCX     H
16E6 46                         MOV     B,M     ;SAVE EXPONENT
16E7 77                         MOV     M,A     ;CLEAR
16E8 C8                         RZ              ;RETURN ON NOTHING TO NORMALIZE
16E9 70                         MOV     M,B     ;RESTORE EXP
16EA C5                         PUSH    B       ;SAVE C FOR CALLER
16EB CD8917                     CALL    SVSGN   ;SAVE SIGN
16EE 77                         MOV     M,A     ;STORE EXPANDED EXPONENT
16EF 23                 FNRM1:  INX     H       ;POINT TO MOST SIGN BYTE
16F0 7E                         MOV     A,M     ;GET MSB
16F1 B7                         ORA     A       ;TEST IT
16F2 FA0517                     JM      FNRM3   ;BRIF NORMALIZED
16F5 23                         INX     H       ;POINT LSB
16F6 23                         INX     H
16F7 0603                       MVI     B,3     ;SHIFT COUNT
16F9 7E                 FNRM2:  MOV     A,M     ;SHIFT LEFT
16FA 17                         RAL
16FB 77                         MOV     M,A
16FC 2B                         DCX     H
16FD 05                         DCR     B
16FE C2F916                     JNZ     FNRM2
1701 35                         DCR     M       ;ADJUST EXPONENT
1702 C3EF16                     JMP     FNRM1   ;LOOP
1705 2B                 FNRM3:  DCX     H       ;POINT BACK TO EXPONENT
1706 7E                         MOV     A,M
1707 CD9117                     CALL    RSSGN   ;RESTORE SIGN
170A C1                         POP     B       ;RESTORE C
170B C9                         RET



                        ;
170C =                  FSUB    EQU     $
                        ;
                        ;
                        ; FLOATING POINT SUBTRACT THE NUMBER AT (H,L) FROM THE FACC
                        ;
                        ;
170C CD7A0C                     CALL    NEG     ;NEGATE FACC
170F CD3716                     CALL    FADD    ;ADD
1712 CD7A0C                     CALL    NEG     ;NEGATE RESULT
1715 C3CE18                     JMP     FTEST

                                ;PAGE
```

```
                        ;
1718 =                  FMUL    EQU     $
                        ;
                        ;
                        ; FLOATING POINT MULTIPLY THE NUMBER AT (H,L) TO THE FACC
                        ;
                        ;
1718 CDCE18                     CALL    FTEST   ;TEST FACC
1718 C8                         RZ              ;RETURN IF ZERO
171C 23                         INX     H       ;POINT 1ST DIGIT OF MULTIPLIER
171D 7E                         MOV     A,M     ;LOAD IT
171E 28                         DCX     H       ;RESTORE
171F B7                         ORA     A       ;TEST IF ZERO
1720 CA2800                     JZ      RST5    ;GO LOAD TO FACC IF IT IS
1723 E5                         PUSH    H       ;SAVE MULTIPLIER ADDRESS
1724 CD7F17                     CALL    MDSGN   ;GET SIGN PRODUCT, & BOTH EXPONENTS
1727 80                         ADD     B       ;ADD EXPONENTS
1728 CD9117                     CALL    RSSGN   ;RESTORE SIGN
1728 E1                         POP     H       ;RESTORE
172C 116522                     LXI     D,FTEMP+9       ;POINT TEMP STORAGE
172F 0603                       MVI     B,3     ;BYTE COUNT
1731 23                         INX     H       ;POINT MSD
1732 CD581C                     CALL    COPYH   ;MOVE MULTIPLIER
1735 215C22                     LXI     H,FTEMP ;POINT DIGIT 7 OF RESULT
1738 0606                       MVI     B,6     ;LOOP CTR
173A CD5E1C                     CALL    ZEROM   ;GO ZERO EIGHT BYTES
173D 115922                     LXI     D,FACC+1        ;POINT 1ST DIGIT OF ACC
1740 0603                       MVI     B,3     ;LOOP CTR
1742 1A                 FMUL5:  LDAX    D       ;GET AN ACC DIGIT PAIR
1743 77                         MOV     M,A     ;PUT TO TEMP STORAGE
1744 AF                         XRA     A       ;ZERO A
1745 12                         STAX    D       ;CLEAR ACC
1746 13                         INX     D       ;POINT NEXT
1747 23                         INX     H       ;DITTO
1748 05                         DCR     B       ;DECR CTR
1749 C24217                     JNZ     FMUL5   ;LOOP
174C 0E18                       MVI     C,24    ;OUTTER LOOP CTR
174E 0603               FMUL6:  MVI     B,3     ;CTR
1750 216522                     LXI     H,FTEMP+9       ;POINT MULTIPLIER
1753 AF                         XRA     A       ;CLEAR CY
1754 7E                 FMUL7:  MOV     A,M     ;GET BYTE
1755 1F                         RAR             ;SHIFT RIGHT
1756 77                         MOV     M,A     ;PUT DOWN
1757 23                         INX     H       ;POINT NEXT
1758 05                         DCR     B       ;DECR CTR
1759 C25417                     JNZ     FMUL7   ;LOOP
175C D26A17                     JNC     FMUL8   ;BRIF ZERO BIT
175F 115E22                     LXI     D,FTEMP+2       ;POINT RESULT
1762 216422                     LXI     H,FTEMP+8       ;POINT MULTIPLICAND
1765 0606                       MVI     B,6     ;SIX BYTE ADD
1767 CDF018                     CALL    FADDT   ;GO ADD
176A 0606               FMUL8:  MVI     B,6     ;SIZ BYTE SHIFT
176C 216422                     LXI     H,FTEMP+8       ;POINT MULTIPLICAND
176F AF                         XRA     A       ;CLEAR CY
1770 7E                 FMUL9:  MOV     A,M     ;GET BYTE
1771 17                         RAL             ;SHIFT LEFT
1772 77                         MOV     M,A     ;PUT BACT
1773 28                         DCX     H       ;POINT NEXT BYTE
```

```
1774 05                 DCR     B       ;DECR CTR
1775 C27017             JNZ     FMUL9   ;LOOP
1778 0D                 DCR     C       ;DEC BIT COUNT
1779 C24E17             JNZ     FMUL6   ;CONTINUE
177C C3DD16             JMP     FNORM   ;GO NORMALIZE

                ; MDSGN    GET SIGN PRODUCT AND EXPONENTS FOR MULT & DIV
                ; ON ENTRY:
                ;       (HL) = ONE NUMBER
                ;       (FACC)=THE OTHER
                ; ON RETURN:
                ;       A = EXPONENT OF FACC(EXPANDED)
                ;       B = OTHER EXPONENT
                ;       C = SIGN PRODUCT
                ;       HL DESTROYED

177F CD8917     MDSGN:  CALL    SVSGN   ;GET SIGN IN C, EXP IN A
1782 47                 MOV     B,A     ;SAVE EXPONENT
1783 215822             LXI     H,FACC
1786 79                 MOV     A,C     ;GET SIGN
1787 86                 ADD     M       ;MULTIPLY SIGNS
1788 77                 MOV     M,A     ;PUT DOWN


                ; SVSGN          GET SIGN AND EXP
                ; ON ENTRY:
                ;       (HL) = EXPONENT
                ; ON RETURN:
                ;       A = EXPANDED EXPONENT
                ;       C = SIGN IN HI ORDER BIT

1789 7E         SVSGN:  MOV     A,M     ;GET EXPONENT
178A E680               ANI     80H     ;ISOLATE SIGN
178C 4F                 MOV     C,A
178D 7E                 MOV     A,M
178E C3DC18             JMP     FEXP    ;EXPAND EXP AND RETURN


                ; RSSGN          RESTORE SIGN TO EXPONENT
                ; ON ENTRY:
                ;       (HL)=EXPONENT
                ;       A = EXPANDED EXPONENT
                ;       C = SIGN
                ; ON RETURN
                ;       A = EXPONENT
                ;       (HL) = EXPONENT WITH SIGN
                ;       Z,M BITS SET FOR EXPONENT

1791 CD7118     RSSGN:  CALL    FOVUN   ;CHECK FOR OVER/UNDERFLOW
1794 E67F               ANI     7FH     ;REMOVE EXPONENT SIGN
1796 81                 ORA     C       ;ADD SIGN
1797 77                 MOV     M,A     ;SET DOWN
1798 C3CE18             JMP     FTEST   ;SET Z,M BITS


                        ;PAGE
```

```
                        ;
1798 =                  FDIV    EQU     $
                        ;
                        ;
                        ; FLOATING POINT DIVIDE THE NUMBER AT (H,L) INTO THE FACC
                        ;
                        ;
1798 CDCE18                     CALL    FTEST   ;TEST IF FACC ZERO
179E C8                         RZ              ;RETURN IF IT IS
179F 23                         INX     H       ;POINT 1ST DIGIT OF DIVISOR
17A0 7E                         MOV     A,M     ;LOAD IT
17A1 2B                         DCX     H       ;POINT BACK
17A2 B7                         ORA     A       ;TEST IF ZERO
17A3 CA071C                     JZ      ZMERR   ;DIVISION BY ZERO = ERROR
17A6 E5                         PUSH    H       ;SAVE DIVISOR PTR
17A7 CD7F17                     CALL    MDSGN   ;GET SIGN ON STACK, EXPS INTO A,B
17AA 90                         SUB     B       ;SUBTRACT EXPONENTS
17AB 3C                         INR     A       ;PLUS ONE
17AC CD9117                     CALL    RSSGN   ;SET SIGN/EXPONENT IN FACC
17AF 115922                     LXI     D,FACC+1
17B2 215C22                     LXI     H,FTEMP ;POINT TEMPORARY STORAGE
17B5 3600                       MVI     M,0     ;CLEAR MSB
17B7 23                         INX     H       ;POINT NEXT
17B8 0603                       MVI     B,3     ;LOOP CTR
17BA 1A                 FDIV3:  LDAX    D       ;GET BYTE FROM FACC
17BB 77                         MOV     M,A     ;PUT TO FTEMP
17BC AF                         XRA     A       ;CLEAR A
17BD 12                         STAX    D       ;ZERO FACC
 7BE 23                         INX     H       ;POINT NEXT
17BF 13                         INX     D       ;DITTO
17C0 05                         DCR     B       ;DECR CTR
17C1 C2BA17                     JNZ     FDIV3   ;LOOP
17C4 D1                         POP     D       ;GET ADDR
17C5 0603                       MVI     B,3     ;LOOP CTR
17C7 13                         INX     D       ;POINT MSD OF DIVISOR
17C8 3600                       MVI     M,0     ;CLEAR MSB
17CA 23                         INX     H       ;POINT NEXT
17CB CD4D1C                     CALL    COPYD   ;GO MOVE IT
17CE 0E18                       MVI     C,24    ;OUTER LOOP CTR
17D0 115F22             FDIV5:  LXI     D,FTEMP+3       ;POINT DIVIDEND
17D3 216322                     LXI     H,FTEMP+7       ;AND DIVISOR
17D6 0604                       MVI     B,4     ;CTR
17D8 CDE318                     CALL    FSUBT   ;GO SUBTRACT
17DB D2EA17                     JNC     FDIV6   ;BRIF NO GO
17DE 115F22                     LXI     D,FTEMP+3       ;POINT DIVIDEND
17E1 216322                     LXI     H,FTEMP+7       ;AND DIVISOR
17E4 0604                       MVI     B,4     ;CTR
17E6 CDF018                     CALL    FADDT   ;GO RE-ADD
17E9 37                         STC             ;TURN ON CY
17EA 3F                 FDIV6:  CMC             ;REVERSE CY
17EB 0603                       MVI     B,3     ;CTR
17ED 215B22                     LXI     H,FACC+3        ;POINT LSB
17F0 7E                 FDIV7:  MOV     A,M     ;LOAD BYTE
17F1 17                         RAL             ;SHIFT LEFT
 7F2 77                         MOV     M,A     ;REPLACE
17F3 2B                         DCX     H       ;POINT NEXT
17F4 05                         DCR     B       ;DECR CTR
17F5 C2F017                     JNZ     FDIV7   ;LOOP
```

```
17F8 AF              XRA     A         ;CLEAR FLAGS
17F9 0604            MVI     B,4       ;CTR
17FB 215F22          LXI     H,FTEMP+3         ;POINT DIVIDEND
17FE 7E        FDIV8:MOV     A,M       ;LOAD BYTE
17FF 17              RAL               ;SHIFT LEFT
1800 77              MOV     M,A       ;REPLACE
1801 2B              DCX     H         ;POINT ENXT
1802 05              DCR     B         ;DECR CTR
1803 C2FE17          JNZ     FDIV8     ;LOOP
1806 0D              DCR     C         ;DECR OTR CTR
1807 C2D017          JNZ     FDIV5     ;LOOP
180A C3DD16          JMP     FNORM     ;WRAPUP
                ;
                ; UTILITY ROUTINE TO GET A VARIABLE'S ADDRESS TO H,L
                ;
180D 112021    GETST:LXI     D,STRIN ;POINT BUFFER
1810 0600            MVI     B,0       ;INIT CTR
1812 7E              MOV     A,M       ;GET THE CHAR
1813 FE22            CPI     '"'       ;TEST IF LIT TYPE
1815 CA2E18          JZ      GETS2     ;BRIF IS
1818 FE27            CPI     27H       ;TEST IF QUOTED LITERAL
181A CA2E18          JZ      GETS2     ;BRIF IS
181D FE2C      GETS1:CPI     ','       ;TEST IF COMMA
181F CA4118          JZ      GETS5     ;BRIF IS
1822 B7              ORA     A         ;TEST IF END
1823 CA4118          JZ      GETS5     ;BRIF IS
1826 04              INR     B         ;COUNT IT
1827 13              INX     D         ;POINT NEXT
1828 12              STAX    D         ;PUT CHAR
1829 23              INX     H         ;POINT NEXT
182A CF              RST     1         ;SKIP SPACES
182B C31D18          JMP     GETS1     ;LOOP
182E 4F        GETS2:MOV     C,A       ;SAVE DELIM
182F 23        GETS3:INX     H         ;SKIP THE QUOTE
1830 7E              MOV     A,M       ;GET NEXT CHAR
1831 B9              CMP     C         ;TEST IF END OF LITERAL
1832 CA3F18          JZ      GETS4     ;BRIF IS
1835 B7              ORA     A         ;TEST IF END LINE
1836 CA1F1C          JZ      CVERR     ;BRIF IS
1839 04              INR     B         ;COUNT IT
183A 13              INX     D         ;POINT NEXT
183B 12              STAX    D         ;PUT CHAR
183C C32F18          JMP     GETS3     ;LOOP
183F 23        GETS4:INX     H         ;SKIP END QUOTE
1840 CF              RST     1         ;SKIP TRAILING SPACES
1841 112021    GETS5:LXI     D,STRIN ;POINT BEGIN BUFFER
1844 78              MOV     A,B       ;GET COUNT
1845 12              STAX    D         ;PUT COUNT
1846 D1              POP     D         ;GET RETURN ADDR
1847 EB              XCHG              ;FLIP/FLOP
1848 E3              XTHL              ;PUT RET ON STACK, HL OF VAR IN HL
1849 D5              PUSH    D         ;SAVE H,L OF LOC
184A CD3106          CALL    LET2A     ;GO STORE STRING
184D E1              POP     H         ;RESTORE LOCATION
184E C9              RET               ;RETURN
184F CDC918    GETS8:CALL    VAR       ;GET VAR NAME
1852 D5              PUSH    D         ;SAVE ON STACK
1853 7A              MOV     A,D       ;GET HI BYTE
```

```
1854 B7                 ORA     A       ;TEST IF ARRAY
1855 F26C18             JP      GETS9   ;BRIF NOT
1858 CD3418             CALL    SEARC   ;GO GET ARRAY PARAMS
185B 3EFF               MVI     A,0FFH  ;TURN ON SW
185D 327220             STA     DIMSW   ;SET IT
1860 E3                 XTHL            ;SWAP ADDR ON STACK
1861 CD800F             CALL    EXPR    ;GO GET ROW, COL PTRS
1864 E3                 XTHL            ;SWAP ADDR ON STACK
1865 CD8518             CALL    SUBSC   ;GO POINT TO ENTRY
1868 EB                 XCHG            ;EXCHANGE
1869 E1                 POP     H       ;GET ADDRESS OF STMT
186A C1                 POP     B       ;GET NAME
186B C9                 RET             ;RETURN
186C CD3418     GETS9:  CALL    SEARC   ;FIND ADDR
186F C1                 POP     B       ;RESTORE NAME
1870 C9                 RET             ;RETURN
                ;
1871 =          FOVUN   EQU     $
                ;
                ; TEST EXPONENT FOR OVERFLO OR UNDERFLOW
                ;
1871 B7         OVUN:   ORA     A       ;TEST IT
1872 F27D18             JP      FOV1    ;BRIF POS.
1875 FEC1               CPI     0C1H    ;TEST FOR MAX NEG
1877 D0                 RNC             ;RETURN IF NO UNDER.
1878 3EC1               MVI     A,0C1H  ;SET EXPONENT AT MINIMUM
187A C32C1C             JMP     UNERR
187D FE40       FOV1:   CPI     40H     ;TEST MAX POS
187F D8                 RC              ;RETURN IF NO OVER.
1880 3E3F               MVI     A,3FH   ;SET EXPONENT AT MAXIMUM
1882 C3271C             JMP     OVERR
                ;
1885 =          SUBSC   EQU     $
                ;
                ;
                ; COMPUTES SUBSCR ADDR
                ; INPUT: B HAS ROW NUMBER (1ST SUB)
                ;         D HAS COL NUMBER (2ND SUB)
                ;         H HAS ADDR NAME
                ;
1885 D5                 PUSH    D       ;SAVE COL
1886 E7                 RST     4       ;ADJUST H,L
1887 FC                 DB      -4 AND 0FFH     ;BY FOUR
1888 56                 MOV     D,M     ;GET HI
1889 2B                 DCX     H       ;POINT LO
188A 5E                 MOV     E,M     ;GET LO
188B 7A                 MOV     A,D     ;GET HI
188C B8                 CMP     B       ;COMPARE
188D DA0F1C             JC      SNERR   ;BRIF EXCESS
1890 C29818             JNZ     SUB1    ;BRIF NOT EQUAL
1893 7B                 MOV     A,E     ;GET LO
1894 B9                 CMP     C       ;COMPARE
1895 DA0F1C             JC      SNERR   ;BRIF EXCESS
1898 2B         SUB1:   DCX     H       ;POINT HI COLS
1899 56                 MOV     D,M     ;LOAD IT
189A 2B                 DCX     H       ;POINT LO COLS
189B 5E                 MOV     E,M     ;LOAD IT
189C E3                 XTHL            ;SAVE ADDRESS
```

```
189D E5                 PUSH    H       ;SAVE SUB COL
189E D5                 PUSH    D       ;SAVE DIM COLS
189F 13                 INX     D       ;MAKE COLS=MAX+1 (ACCOUNT FOR 0 L  KE
18A0 210000             LXI     H,0     ;GET A ZERO
18A3 78         SUB2:   MOV     A,B     ;GET HI
18A4 B1                 ORA     C       ;PLUS LO
18A5 CAAD18             JZ      SUB3    ;BRIF ZERO
18A8 19                 DAD     D       ;ADD ONCE
18A9 0B                 DCX     B       ;SUB ONCE
18AA C3A318             JMP     SUB2    ;LOOP
18AD D1         SUB3:   POP     D       ;GET DIM COL
18AE C1                 POP     B       ;GET SUB COL
18AF 7A                 MOV     A,D     ;GET HI
18B0 B8                 CMP     B       ;COMPARE
18B1 DA0F1C             JC      SNERR   ;BRIF GT
18B4 C2BC18             JNZ     SUB4    ;BRIF NOT ZERO
18B7 7B                 MOV     A,E     ;GET LO
18B8 B9                 CMP     C       ;COMPARE
18B9 DA0F1C             JC      SNERR   ;BRIF GT
18BC 09         SUB4:   DAD     B       ;ADD TO PROD
18BD 29                 DAD     H       ;TIMES TWO
18BE 29                 DAD     H       ;TIMES FOUR
18BF 7D                 MOV     A,L     ;GET LOW
18C0 2F                 CMA             ;COMPLEMENT
18C1 C601               ADI     1       ;PLUS ONE
18C3 5F                 MOV     E,A     ;SAVE IT
18C4 7C                 MOV     A,H     ;GET HI
18C5 2F                 CMA             ;COMPLEMENT
18C6 CE00               ACI     0       ;PLUS CARRY
18C8 57                 MOV     D,A     ;SAVE
18C9 E1                 POP     H       ;GET ADDR (0,0)
18CA 19                 DAD     D       ;COMPUTE (I,J) RIGHT SIDE
18CB E7                 RST     4       ;ADJUST H,L
18CC FC                 DB      -4 AND 0FFH
18CD C9                 RET             ;RETURN
18CE =          FTEST   EQU     $
                ;
                ; TEST THE SIGN OF THE NUMBER IN THE FACC
                ; RETURN WITH S & Z SET TO SIGN
                ;
18CE 3A5922             LDA     FACC+1  ;GET MSD
18D1 B7                 ORA     A       ;TEST IT
18D2 C8                 RZ              ;RETURN IF ZERO
18D3 3A5822             LDA     FACC    ;GET SIGN&EXPON BYTE
18D6 F67F               ORI     7FH     ;TEST SIGN BIT ONLY
18D8 3A5822             LDA     FACC    ;RE-LOAD EXPON BYTE
18DB C9                 RET             ;THEN RETURN
18DC =          FEXP    EQU     $
                ;
                ; EXPAND EXPONENT INTO 8 BINARY BITS
                ;
18DC E67F               ANI     7FH     ;MASK MANTISA SIGN
18DE C640               ADI     40H     ;PROPAGATE CHAR SIGN TO LEFTMOST
18E0 EE40               XRI     40H     ;RESTORE ORIGINAL SIGN BIT
18E2 C9                 RET             ;RETURN
                ;
18E3 =          FSUBT   EQU     $
                ;
```

```
                ; SUBTRACT THE TWO MULTIPRECISION NUMBERS (D,E) & (H,L)
                ;
 8E3 AF                 XRA     A       ;TURN OFF CY
18E4 1A         FSB1:  LDAX    D       ;GET A BYTE
18E5 9E                SBB     M       ;SUB OTHER BYTE
18E6 12                STAX    D       ;PUT DOWN
18E7 1B                DCX     D       ;POINT NEXT
18E8 2B                DCX     H       ;DITTO
18E9 05                DCR     B       ;DECR CTR
18EA C2E418            JNZ     FSB1    ;LOOP
18ED C9                RET             ;RETURN
                ;
                ;
                ; ADD TWO MULTI-PRECISION NUMBERS (D,E) & (H,L)
                ;
18EE 0603       FADT3: MVI     B,3
18F0 AF         FADDT: XRA     A       ;CLEAR STATUS
18F1 1A         FAD1:  LDAX    D       ;GET BYTE
18F2 8E                ADC     M       ;ADD OTHER BYTE
18F3 12                STAX    D       ;PUT DOWN
18F4 1B                DCX     D       ;POINT NEXT
18F5 2B                DCX     H       ;DITTO
18F6 05                DCR     B       ;DECR LOOP CTR
18F7 C2F118            JNZ     FAD1    ;LOOP
18FA C9                RET             ;RETURN
                ;
18FB =          FSHFT  EQU     $
                ;
                ; INCREMENTING SHIFT RIGHT
                ;
18FB 7E                MOV     A,M     ;GET A BYTE
18FC 1F                RAR             ;SHIFT RIGHT
18FD 77                MOV     M,A     ;PUT DOWN
18FE 23                INX     H       ;POINT NEXT
18FF 05                DCR     B       ;DECR CTR
1900 C2FB18            JNZ     FSHFT   ;LOOP
1903 C9                RET             ;RETURN
                       ;PAGE
```

```
                        ;
1904 =                  TERMI   EQU     $
                        ;
                        ; READ A LINE FROM THE TTY
                        ; FIRST PROMPT WITH THE CHAR IN THE A REG
                        ; TERMINATE THE LINE WITH A X'00'
                        ; IGNORE EMPTY LINES
                        ; CONTROL C WILL CANCEL THE LINE
                        ; CONTROL O WILL TOGGLE THE OUTPUT SWITCH
                        ; RUBOUT WILL DELETE THE LAST CHAR INPUT
                        ;
                        ;
1904 324F22                     STA     PROMP   ;SAVE THE PROMPT CHAR
1907 21CE20             REIN:   LXI     H,IOBUF ;POINT TO INPUT BUFFER
190A 3600                       MVI     M,00H   ;MARK BEGIN
190C 23                         INX     H       ;POINT START
190D 3A4F22                     LDA     PROMP   ;GET THE PROMPT AGAIN
1910 CD4F19                     CALL    TESTO   ;WRITE TO TERMINAL
1913 FE3F                       CPI     '?'     ;TEST IF Q.M.
1915 C21D19                     JNZ     TREAD   ;BRIF NOT
1918 3E20                       MVI     A,' '   ;GET SPACE
191A CD4F19                     CALL    TESTO   ;WRITE TO TERMINAL
191D =                  TREAD   EQU     $
                                IF      NOT CPM
191D DB03                       IN      TTY+1   ;GET TTY STATUS
191F E602                       ANI     2       ;TEST IF RXRDY
1921 CA1D19                     JZ      TREAD   ;LOOP TILL CHAR
                                ENDIF
1924 CD3F1A                     CALL    GETCH   ;GO READ THE CHAR
1927 77                         MOV     M,A     ;PUT IN BUFFER
1928 FE0A                       CPI     0AH     ;TEST IF LINE FEED
192A CA1D19                     JZ      TREAD   ;IGNORE IF IT IS
192D FE0D                       CPI     0DH     ;TEST IF CR
192F C27519                     JNZ     NOTCR   ;BRIF NOT
1932 3A7120                     LDA     TAPES   ;GET PAPER TAPE SWITCH
1935 1F                         RAR             ;TEST IF LOAD
1936 D45A19                     CNC     CRLF    ;CR/LF IF NOT
1939 3600               CR1:    MVI     M,0     ;MARK END
193B 3A7420                     LDA     ILSW    ;GET INPUT LINE SW
193E B7                         ORA     A       ;TEST IT
193F C0                         RNZ             ;RETURN IF ON
1940 2B                         DCX     H       ;POINT PRIOR
1941 7E                         MOV     A,M     ;LOAD IT
1942 FE20                       CPI     20H     ;TEST IF SPACE
1944 CA3919                     JZ      CR1     ;BRIF SPACE
1947 B7                         ORA     A       ;TEST IF AT BEGINNING
1948 CA0719                     JZ      REIN    ;BRIF IS (NULL LINE)
194B 21CF20                     LXI     H,IOBUF+1       ;POINT BEGIN
194E C9                         RET             ;ELSE, RETURN
194F =                  TESTO   EQU     $
                                IF      NOT CPM
194F F5                         PUSH    PSW     ;SAVE CHAR
1950 DB03               TOUT1:  IN      TTY+1   ;GET STATUS
1952 1F                         RAR             ;TEST IF TXRDY
1953 D25019                     JNC     TOUT1   ;LOOP TILL READY
1956 F1                         POP     PSW     ;GET CHAR
1957 D302                       OUT     TTY     ;WRITE IT
                                ENDIF
```

```
                        IF      CPM
                        PUSH    B       ;BIOS CALL DESTROYS C,DE
                        PUSH    D
                        PUSH H
                        MOV     C,A     ;OUTPUT BYTE
                        CALL    BTOUT   ;CALL BIOS
                        POP H
                        POP     D       ;RESTORE
                        POP     B
                        ENDIF
                        IF      LARGE   ;SAVE SPACE ONLY IN 8+K VERSIONS
                        DB      0,0,0   ;SAVE ROOM FOR CALL TO USER ROUTINE
                        ENDIF
1959 C9                 RET             ;RETURN
195A 3E0D       CRLF:   MVI     A,0DH   ;LOAD A CR
195C CD4F19             CALL    TESTO   ;WRITE IT
195F 3E0A               MVI     A,0AH   ;LF
1961 CD4F19             CALL    TESTO   ;WRITE IT
1964 3EFF               MVI     A,255   ;GET RUBOUT CHAR
1966 06FA               MVI     B,0FAH  ;LOAD 255-RUBOUT COUNT
1968 CD4F19     PAUZ:   CALL    TESTO   ;SEND RUBOUT
196B 04                 INR     B       ;INCREMENT COUNT
196C B8                 CMP     B       ;COMARE TO 255
196D C26819             JNZ     PAUZ    ;SEND ANOTHER RUBOUT
1970 AF         CRLF2:  XRA     A       ;GET A ZERO
1971 327622             STA     COLUM   ;RESET COLUMN POINTER
1974 C9                 RET             ;RETURN
 975 FE15       NOTCR:  CPI     15H     ;TEST IF CONTROL-U
1977 C28319             JNZ     NOTCO   ;BRIF NOT
197A CD6D1A             CALL    PRCNT   ;GO PRINT #U
197D CD5A19             CALL    CRLF    ;GET CR/LF
1980 C30719             JMP     REIN    ;GO RE-ENTER
1983 FE7F       NOTCO:  CPI     7FH     ;TEST IF RUBOUT
1985 C2A619             JNZ     NOTBS   ;BRIF NOT
1988 3A7120             LDA     TAPES   ;GET PAPER TAPE SW
198B 1F                 RAR             ;TEST IF LOAD
198C DA1D19             JC      TREAD   ;IGNORE IF LOAD
198F 2B                 DCX     H       ;POINT PRIOR
1990 7E                 MOV     A,M     ;LOAD PREV CHAR
1991 B7                 ORA     A       ;TEST IF BEGIN
1992 CAB119             JZ      ECHO    ;BRIF IS
1995 3E5C               MVI     A,'\'   ;BACK SLASH
1997 CD4F19             CALL    TESTO   ;WRITE IT
199A 7E                 MOV     A,M     ;FETCH CHARACTER TO BE DISCARDED
199B CD4F19             CALL    TESTO   ;WRITE IT
199E 3E5C               MVI     A,'\'   ;BACK SLASH
19A0 CD4F19             CALL    TESTO   ;WRITE IT
19A3 C31D19             JMP     TREAD   ;GET REPLACEMENT CHARACTER
19A6 =          NOTBS   EQU     $
                        IF      LARGE   ;CONTROL H WORKS ONLY ON 9K VERSION
                        CPI     8       ;TEST FOR ASCII BACKSPACE
                        JNZ     NOTCH   ;BRIF NOT CONTROL H
                        DCX     H       ;POINT PRIOR
                        MOV     A,M     ;FETCH CHARACTER
                        ORA     A       ;TEST FOR BEGINNING
                        JZ      ECHO    ;BRIF IT IS
                        PUSH    H       ;SAVE POSITION
                        LXI     H,RBOUT ;POINT RUBOUT SEQUENCE
```

```
                        CALL    TERMM   ;WRITE IT
                        POP     H       ;RESTORE H,L
                        JMP     TREAD   ;GET REPLACEMENT CHARACTER
                        ENDIF
19A6 3A7120     NOTCH:  LDA     TAPES   ;GET PAPER TAPE SWITCH
19A9 1F                 RAR             ;FLAG TO CARRY
19AA DA8119             JC      ECHO    ;NO ECHO IF TAPE
19AD 7E                 MOV     A,M     ;ELSE, LOAD THE CHAR
19AE CD4F19             CALL    TESTO   ;ECHO THE CHARACTER
1981 23         ECHO:   INX     H       ;POINT NEXT POSIT
1982 C31D19             JMP     TREAD   ;LOOP FOR NEXT
                ;
1985 =          TERMO   EQU     $
                ;
                ; TTY PRINT ROUTINE
                ;
                ; OUTPUT STRING OF CHARS
                ; STARTING AT IOBUF +0 THRU END (FF OR FE OR 00)
                ; FOLLOWING IMBEDDED CHARACTERS ARE INTERPRETED AS CONTROLS:
                ; X'00' END OF BUFFER, TYPE CR/LF AND RETURN
                ; X'FE' END OF BUFFER, RETURN (NO CR/LF)
                ; X'FD' TYPE CR/LF, CONTINUE
                ;
                ; RETURN WITHOUT OUTPUT IF OUTPUT SW IS OFF
                ;
                ;
1985 3A7320             LDA     OUTSW   ;GET OUTPUT SW
1988 B7                 ORA     A       ;TEST IT
1989 C0                 RNZ             ;RETURN IF NO PRINT
198A 21CE20             LXI     H,IOBUF ;POINT I/O BUFFER
198D 7E         OT1:    MOV     A,M     ;LOAD A BYTE
198E FEFE               CPI     0FEH    ;SEE IF END OF LINE (NO CR/LF)
19C0 C8                 RZ              ;RETURN IF EQUAL
19C1 FEFD               CPI     0FDH    ;SEE IF IMBEDDED CR/LF
19C3 C2CC19             JNZ     OT2     ;BRIF NOT
19C6 CD5A19             CALL    CRLF    ;LINE FEED
19C9 C3D819             JMP     OT4     ;CONTINUE
19CC B7         OT2:    ORA     A       ;TEST IF END OF OUTPUT
19CD CA5A19             JZ      CRLF    ;BRIF IS
19D0 7E                 MOV     A,M     ;LOAD THE BYTE
19D1 CD4F19             CALL    TESTO   ;TYPE IT
19D4 3A7622             LDA     COLUM   ;GET COLUMN POINTER
19D7 3C                 INR     A       ;ADD ONE
19D8 327622             STA     COLUM   ;RESTORE IT
19DB 23         OT4:    INX     H       ;POINT NEXT
19DC C38D19             JMP     OT1     ;LOOP
198D =          TERMM   EQU     OT1
                ;
19DF =          TABST   EQU     $
                ;
                ;
                ; POSITION TTY AT NEXT TAB STOP
                ;
                ;
19DF 3A7320             LDA     OUTSW   ;GET OUTPUT SWITCH
19E2 B7                 ORA     A       ;TEST IT
19E3 C0                 RNZ             ;RETURN IF SUPPRESSED
19E4 3A7622             LDA     COLUM   ;GET COLUMN POINTER
```

```
19E7 FE38              CPI     56      ;COMPARE TO 56
19E9 D25A19            JNC     CRLF    ;BRIF NO ROOM LEFT
19EC 47                MOV     B,A     ;SAVE IT
  ED AF                XRA     A       ;INIT POSITION
19EE B8          TBLP: CMP     B       ;COMPARE
19EF CAF519            JZ      TBLP2
19F2 D2FA19            JNC     TBON    ;BRIF SHY OF TAB
19F5 C60E        TBLP2:ADI     14      ;POINT NEXT STOP
19F7 C3EE19            JMP     TBLP    ;LOOP
19FA 327622      TBON: STA     COLUM   ;UPDATE CTR
19FD 90                SUB     B       ;COMPUTE NUMBER OF SPACES
19FE 47                MOV     B,A     ;SAVE IT
19FF 3E20        TBSPA:MVI     A,' '   ;SPACE TO REG A
1A01 CD4F19            CALL    TESTO   ;OUTPUT IT
1A04 05                DCR     B       ;SUB 1 FROM CTR
1A05 C8                RZ              ;RETURN IF ZERO
1A06 C3FF19            JMP     TBSPA   ;ELSE, LOOP
                ;
1A09 =          LINEO  EQU     $
                ;
                ; UNPACK LINE NUMBER FROM (H,L) TO (D,E)
                ; ZERO SUPPRESS LEADING ZEROS
                ;
                ;
1A09 C5                PUSH    B       ;SAVE B,C
1A0A 0601              MVI     B,1     ;SET SWITCH
1A0C CD141A            CALL    LOUT    ;GO FORMAT 2 BYTES
1A0F CD141A            CALL    LOUT    ;THEN THE NEXT 2
 \12 C1                POP     B       ;RESTORE B,C
.A13 C9                RET             ;RETURN
                ;
1A14 =          LOUT   EQU     $
1A14 7E                MOV     A,M     ;GET BYTE
1A15 E6F0              ANI     0F0H    ;ISOLATE LEFT HALF
1A17 1F                RAR             ;SHIFT RIGHT 1 BIT
1A18 1F                RAR             ;AGAIN
1A19 1F                RAR             ;AGAIN
1A1A 1F                RAR             ;LAST TIME
1A1B C2221A            JNZ     NOTZ1   ;BRIF NOT ZERO
1A1E B0                ORA     B       ;MERGE IN B
1A1F C2281A            JNZ     Z1      ;BRIF ZERO
1A22 0600        NOTZ1:MVI     B,0     ;RESET SWITCH
1A24 F630              ORI     30H     ;ZONE
1A26 12                STAX    D       ;PUT TO BUFFER
1A27 13                INX     D       ;POINT NEXT
1A28 7E          Z1:   MOV     A,M     ;LOAD BYTE
1A29 E60F              ANI     0FH     ;MASK
1A2B C2321A            JNZ     NOTZ2   ;BRIF NOT ZERO
1A2E B0                ORA     B       ;MERGE SWITCH
1A2F C2381A            JNZ     Z2      ;BRIF ZERO
1A32 0600        NOTZ2:MVI     B,0     ;SET SWITCH OFF
1A34 F630              ORI     30H     ;ZONE
1A36 12                STAX    D       ;PUT TO BUFFER
 A37 13                INX     D       ;POINT NEXT
 A38 23          Z2:   INX     H       ;AND NEXT LINE BYTE
1A39 C9                RET             ;RETURN
                ;
1A3A =          TSTCC  EQU     $
```

```
                        ;
                        ; TEST IF KEY WAS PRESSED DURING EXECUTION
                        ; CANCEL IF CONTROL-C
                        ; TOGGLE OUTPUT SUPPRESS SW IF CONTROL-O
                        ;
                                IF      NOT CPM
1A3A DB03                       IN      TTY+1   ;GET TTY STATUS
1A3C E602                       ANI     02H     ;MASK FOR RXRDY
1A3E C8                         RZ              ;RETURN IF NO CHAR
1A3F DB02               GETCH:  IN      TTY     ;READ THE CHAR
1A41 E67F                       ANI     7FH     ;TURN OFF PARITY
                                ENDIF
                                IF      CPM
                                ;NOTE: FOLLOWING CLOBBERS REGISTERS,
                                ; PUSH AND POP IF FOUND TO CREATE BUGS.
                                CALL    BTSTAT  ;CALL BIOS
                                RZ              ;RETURN ON NO CHAR
                        GETCH:  PUSH B          ;SAVE REGS - CPM CAN CLOBBER
                                PUSH D
                                PUSH H
                                CALL    BTIN    ;CALL BIOS TO INPUT
                                POP H           ;RESTORE REGS
                                POP D
                                POP B
                                ENDIF
1A43 FE03                       CPI     03H     ;TEST IF CONTROL C
1A45 C25E1A                     JNZ     TSTC1   ;BRIF NOT
1A48 CD6D1A                     CALL    PRCNT   ;GO PRINT ^C
1A4B 3A7620                     LDA     EDSW    ;GET MODE SW
1A4E B7                         ORA     A       ;TEST IT
1A4F C2DC01                     JNZ     KEY     ;**;BRIF COMMAND MODE
1A52 212D1E                     LXI     H,STOPM ;POINT MSG
1A55 CD8D19                     CALL    TERMM   ;GO PRINT IT
1A58 CDF118                     CALL    PRLIN   ;GO PRINT LINE
1A5B C3DC01                     JMP     KEY     ;GOTO READY
1A5E FE0F               TSTC1:  CPI     0FH     ;TEST IF CONTROL O
1A60 C0                         RNZ             ;RETURN IF NOT
1A61 CD6D1A                     CALL    PRCNT   ;GO PRINT ^O
1A64 3A7320                     LDA     OUTSW   ;GET OUTPUT SWITCH
1A67 EE01                       XRI     1       ;TOGGLE
1A69 327320                     STA     OUTSW   ;PUT SW
1A6C C9                         RET             ;RETURN
                        ;
1A6D =                  PRCNT   EQU     $
                        ;
                        ;
                        ; PRINTS ^ AND CHAR
                        ;
1A6D F5                         PUSH    PSW     ;SAVE CHAR
1A6E 3E5E                       MVI     A,'^'   ;GET UP ARROW
1A70 CD4F19                     CALL    TESTO   ;WRITE IT
1A73 F1                         POP     PSW     ;GET CHAR
1A74 C640                       ADI     64      ;TRNSLATE
1A76 C34F19                     JMP     TESTO   ;WRITE IT
                                ;PAGE
```

```
                        ;
1A79 =                  COMP2   EQU     $
                        ;
                        ; CONTINUATION OF COMPARE (RST 2) ROUTINE
                        ;
1A79 B7                         ORA     A       ;TEST IT
1A7A C2811A                     JNZ     COMP5   ;BRIF NOT END
1A7D AF                 COMP3:  XRA     A       ;SET EQUAL STATUS
1A7E 7E                 COMP4:  MOV     A,M     ;GET LAST CHAR
1A7F C1                         POP     B       ;RESTORE B,C
1A80 C9                         RET             ;RETURN
1A81 BE                 COMP5:  CMP     M       ;COMPARE THE TWO CHARS
1A82 CA8E1A                     JZ      COMP6   ;BRIF EQUAL
1A85 78                         MOV     A,B     ;GET COUNT
1A86 FE03                       CPI     3       ;TEST IF >= 3
1A88 D27D1A                     JNC     COMP3   ;BRIF NOT LESS THAN 3
1A8B C37E1A                     JMP     COMP4   ;BRIF LESS THAN 3 AND NOT EQUAL
1A8E 04                 COMP6:  INR     B       ;COUNT IT
1A8F 13                         INX     D       ;POINT NEXT LIT
1A90 23                         INX     H       ;POINT NEXT VAR
1A91 C31300                     JMP     COMP1   ;CONTINUE
                        ;
1A94 =                  EOL     EQU     $
                        ;
                        ; TESTS IF (H,L) IS END OF LINE
                        ; ERROR-DL IF NOT
                        ;
1A94 CF                         RST     1       ;SKIP TO NON-BLANK
1A95 CDA81A                     CALL    TSTEL   ;TEST IF END LINE
1A98 C20F1C                     JNZ     SNERR   ;ERROR IF NOT
1A9B FE3A                       CPI     ':'     ;TEST FOR MULTIPLE STATEMENT
1A9D C2A31A                     JNZ     EOL1    ;BRIF NOT
1AA0 327422                     STA     MULTI   ;SET SWITCH
1AA3 23                 EOL1:   INX     H       ;POINT NEXT
1AA4 227222                     SHLD    ENDLI   ;SAVE POINTER
1AA7 C9                         RET             ;RETURN
                        ;
1AA8 =                  TSTEL   EQU     $
                        ;
                        ; TEST (H,L) FOR END OF STATEMENT (00H OR ':')
                        ; RETURN WITH Z SET IF IT IS
                        ;
1AA8 B7                         ORA     A       ;TEST FOR ZERO
1AA9 C8                         RZ              ;RETURN IF IS
1AAA FE3A                       CPI     ':'     ;TEST FOR MULTIPLE STATEMENT
1AAC C9                         RET             ;RETURN
                        ;
1AAD =                  NOTEO   EQU     $
                        ;
                        ;
                        ; TEST IF (H,L) IS END OF LINE
                        ; RETURN IF NOT, ERROR-DL IF IS
                        ;
1AAD CF                         RST     1       ;SKIP TO NON-BLANK
1AAE CDA81A                     CALL    TSTEL   ;TEST IF END OF LINE
1AB1 CA0F1C                     JZ      SNERR   ;ERROR IF IS
1AB4 C9                         RET             ;ELSE, RETURN
                        ;
```

```
                        ;
1AB5 =                  PACK    EQU     $
                        ;
                        ; PACK LINE NUMBER FROM (H,L) TO B,C
                        ;
                        ;
1AB5 010000                     LXI     B,0     ;CLEAR B AND C
1AB8 3E04                       MVI     A,4     ;INIT DIGIT COUNTER
1ABA 328D22                     STA     PRSW    ;SAVE A
1ABD 7E                 PK1:    MOV     A,M     ;GET CHAR
1ABE CD2A18                     CALL    NUMER   ;TEST FOR NUMERIC
1AC1 C0                         RNZ             ;RETURN IF NOT NUMERIC
1AC2 E60F                       ANI     0FH     ;STRIP OFF ZONE
1AC4 57                         MOV     D,A     ;SAVE IT
1AC5 3A8D22                     LDA     PRSW    ;GET COUNT
1AC8 3D                         DCR     A       ;SUBTRACT ONE
1AC9 FA0F1C                     JM      SNERR   ;BRIF ERROR
1ACC 328D22                     STA     PRSW    ;SAVE CTR
1ACF 1E04                       MVI     E,4     ;4 BIT SHIFT LOOP
1AD1 79                 PK3:    MOV     A,C     ;GET LOW BYTE
1AD2 17                         RAL             ;ROTATE LEFT 1 BIT
1AD3 4F                         MOV     C,A     ;REPLACE
1AD4 78                         MOV     A,B     ;GET HIGH BYTE
1AD5 17                         RAL             ;ROTATE LEFT 1 BIT
1AD6 47                         MOV     B,A     ;REPLACE
1AD7 1D                         DCR     E       ;DECR CTR
1AD8 C2D11A                     JNZ     PK3     ;LOOP
1ADB 79                         MOV     A,C     ;GET LOW
1ADC B2                         ORA     D       ;PUT DIGIT IN RIGHT HALF OF BYTE
1ADD 4F                         MOV     C,A     ;REPLACE
1ADE 23                         INX     H       ;POINT NEXT BYTE
1ADF C38D1A                     JMP     PK1     ;LOOP
                        ;
1AE2 =                  SQUIS   EQU     $
                        ;
                        ; COMPRESSES THE EXPR STACK
                        ; REG A CONTAINS # OF BYTES TO REMOVE STARTING AT (H,L+1)
                        ; CONTAINS TOTAL NUBER OF CHARACTERS IN STACK THUS FAR
                        ;
1AE2 E5                         PUSH    H       ;SAVE H,L
1AE3 5F                         MOV     E,A     ;COUNT TO E
1AE4 1600                       MVI     D,0     ;ZERO HI BYTE
1AE6 19                         DAD     D       ;COMPUTE START
1AE7 E8                         XCHG            ;PUT TO D,E
1AE8 E1                         POP     H       ;GET H,L
1AE9 2F                         CMA             ;COMPLEMENT COUNT
1AEA 3C                         INR     A       ;THEN 2'S COMPLEMENT
1AEB 80                         ADD     B       ;COMPUTE B-A
1AEC 47                         MOV     B,A     ;PUT TO B
1AED 13                 SQUI2:  INX     D       ;POINT NEXT SEND
1AEE 23                         INX     H       ;POINT NEXT RECEIVE
1AEF 1A                         LDAX    D       ;GET A CHAR
1AF0 77                         MOV     M,A     ;PUT IT DOWN
1AF1 05                         DCR     B       ;DECR CTR
1AF2 C2ED1A                     JNZ     SQUI2   ;LOOP
1AF5 225022                     SHLD    EXPRS   ;UPDATE NEW START OF EXPR
1AF8 C9                         RET             ;RETURN
                        ;
```

```
1AF9 =          SKP2Z   EQU     $
                ;
                ; FIND END OF LITERAL IN (D,E)
                ;
1AF9 1A                 LDAX    D       ;GET BYTE OF LIT
1AFA B7                 ORA     A       ;TEST IT
1AFB C8                 RZ              ;RETURN IF ZERO (END)
1AFC 13                 INX     D       ;ELSE, POINT NEXT
1AFD C3F91A             JMP     SKP2Z   ;LOOP
                ;
1B00 =          GTEMP   EQU     $
                ;
                ; GETS FOUR BYTE TEMPORARY STORAGE AREA,
                ; STORES THE FACC THERE,
                ; PUTS ADDR OF AREA IN EXPR STACK (H,L)
                ;
1B00 EB                 XCHG            ;SAVE H,L IN D,E
1B01 E3                 XTHL            ;EXCHANGE 0 AND RET ADDR
1B02 E5                 PUSH    H       ;PUT NEW RET ADDR
1B03 E5                 PUSH    H       ;DOIT IT AGAIN
1B04 210000             LXI     H,0     ;ZERO H,L
1B07 39                 DAD     SP      ;GET SP ADDR IN H,L
1B08 23                 INX     H       ;PLUS ONE
1B09 23                 INX     H       ;PLUS ONE MORE (POINT TO NEW AREA)
1B0A C5                 PUSH    B       ;SAVE CTRS
1B0B D5                 PUSH    D       ;SAVE EXPR ADDR
1B0C E5                 PUSH    H       ;SAVE TEMP ADDR
1B0D DF                 RST     3       ;GO STORE FACC
1B0E D1                 POP     D       ;RESTORE TEMP ADDR
1B0F 2A6922             LHLD    SPCTR   ;GET COUNT
1B12 23                 INX     H       ;PLUS ONE
1B13 23                 INX     H       ;ONE MORE
1B14 226922             SHLD    SPCTR   ;PUT BACK
1B17 E1                 POP     H       ;RESTORE EXPR ADDR
1B18 C1                 POP     B       ;RESTORE CTRS
1B19 23         SADR:   INX     H       ;POINT NEXT BYTE
1B1A 72                 MOV     M,D     ;HIGH BYTE TO EXPRSTK
1B1B 23                 INX     H       ;POINT NEXT
1B1C 73                 MOV     M,E     ;LOW BYTE TO EXPR STK
1B1D 23                 INX     H       ;POINT NEXT
1B1E 36E3               MVI     M,0E3H  ;CODE = NUMERIC DATA
1B20 C9                 RET             ;RETURN
                ;
1B21 =          ALPHA   EQU     $
                ;
                ; TESTS THE CHAR AT (H,L)
                ; RETURNS WITH Z SET IF CHAR IS ALPHA (A-Z)
                ; RETURNS WITH Z OFF IF NOT ALPHA
                ; CHAR IS LEFT IN REG A
                ;
1B21 7E                 MOV     A,M     ;PUT CHAR TO REG A
1B22 FE41               CPI     'A'     ;TEST IF A OR HIGHER
1B24 D8                 RC              ;RETURN IF NOT ALPHA (Z IS OFF)
1B25 FE5A               CPI     'Z'     ;TEST IF Z OR LESS
1B27 C33301B            JMP     NUMEN   ;GO WRAPUP
                ;
1B2A =          NUMER   EQU     $
                ;
```

```
                ; TESTS THE CHAR AT (H,L)
                ; RETURNS WITH Z SET IF NUMERIC (0-9)
                ; ELSE Z IS OFF
                ; CHAR IS LEFT IN THE A REG
                ;
1B2A 7E                 MOV     A,M     ;GET CHAR TO REG A
1B2B FE30               CPI     '0'     ;TEST IF ZERO OR GREATER
1B2D D8                 RC              ;RETURN IF LESS THAN ZERO
1B2E FE39               CPI     '9'     ;TEST IF 9 OR LESS
1B30 C8         NUMEN:  RZ              ;RETURN IF 9
1B31 D0                 RNC             ;RETURN IF NOT NUMERIC
1B32 BF                 CMP     A       ;SET Z
1B33 C9                 RET             ;RETURN
                ;
1834 =          SEARC   EQU     $
                ;
                ; SEARCHES FOR THE VARIABLE IN D,E
                ; RETURNS WITH ADDR OF DATA AREA FOR VARIABLE
                ;
1834 E5                 PUSH    H       ;SAVE H,L
1835 3A8822             LDA     FNMOD   ;GET FUNCTION MODE
1838 B7                 ORA     A       ;TEST IT
1839 C28F18             JNZ     SCH6    ;BRIF IN A FUNCTION
183C 2A9122     SCH0:   LHLD    DATAB   ;GET ADDR OF DATA POOL
183F 7E         SCH1:   MOV     A,M     ;GET THE BYTE
1840 B7                 ORA     A       ;TEST IF END
1841 CA651B             JZ      SCH3    ;BRIF END
1844 2B                 DCX     H       ;POINT NEXT
1845 2B                 DCX     H       ;DITTO
1846 46                 MOV     B,M     ;GET HI LEN
1847 2B                 DCX     H       ;POINT NEXT
1848 4E                 MOV     C,M     ;GET LO LEN
1849 E7                 RST     4       ;ADJUST H,L
184A 03                 DB      3
184B 7E                 MOV     A,M     ;LOAD 1ST CHAR
184C BA                 CMP     D       ;COMPARE 1ST CHAR
184D C2611B             JNZ     SCH2    ;BRIF NOT EQUAL
1850 2B                 DCX     H       ;POINT NEXT
1851 7E                 MOV     A,M     ;LOAD 2ND DIGIT
1852 23                 INX     H       ;POINT BACK
1853 BB                 CMP     E       ;COMPARE 2ND CHAR
1854 C2611B             JNZ     SCH2    ;BRIF NOT EQUAL
1857 7A                 MOV     A,D     ;GET HI NAME
1858 B7                 ORA     A       ;TEST IT
1859 FAC418             JM      SCH9    ;RETURN IF MATRIX
185C 09                 DAD     B       ;POINT NEXT ENTRY
185D 23                 INX     H       ;PLUS ONE
185E EB                 XCHG            ;FLIP/FLOP
185F E1                 POP     H       ;RESTORE H
1860 C9                 RET             ;RETURN
1861 09         SCH2:   DAD     B       ;MINUS LEN
1862 C33F18             JMP     SCH1    ;LOOP
1865 72         SCH3:   MOV     M,D     ;PUT 1ST CHAR
1866 2B                 DCX     H       ;POINT NEXT
1867 73                 MOV     M,E     ;PUT 2ND CHAR
1868 2B                 DCX     H       ;POINT NEXT
1869 7A                 MOV     A,D     ;GET HI NAME
186A B7                 ORA     A       ;TEST IT
```

```
1868 FAA318            JM      SCH7     ;BRIF ARRAY
186E 36FF              MVI     M,0FFH   ;HI LEN
1870 2B                DCX     H        ;POINT NEXT
1871 7B                MOV     A,E      ;GET LO NAME
1872 B7                ORA     A        ;TEST TYPE
1873 FA7D18            JM      SCH4     ;BRIF CHAR
1876 36F8              MVI     M,0F8H   ;LO LEN
1878 0604              MVI     B,4      ;LOOP CTR
187A C38118            JMP     SCH5     ;BRARND
187D 36F8      SCH4:   MVI     M,0FBH   ;LO LEN
187F 0601              MVI     B,1      ;LOOP CTR
1881 2B        SCH5:   DCX     H        ;POINT NEXT
1882 3600              MVI     M,0      ;ZERO THE VALUE
1884 05                DCR     B        ;DECR CTR
1885 C28118            JNZ     SCH5     ;LOOP
1888 2B                DCX     H        ;POINT NEXT
1889 3600              MVI     M,0      ;MARK NEW END
188B 23                INX     H        ;POINT ADDR OF VARIABLE
188C EB                XCHG             ;PUT LOCATION TO D,E
188D E1                POP     H        ;RESTORE H,L
188E C9                RET              ;RETURN
188F 216C22    SCH6:   LXI     H,FNARG  ;POINT DUMMY ARG
1892 7E                MOV     A,M      ;LOAD 1ST CHAR
1893 BA                CMP     D        ;COMPARE
1894 C23C18            JNZ     SCH0     ;BRIF NOT EQUAL
1897 23                INX     H        ;POINT NEXT
1898 7E                MOV     A,M      ;LOAD 2ND CHAR
1899 BB                CMP     E        ;COMPARE
189A C23C18            JNZ     SCH0     ;BRIF NOT EQUAL
189D 23                INX     H        ;POINT NEXT
189E 56                MOV     D,M      ;GET HI ADDR
189F 23                INX     H        ;POINT NEXT
18A0 5E                MOV     E,M      ;GET LO ADDR
18A1 E1                POP     H        ;RESTORE H,L
18A2 C9                RET              ;RETURN
18A3 E5        SCH7:   PUSH    H        ;SAVE ADDRESS
18A4 36FE              MVI     M,0FEH   ;MOVE HI DISP
18A6 2B                DCX     H        ;POINT NEXT
18A7 3614              MVI     M,14H    ;MOVE LO DISP
18A9 2B                DCX     H
18AA 3600              MVI     M,0      ;MOVE A ZERO
18AC 2B                DCX     H        ;POINT NEXT
18AD 360A              MVI     M,10     ;MOVE 10
18AF 2B                DCX     H        ;POINT NEXT
18B0 3600              MVI     M,0      ;MOVE A ZERO
18B2 2B                DCX     H        ;POINT NEXT
18B3 360A              MVI     M,10     ;MOVE A 10 (DEFAULT IS 10 X 10)
18B5 01E501            LXI     B,485    ;TOTAL # OF BYTES TAKEN BY ARRAY
18B8 2B        SCH8:   DCX     H        ;POINT NEXT
18B9 3600              MVI     M,0      ;CLEAR ONE BYTE
18BB 0B                DCX     B        ;DCR CTR
18BC 78                MOV     A,B      ;GET HI
18BD B1                ORA     C        ;PLUS LO
18BE C2B818            JNZ     SCH8     ;LOOP
18C1 E1                POP     H        ;RESTORE PTR TO START
18C2 23                INX     H        ;POINT LO NAME
18C3 23                INX     H        ;POINT HI NAME
18C4 C1        SCH9:   POP     B        ;NEED TO XCHANGE LAST 2 STACK ENTRIES
```

```
18C5 D1                 POP     D       ;SO DOIT
18C6 C5                 PUSH    B
18C7 D5                 PUSH    D
18C8 C9                 RET             ;RETURN
                ;
18C9 =          VAR     EQU     $
                ;
                ;
                ; TEST (H,L) FOR A VARIABLE NAME
                ; PUTS THE NAME IN D,E IF FOUND
                ; ERROR SN IF NONE FOUND
                ;
18C9 CF                 RST     1       ;SKIP TO NON-BLANK
18CA CD2118             CALL    ALPHA   ;TEST IF ALPHA
18CD C20F1C             JNZ     SNERR   ;BRIF NOT ALPHA
18D0 57                 MOV     D,A     ;FIRST CHAR
18D1 1E20               MVI     E,' '   ;DEFAULT
18D3 23                 INX     H       ;POINT NEXT
18D4 CF                 RST     1       ;GET 2ND CHAR
18D5 CD2A18             CALL    NUMER   ;TEST IF NUMERIC
18D8 C2DE18             JNZ     VAR2    ;BRIF NOT NUMERIC
18DB 5F                 MOV     E,A     ;SAVE 2ND CHAR
18DC 23                 INX     H       ;POINT NEXT
18DD CF                 RST     1       ;GET NON-BLANK FOLLOWING
18DE FE24       VAR2:   CPI     '$'     ;TEST IF STRING
18E0 C2E918             JNZ     VAR3    ;BRIF NOT
18E3 78                 MOV     A,E     ;GET 2ND CHAR
18E4 F680               ORI     80H     ;SET TYPE
18E6 5F                 MOV     E,A     ;SAVE IT
18E7 23                 INX     H       ;SKIP $
18E8 C9                 RET             ;THEN RETURN
18E9 FE28       VAR3:   CPI     '('     ;TEST IF ARRAY
18EB C0                 RNZ             ;RETURN IF NOT
18EC 7A                 MOV     A,D     ;GET HI NAME
18ED F680               ORI     80H     ;TURN ON D7
18EF 57                 MOV     D,A     ;RESTORE
18F0 C9                 RET             ;RETURN
                ;
18F1 =          PRLIN   EQU     $
                ;
                ; PRINTS LINE NUMBER FOLLOWED BY CR/LF
                ;
18F1 117720             LXI     D,LINEN ;POINT AREA
18F4 2A8922             LHLD    LINE    ;GET ADDR OF LINE NUMBER
18F7 CD091A             CALL    LINEO   ;GO UNPACK
18FA E8                 XCHG            ;PUT TO H,L
18FB 3600               MVI     M,0     ;END OF MSG
18FD 217720             LXI     H,LINEN ;POINT AREA
1C00 C3BD19             JMP     TERMM   ;GO PRINT IT
                        ;PAGE
```

```
                        ;
                        ; ERROR MESSAGE ROUTINES
                        ; FATAL ERROR MUST BE FIRST

00FE =                  EM      EQU     0FEH


1C03 F7         ULERR:  RST     6
1C04 554CFEF7           DB      'UL',EM,FATAL   ;NOTE FATAL = CODE FOR RST 6
1C07 =          ZMERR   EQU     $-1             ;LOG(X<=0),SQR(-X),0 DIVIDE
1C08 4F46FEF7           DB      'OF',EM,FATAL
1C0B =          STERR   EQU     $-1             ;ERROR IN EXPRESSION STACK
1C0C 5354FEF7           DB      'ST',EM,FATAL
1C0F =          SNERR   EQU     $-1             ;DELIMITER ERROR
1C10 534EFEF7           DB      'SN',EM,FATAL
1C13 =          RTERR   EQU     $-1             ;RETURN & NO GOSUB
1C14 5254FEF7           DB      'RT',EM,FATAL
1C17 =          DAERR   EQU     $-1             ;OUT OF DATA
1C18 4441FEF7           DB      'DA',EM,FATAL
1C1B =          NXERR   EQU     $-1             ;NEXT & NO FOR / >8 FOR'S
1C1C 4E58FEF7           DB      'NX',EM,FATAL
1C1F =          CVERR   EQU     $-1             ;CONVERSION ERROR
1C20 4356FEF7           DB      'CV',EM,FATAL
1C23 =          CKERR   EQU     $-1             ;CHECKSUM ERROR
1C24 4348FEF7           DB      'CK',EM,FATAL

                        ; NON-FATAL ERRORS

1C27 =          OVERR   EQU     $-1             ;OVERFLOW ERROR
1C28 4F56FE             DB      'OV',EM
1C2B C9                 RET                     ;RETURN TO ROUTINE
1C2C F7         UNERR:  RST     6               ;CALL   ERROR ROUTINE
1C2D 554EFE             DB      'UN',EM
1C30 C9                 RET

                        ; CONTINUATION OF ERROR MESSAGE ROUTINE (RST 6)

1C31 CD8D19     ERROR:  CALL    TERMM   ;PRINT 'XX'
1C34 E5                 PUSH    H       ;SAVE RETURN
1C35 213C1E             LXI     H,ERRMS ;PRINT 'ERROR IN LINE'
1C38 CD8D19             CALL    TERMM
1C3B CDF118             CALL    PRLIN   ;PRINT LINE #
1C3E E1                 POP     H
1C3F 23                 INX     H       ;RETURN ADDRESS
1C40 7E                 MOV     A,M     ;GET INSTRUCTION
1C41 FEF7               CPI     FATAL   ;IS IT AN RST 6?
1C43 CADC01             JZ      KEY     ;IF ZERO, YES, ABORT
1C46 C1                 POP     B       ;RESTORE REGISTERS
1C47 D1                 POP     D
1C48 F1                 POP     PSW
1C49 E3                 XTHL
1C4A C9                 RET
                        ;PAGE
```

```
                        ;
                        ;
                        ; MOVE THE STRING FROM (D,E) TO (H,L) COUNT IN B
                        ;
1C4B 0604       CPY4D:  MVI     B,4
1C4D 1A         COPYD:  LDAX    D       ;GET A BYTE
1C4E 77                 MOV     M,A     ;MOVE IT
1C4F 23                 INX     H       ;POINT NEXT
1C50 13                 INX     D       ;DITTO
1C51 05                 DCR     B       ;DECR CTR
1C52 C24D1C             JNZ     COPYD   ;LOOP
1C55 C9                 RET             ;THEN RETURN
                        ;
                        ;
                        ; MOVE THE STRING FROM (H,L) TO (D,E) COUNT IN B
                        ;
1C56 0604       CPY4H:  MVI     B,4
1C58 EB         COPYH:  XCHG            ;FLIP/FLOP
1C59 CD4D1C             CALL    COPYD   ;GO COPY
1C5C EB                 XCHG            ;FLIP/FLOP BACK
1C5D C9                 RET             ;RETURN
                        ;
1C5E =          ZEROM   EQU     $
                        ;
                        ; MOVES A STRING OF BINARY ZEROS, COUNT IN B
                        ;
1C5E 3600               MVI     M,0     ;MOVE A ZERO
1C60 23                 INX     H       ;POINT NEXT
1C61 05                 DCR     B       ;DECR CTR
1C62 C25E1C             JNZ     ZEROM   ;LOOP
1C65 C9                 RET             ;RETURN
                        ;
1C66 =          FBIN    EQU     $
                        ;
                        ;
                        ; CONVERT FLOAT ACC TO UNSIGNED BINARY NUMBER IN A REG
                        ; RETURNS 0 IN A REG IF FACC<0 OR FACC>255
                        ;
                        ;
1C66 E5                 PUSH    H       ;SAVE H,L
1C67 D5                 PUSH    D       ;SAVE D,E
1C68 CD351F             CALL    FACDE   ;CONVERT FACC TO D,E
1C6B AF                 XRA     A       ;ZERO A
1C6C B2                 ORA     D       ;TEST HIGH VALUE
1C6D C2711C             JNZ     FBIN1   ;BRIF NOT ZERO
1C70 7B                 MOV     A,E     ;VALUE TO A
1C71 D1         FBIN1:  POP     D       ;RESTORE D,E
1C72 E1                 POP     H       ;RESTORE H,L
1C73 C9                 RET             ;RETURN
                        ;
1C74 =          ARG     EQU     $
                        ;
                        ; GET NEXT ARGUMENT FROM POLISH STACK
                        ;
1C74 2A5222             LHLD    ADDR1   ;GET ADDRESS
1C77 23                 INX     H       ;POINT NEXT
1C78 56                 MOV     D,M     ;GET HI ADDRESS
1C79 23                 INX     H       ;POINT NEXT
```

```
1C7A 5E                 MOV     E,M     ;GET LO ADDRESS
1C7B 23                 INX     H       ;POINT TYPE
1C7C 225222             SHLD    ADDR1   ;GET ADDRESS
 7F 2B                  DCX     H       ;POINT BACK
 80 C38313              JMP     EVLD    ;CALL EVLOAD AND RETURN
                ;
                ;
1C83 =          ARGNU   EQU     $
                ;
1C83 CD741C             CALL    ARG     ;GET ARGUMENT
1C86 C3661C             JMP     FBIN    ;THEN CONVERT FACC TO BIN
                ;
1C89 =          BINFL   EQU     $
                ;
                ; CONVERT D,E TO FLOATING POINT NUMBER IN FAC
                ;
                ;
1C89 215822             LXI     H,FACC  ;POINT ACC
1C8C 3618               MVI     M,24    ;MAX BITS
1C8E 23                 INX     H       ;POINT NEXT
1C8F 3600               MVI     M,0     ;CLEAR MSB
1C91 23                 INX     H       ;POINT NEXT
1C92 72                 MOV     M,D     ;MOVE MID
1C93 23                 INX     H       ;POINT NEXT
1C94 73                 MOV     M,E     ;MOVE LSB
1C95 C3DD16             JMP     FNORM   ;GO NORMALIZE & RETURN
                        ;PAGE
```

```
                        ;
                        ; FUNCTION TABLE. FORMAT IS:
                        ;       DB <LITERAL>,0
                        ;       DW <ADDRESS>
                        ;       DB <FUNCTION TYPE>
                        ;
                        ; TABLE IS TERMINATED WITH A '00'
                        ;
1C98 =                  FUNCT   EQU     $
1C98 41425300                   DB      'ABS',0
1C9C C708                       DW      ABS
1C9E AB                         DB      0ABH
1C9F 53515200                   DB      'SQR',0
1CA3 270C                       DW      SQR
1CA5 AB                         DB      0ABH
1CA6 494E5400                   DB      'INT',0
1CAA E208                       DW      INT
1CAC AB                         DB      0ABH
1CAD 53474E00                   DB      'SGN',0
1CB1 D00B                       DW      SGN
1CB3 AB                         DB      0ABH
1CB4 524E4400           RNDLI:  DB      'RND',0
1CB8 840C                       DW      RND
1CBA AB                         DB      0ABH
1CBB 53494E00                   DB      'SIN',0
1CBF 410A                       DW      SIN
1CC1 AB                         DB      0ABH
1CC2 434F5300                   DB      'COS',0
1CC6 830A                       DW      COS
1CC8 AB                         DB      0ABH
1CC9 54414E00                   DB      'TAN',0
1CCD 8C0A                       DW      TAN
1CCF AB                         DB      0ABH
1CD0 41544E00                   DB      'ATN',0
1CD4 D40A                       DW      ATN
1CD6 AB                         DB      0ABH
1CD7 494E5000                   DB      'INP',0
1CD8 0A0D                       DW      INP
1CDD AB                         DB      0ABH
1CDE 4C4E00                     DB      'LN',0
1CE1 1308                       DW      LN
1CE3 AB                         DB      0ABH
1CE4 4C4F4700                   DB      'LOG',0
1CE8 6108                       DW      LOG
1CEA AB                         DB      0ABH
1CEB 45585000                   DB      'EXP',0
1CEF 6A08                       DW      EXP
1CF1 AB                         DB      0ABH
1CF2 504F5300                   DB      'POS',0
1CF6 200D                       DW      POS
1CF8 AB                         DB      0ABH
1CF9 4C454E00                   DB      'LEN',0
1CFD 890D                       DW      LENFN
1CFF AB                         DB      0ABH
1D00 4348522400                 DB      'CHR$',0
1D05 8F0D                       DW      CHRFN
1D07 CB                         DB      0CBH
1D08 4153434949                 DB      'ASCII',0
```

```
1D0E 9A0D               DW      ASCII
1D10 AB                 DB      0ABH
1D11 4E554D2400         DB      'NUM$',0
1D16 A70D               DW      NUMFN
1D18 C8                 DB      0C8H
1D19 56414C00           DB      'VAL',0
1D1D 8A0D               DW      VAL
1D1F AB                 DB      0ABH
1D20 5350414345         DB      'SPACE$',0
1D27 E10D               DW      SPACE
1D29 C8                 DB      0C8H
1D2A 535452494E         DB      'STRING$',0
1D32 F10D               DW      STRFN
1D34 D3                 DB      0D3H
1D35 4C45465424         DB      'LEFT$',0
1D3B 050E               DW      LEFT
1D3D D3                 DB      0D3H
1D3E 5249474854         DB      'RIGHT$',0
1D45 0E0E               DW      RIGHT
1D47 D3                 DB      0D3H
1D48 4D49442400         DB      'MID$',0
1D4D 170E               DW      MIDFN
1D4F D8                 DB      0D8H
1D50 494E535452         DB      'INSTR',0
1D56 510E               DW      INSTR
1D58 88                 DB      088H
1D59 5045454800         DB      'PEEK',0
1D5E A81F               DW      PEEK
1D60 AB                 DB      0ABH
                        IF      LARGE
                        DB      0,0,0,0 ;ROOM FOR ONE MORE FUNCTION
                        DB      0,0,0,0
                        ENDIF
1D61 00                 DB      0       ;END OF FUNCTION TABLE
                        ;PAGE
```

```
                        ;
                        ; PROGRAM CONSTANTS
                        ;
1D62 131400     PCHOF:  DB      19,20,0
1D65 3FFD       RNDP:   DB      3FH,0FDH        ;16381
1D67 3FEB               DB      3FH,0EBH        ;16363
1D69 3FDD               DB      3FH,0DDH        ;16349
1D6B 18EC       NRNDX:  DB      18H,0ECH
1D6D 33D3               DB      33H,0D3H
1D6F 1A85               DB      1AH,85H
1D71 281E               DB      28H,1EH
1D73 5748415400WHATL:   DB      'WHAT',0
1D78 =          VERS    EQU     $       ;VERSION MESSAGE
                        IF      LARGE
                        DB      '9K VERS 1.4',0
                RBOUT:  DB      08H,20H,08H,0FEH ;RUBOUT SEQUENCE (9K ONLY)
                        ENDIF
                        IF      NOT LARGE
1D78 3848205645         DB      '8K VERS 1.4',0
                        ENDIF
1D84 4C494E4500LLINE:   DB      'LINE',0
1D89 54414200   TABLI:  DB      'TAB',0
1D8D 5354455000STEPL:   DB      'STEP',0
1D92 5448454E00THENL:   DB      'THEN',0
1D97 504900     PILIT:  DB      'PI',0
1D9A 02800000   TWO:    DB      02H,80H,00H,00H     ;CONSTANT:  2
1D9E 04A00000   TEN:    DB      04H,0A0H,00H,00H    ;CONSTANT:  10
1DA2 02C90FD7   PI:     DB      02H,0C9H,0FH,0D7H   ;CONSTANT:  3.141593
1DA6 00C90FD7   QTRPI:  DB      00H,0C9H,0FH,0D7H   ;CONSTANT:  0.7853892
1DAA 80FFFFFF   NEGON:  DB      80H,0FFH,0FFH,0FFH  ;CONSTANT: -0.9999999
1DAE 00817216   LN2C:   DB      00H,0B1H,72H,16H    ;CONSTANT:  0.6931472
1DB2 009714E8   SQC1:   DB      00H,97H,14H,0E8H    ;CONSTANT:  0.59016206
1DB6 7FD5A956   SQC2:   DB      7FH,0D5H,0A9H,56H   ;CONSTANT:  0.41730759
                        ;PAGE
```

```
                ; THE FOLLOWING CONSTANTS MUST BE IN THIS ORDER  ************

                ;       CONSTANT WITH EXPONENT OF 1
                ;       COEFFICIENT OF FIRST TERM
                ;       ...
                ;       COEFFICIENT OF NTH TERM

                ; SINCE ALL COEFFICIENTS ARE LESS THAN 1,
                ; THE ITERATION LOOP USES THE
                ; CONSTANT WITH EXPONENT 1 TO TERMINATE THE EVALUATION.

1D8A 01B504F3   SQC3:   DB      01H,0B5H,04H,0F3H       ;CONSTANT:  1.41421356
1D8E FFAA95BC           DB      0FFH,0AAH,95H,0BCH      ;CONSTANT: -0.3331738
1DC2 7ECAD520           DB      7EH,0CAH,0D5H,20H       ;CONSTANT:  0.1980787
1DC6 FE8782D6           DB      0FEH,87H,82H,0D6H       ;CONSTANT: -0.1323351
1DCA 7DA3131C           DB      7DH,0A3H,13H,1CH        ;CONSTANT:  0.07962632
1DCE FC89A688           DB      0FCH,89H,0A6H,088H      ;CONSTANT: -0.03360627
1DD2 79DF3A9E   ATNCO:  DB      79H,0DFH,3AH,9EH        ;CONSTANT:  0.006812411

1DD6 01C90FD7   HALFP:  DB      01H,0C9H,0FH,0D7H       ;CONSTANT:  1.570796
1DDA 80A55DDE           DB      80H,0A5H,5DH,0DEH       ;CONSTANT: -0.64596371
1DDE 7DA33455           DB      7DH,0A3H,34H,55H        ;CONSTANT:  0.079689679
1DE2 F9993860           DB      0F9H,99H,38H,60H        ;CONSTANT: -0.0046737656
1DE6 749ED786   SINCO:  DB      74H,9EH,0D7H,086H       ;CONSTANT:  0.0001514841

1DEA 0180       ONE:    DB      001H,080H
1DEC 0000       NULLI:  DB      00H,00H                 ;CONSTANT:  1.0
1DEE 00FFFEC1           DB      00H,0FFH,0FEH,0C1H      ;CONSTANT:  0.99998103
 DF2 FFFF8A80           DB      0FFH,0FFH,08AH,080H     ;CONSTANT: -0.4994712
 DF6 7FA80E28           DB      7FH,0A8H,0EH,28H        ;CONSTANT:  0.3282331
1DFA FEE74855           DB      0FEH,0E7H,48H,55H       ;CONSTANT: -0.2258733
1DFE 7E89DEE3           DB      7EH,89H,0DEH,0E3H       ;CONSTANT:  0.134693
1E02 FCE1C578           DB      0FCH,0E1H,0C5H,078H     ;CONSTANT: -0.05511996
1E06 7A803FAE   LNCO:   DB      7AH,080H,3FH,0AEH       ;CONSTANT:  0.01075737

1E0A 01B8AA3B   LN2E:   DB      001H,0B8H,0AAH,03BH     ;CONSTANT:  1.44269504
1E0E 00B15FE6           DB      000H,081H,05FH,0E6H     ;C=.69311397
1E12 7EF62F70           DB      07EH,0F6H,02FH,070H     ;C=.24041548
1E16 7CE1C2AE           DB      07CH,0E1H,0C2H,0AEH     ;C=.05511732
1E1A 7AA0B87E           DB      07AH,0A0H,0B8H,07EH     ;C=.00981033
1E1E 77CA09C8   EXPCO:  DB      077H,0CAH,009H,0C8H     ;C=.00154143


1E22 7FDE5BD0   LNC:    DB      07FH,0DEH,05BH,0D0H       ;C=LOG BASE 10 OF E
1E26 =          READY   EQU     $
1E26 FD                 DB      0FDH
1E27 5245414459         DB      'READY',0
1E2D =          STOPM   EQU     $
1E2D FD                 DB      0FDH
1E2E 53544F5020         DB      'STOP AT LINE ',254
1E3C 204552524FERRMS:   DB      ' ERROR IN LINE ',0FEH
0002 =          TTY     EQU     2
                        ;PAGE
```

```
                    ;
                    ; VERB (STATEMENT/COMMAND) TABLE
                    ; FORMAT IS: DB 'VERB',0
                    ;            DW ADDR
                    ;            DB 'NEXT VERB',0
                    ;            ETC
                    ;   END OF TABLE IS MARKED BY DB 0
                    ;
1E4C =              JMPTB   EQU     $
1E4C 4C49535400             DB      'LIST',0
1E51 6202                   DW      LIST
1E53 52554E00               DB      'RUN',0
1E57 F401                   DW      RUNCM
1E59 58455100       XEQL:   DB      'XEQ',0
1E5D F901                   DW      XEQ
1E5F 4E455700       NEWL:   DB      'NEW',0
1E63 8801                   DW      NEW
1E65 434F4E00               DB      'CON',0
1E69 EE02                   DW      CONTI
1E6B 5441504500             DB      'TAPE',0
1E70 BE01                   DW      TAPE
1E72 5341564500             DB      'SAVE',0
1E77 5502                   DW      SAVE
1E79 48455900       KEYL:   DB      'KEY',0
1E7D DC01                   DW      KEY
1E7F 46524500               DB      'FRE',0
1E83 A001                   DW      FREE
1E85 494600                 DB      'IF',0
1E88 E904                   DW      IFSTM
1E8A 5245414400             DB      'READ',0
1E8F E107                   DW      READ
1E91 524553544F             DB      'RESTORE',0
1E99 1603                   DW      RESTO
1E9B 4441544100     DATAL:  DB      'DATA',0
1EA0 0802                   DW      RUN
1EA2 464F5200               DB      'FOR',0
1EA6 E503                   DW      FOR
1EA8 4E45585400     NEXTL:  DB      'NEXT',0
1EAD 9206                   DW      NEXT
1EAF 474F535542     GOSBL:  DB      'GOSUB',0
1EB5 3A03                   DW      GOSUB
1EB7 5245545552             DB      'RETURN',0
1EBE 2203                   DW      RETUR
1EC0 494E505554             DB      'INPUT',0
1EC6 2107                   DW      INPUT
1EC8 5052494E54             DB      'PRINT',0
1ECE 5503                   DW      PRINT
1ED0 474F           GOTOL:  DB      'GO'
1ED2 544F00         TOLIT:  DB      'TO',0
1ED5 F602                   DW      GOTO
1ED7 4C455400               DB      'LET',0
1ED8 F105                   DW      LET
1EDD 53544F5000             DB      'STOP',0
1EE2 7208                   DW      STOP
1EE4 454E4400       ENDL:   DB      'END',0
1EE8 C801                   DW      ENDIT
1EEA 52454D00               DB      'REM',0
1EEE 0802                   DW      RUN
```

```
1EF0 2100                  DB      '!',0
1EF2 0802                  DW      RUN
   4 3F00                  DB      '?',0
   6 5503                  DW      PRINT
1EF8 52414E444F            DB      'RANDOMIZE',0
1F02 9F08                  DW      RANDO
1F04 4F4E00                DB      'ON',0
1F07 8508                  DW      ON
1F09 4F555400              DB      'OUT',0
1F0D 4A08                  DW      OUTP
1F0F 44494D00              DB      'DIM',0
1F13 8109                  DW      DIM
1F15 4348414E47            DB      'CHANGE',0
1F1C 2A09                  DW      CHANG
1F1E 444546      DEFLI:    DB      'DEF'
1F21 464E00      FNLIT:    DB      'FN',0
1F24 0802                  DW      RUN
                           IF      CPM
                           DB      'DDT',0
                           DW      DDT
                           DB      'BYE',0
                           DW      BOOT
                           ENDIF
1F26 504F484500            DB      'POKE',0
1F28 861F                  DW      POKE
1F2D 43414C4C00            DB      'CALL',0
1F32 D41F                  DW      JUMP
                           IF      LARGE   ;INCLUDE ONLY IN 8K+ VERSION
                           DB      'EDIT',0
                           DW      FIX
                           DB      'CLOAD',0
                           DW      CLOAD
                           DB      'CSAVE',0
                           DW      CSAVE
                           ENDIF
                           IF      HUNTER
                           DB      'BAUD',0
                           DW      BAUD
                           ENDIF
1F34 00                    DB      0       ;END OF TABLE
                ;

                ; DDT COMMAND, CPM ONLY

                           IF      CPM
                DDT:       RST     7
                           JMP     RDY
                           ENDIF
                           ;PAGE
```

```
                        ;
1F35 =                  FACDE   EQU     $
                        ;
                        ; THIS ROUTINE CONVERTS THE FACC TO AN ADDRESS IN D,E
                        ;
1F35 CDE20B                     CALL    INT     ;INTEGERIZE THE FACC
1F38 3A5822                     LDA     FACC    ;GET THE EXPONENT
1F3B B7                         ORA     A       ;TEST IT
1F3C FA271C                     JM      OVERR   ;BRIF NEGATIVE ADDRESS
1F3F D610                       SUI     16      ;SUBTRACT MAX EXPONENT
1F41 CA571F                     JZ      FDE2    ;BRIF EQUAL MAX
1F44 F2271C                     JP      OVERR   ;BRIF GREATER THAN 64K
1F47 2F                         CMA             ;2'S COMPLIMENT OF A YIELDS..
1F48 3C                         INR     A       ;16-A
1F49 4F                         MOV     C,A     ;SAVE SHIFT COUNT
1F4A AF                 FDE1:   XRA     A       ;CLEAR CARRY
1F4B 215922                     LXI     H,FACC+1        ;POINT MANTISSA
1F4E 0602                       MVI     B,2     ;WORDS TO SHIFT
1F50 CDF818                     CALL    FSHFT   ;GO SHIFT FACC+1 AND FACC+2
1F53 0D                         DCR     C       ;REDUCE COUNT
1F54 C24A1F                     JNZ     FDE1    ;LOOP TILL COMPLETE
1F57 215922             FDE2:   LXI     H,FACC+1        ;POINT HIGH BYTE
1F5A 56                         MOV     D,M     ;LOAD D
1F5B 23                         INX     H       ;POINT LOW BYTE
1F5C 5E                         MOV     E,M     ;LOAD E
1F5D C9                         RET             ;RETURN
                        ;
                        ;
1F5E =                  LOCAT   EQU     $
                        ;
                        ; THIS ROUTINE SEARCHES FOR A LINE IN THE PROGRAM FILE.
                        ; Z SET, C RESET==>LINE FOUND. ADDRESS IS IN H,L
                        ; C SET, Z RESET==>NOT FOUND. H,L POINT TO NEXT LINE
                        ; C SET, Z SET==>NOT FOUND. H,L POINT AT END OF PROGRAM
                        ;
1F5E 219622                     LXI     H,BEGPR ;POINT START
1F61 7E                 FIND1:  MOV     A,M     ;FETCH LENGTH OF LINE
1F62 E5                         PUSH    H       ;SAVE POINTER
1F63 B7                         ORA     A       ;TEST
1F64 CA831F                     JZ      FIND3   ;BRIF END
1F67 23                         INX     H       ;POINT LINE #
1F68 7E                         MOV     A,M     ;FETCH HI #
1F69 B8                         CMP     B       ;COMPARE TO REQUESTED
1F6A DA781F                     JC      FIND2   ;BRIF LOW
1F6D C2831F                     JNZ     FIND3   ;BRIF PAST AND NOT FOUND
1F70 23                         INX     H       ;POINT LO #
1F71 7E                         MOV     A,M     ;FETCH IT
1F72 B9                         CMP     C       ;COMPARE TO REQUESTED
1F73 DA781F                     JC      FIND2   ;BRIF LOW
1F76 C2831F                     JNZ     FIND3   ;BRIF PAST AND NOT FOUND
1F79 E1                         POP     H       ;POINT BEGIN IF MATCH
1F7A C9                         RET             ;RETURN
                        ;
                        ; BUMP H,L TO NEXT LINE
                        ;
1F7B E1                 FIND2:  POP     H       ;POINT START OF LINE
1F7C 5E                         MOV     E,M     ;LENGHT TO E
1F7D 1600                       MVI     D,0     ;CLEAR D
```

```
1F7F 19                 DAD     D       ;BUMP H,L
1F80 C3611F             JMP     FIND1   ;CONTINUE
                ;
                ; LINE NOT FOUND
                ;
1F83 37         FIND3:  STC             ;SET CARRY
1F84 E1                 POP     H       ;POINT LINE JUST PAST REQUESTED
1F85 C9                 RET             ;RETURN
                ;
                ;
1F86 =          SEEK    EQU     $
                ;
                ;  THIS CODE FINDS AN ENTRY IN THE TABLE POINTED TO BY D,E.
                ;  THE SOUGHT ENTRY IS POINTED TO BY H,L.
                ;
1F86 E5         SEEK1:  PUSH    H       ;SAVE ADDRESS OF STRING
1F87 1A                 LDAX    D       ;GET BYTE FROM TABLE
1F88 B7                 ORA     A       ;TEST IT
1F89 CAA91F             JZ      SEEK3   ;BRIF END OF TABLE
1F8C D7                 RST     2       ;COMPARE
1F8D C2991F             JNZ     SEEK2   ;BRIF NOT FOUND
1F90 E3                 XTHL            ;PUT CURRENT H,L ON STACK
1F91 CDF91A             CALL    SKP2Z   ;FIND END TO LITERAL IN TABLE
1F94 13                 INX     D       ;POINT LOW BYTE
1F95 E1                 POP     H       ;RESTORE LINE POINTER
1F96 3C                 INR     A       ;PUT 1 IN A
1F97 B7                 ORA     A       ;RESET Z BIT
1F98 C9                 RET             ;RETURN
1F99 CDF91A     SEEK2:  CALL    SKP2Z   ;FIND END OF TABLE LITERAL
1F9C 13                 INX     D       ;
1F9D 13                 INX     D       ;POINT NEXT LIT IN TABLE
1F9E 13                 INX     D       ;
1F9F E1                 POP     H       ;GET ORGINAL STRING
1FA0 1A                 LDAX    D       ;GET BYTE
1FA1 17                 RAL             ;HIGH BIT TO CARRY
1FA2 D2861F             JNC     SEEK1   ;NOT A FUNCTION SEARCH
1FA5 13                 INX     D       ;POINT NEXT BYTE IN FUNCTION TABLE
1FA6 C3861F             JMP     SEEK1   ;CONTINUE SEARCH
1FA9 E1         SEEK3:  POP     H       ;RESTORE ORGINAL STRING
1FAA C9                 RET             ;RETURN
                        IF      LARGE   ;ASSEMBLE THE REMAINDAR ONLY FOR 8+K
                ;
                ;
                ; EDIT COMMAND
                ; EDIT <LINE #><DELIMITER><OLD TEXT><DELIMITER><NEW TEXT>
                ;
                FIX     EQU     $
                        RST     1       ;SKIP BLANKS
                        CALL    PACK    ;GET LINE # IN B,C
                        RST     1       ;SKIP BLANKS
                        SHLD    ADDR2   ;SAVE COMMAND POINTER
                        CALL    LOCAT   ;SEARCH FOR LINE # IN PROGRAM
                        JC      ULERR   ;BRIF NOT FOUND
                        PUSH    H       ;SAVE ADDR OF EXISTING LINE <SOURCE>
                        PUSH    B       ;SAVE LINE #
                        MOV     B,M     ;GET LENGTH OF <SOURCE>
                        XCHG            ;D,E POINT <SOURCE>
                        LXI     H,STRIN ;POINT STRING BUFFER
```

```
        CALL    COPYD   ;<SOURCE> TO STRING BUFFER
        LDA     STRIN   ;LENGTH OF <SOURCE> TO A
        SUI     2       ;ADJUST
        STA     STRIN   ;STORE
        LXI     D,IOBUF+1       ;POINT BUFFER
        LHLD    ADDR2   ;FETCH COMMAND POINTER
        MOV     B,M     ;FETCH <DELIMITER>
;
; FIND LENGTH OF <OLD TEXT>. STORE IT IN IOBUF.
;
        MVI     C,0     ;INITIAL LENGTH
FIX1:   INX     H       ;POINT NEXT CHARACTER
        MOV     A,M     ;FETCH
        ORA     A       ;TEST
        JZ      SNERR   ;MISSING 2ND <DELIMITER>.
        CMP     B       ;TEST
        JZ      FIX2    ;BRIF IF 2ND <DELIMITER> FOUND
        INR     C       ;ELSE, BUMP C
        STAX    D       ;STORE CHARACTER IN IOBUF
        INX     D       ;BUMP IOBUF POINTER
        JMP     FIX1    ;CONTINUE
;
; GET READY TO SEARCH <SOURCE> FOR <OLD TEXT>
;
FIX2:   MOV     A,C     ;LENGTH OF <OT> TO A
        STA     IOBUF   ;STORE
        SHLD    ADDR2   ;SAVE COMMAND POINTER
        MVI     A,3     ;SEARCH WILL START IN POS 3.
        LHLD    PROGE   ;POINT END OF PROGRAM
        INX     H       ;BUMP TWICE
        INX     H
        SHLD    ADDR1   ;SAVE EXPR. STACK POINTER
        INX     H       ;POINT NEXT
        LXI     D,IOBUF ;POINT BUFFER AREA
        MOV     M,D     ;STORE ADDRESS
        INX     H
        MOV     M,E
        LXI     H,STRIN ; POINT <SOURCE>
;
; USE THE INSTR ROUTINE TO SEARCH
;
        CALL    INST2   ;GO SEARCH
        MOV     A,E     ;RESULT TO A
        ORA     A       ;TEST
        JZ      DAERR   ;BR IF NOT FOUND
        MOV     C,A     ;SAVE POSITION IN C
        DCR     A       ;ADJUST
        MOV     B,A     ;COPY TO B
        LXI     H,STRIN+1       ;POINT <OLD SOURCE>
        LXI     D,IOBUF+1       ;POINT <NEW LINE AREA>
        CALL    COPYH   ;COPY <OLD SOURCE> UP TO <OLD TEXT>
        PUSH    D       ;SAVE DEST POINTER
;
; SKIP OVER <OLD TEXT> IN <SOURCE>
;
        MVI     D,0     ;CLEAR D
        LDA     IOBUF   ;GET LENGTH OF <OT>
        MOV     E,A     ;LENGTH TO E
```

```
        DAD     D       ;BUMP H,L PAST <OT>
        POP     D       ;RESTORE <DEST> POINTER
        PUSH    H       ;SAVE <REMAINING SOURCE> POINTER
;
; APPEND <NEW TEXT> TO <DEST>
;
        LHLD    ADDR2   ;FETCH COMMAND POINTER
FIX3:   INX     H       ;POINT NEXT
        MOV     A,M     ;FETCH CHARACTER
        ORA     A       ;TEST IT
        JZ      FIX4    ;BRIF NO MORE <NEW TEXT>
        INR     C       ;BUMP LENGTH COUNT
        STAX    D       ;STORE CHARACTER
        INX     D       ;BUMP <DEST> POINTER
        JMP     FIX3    ;CONTINUE
;
; APPEND <REMAINING SOURCE> TO <DEST>
;
FIX4:   POP     H       ;GET REMAINING SOURCE POINTER
FIX4A:  MOV     A,M     ;FETCH CHARACTER
        ORA     A       ;TEST
        JZ      FIX5    ;BRIF DONE
        STAX    D       ;STORE CHARACTER
        INR     C       ;BUMP CHAR COUNT
        INX     D       ;BUMP DEST POINTER
        INX     H       ;BUMP <SOURCE> POINTER
        JMP     FIX4A   ;CONTINUE
;
; PREPARE <DEST> FOR SUBMISSION AS NEW LINE
;
FIX5:   STAX    D       ;BUFFER TERMINATOR
        INR     C       ;BUMP LENGTH COUNT
        MOV     A,C     ;FETCH COUNT
        STA     IOBUF   ;STORE IT
        MOV     B,A     ;COPY COUNT TO B
        LXI     H,IMMED ;POINT NEW LINE AREA
        LXI     D,IOBUF ;POINT WHERE IT IS NOW
        CALL    COPYD   ;COPY IT
        POP     B       ;RESTORE LINE #
        POP     H       ;RESTORE PROGRAM POINTER
        PUSH    H       ;SAVE IT
        JMP     EDIT2   ;PROCESS AS NEW LINE
;
```

```
;
; TAPE CASSETTE COMMANDS
;

;       TAPE CASSETTE EQUATES

SWCH    EQU     0FFH    ;SWITCH PORT
CASC    EQU     3       ;STATUS PORT FOR TARBELL
CASD    EQU     0       ;DATA PORT
CFLAG   EQU     4       ;DATA FLAG FOR TARBELL ON MIO

; CASSETTE FILE FORMAT

;    EACH RECORD:
;       TYPE BYTE: 4 FOR BASIC PROGRAM,
;                  PLUS BIT 7 ON IF DATA NOT HEADER RECORD
;       LENGTH BYTE: # DATA BYTES (1-128)
;       2 BYTES OF CHECKSUM

;    EACH FILE BEGINS WITH A HEADER RECORD
;       TYPE: 4
;       LENGTH: 7
;           5 CHARS FILENAME, BLANK-FILLED
;           2 BYTES TOTAL LENGTH OF DATA IN FILE
;       2 BYTES OF CHECKSUM

;    AND HAS N DATA RECORDS
;       TYPE: 84
;       LENGTH: 128 EXCEPT LAST RECORD MAY BE LESS
;       DATA: NEXT (LENGTH) BYTES OF IMAGE OF PROGRAM AREA
;       CHECKSUM: 2 BYTES, 2'S COMPLEMENT OF SUM OF BYTES

;    FILES OF TYPE OTHER THAN 4 ARE IGNORED BY BASIC

; HARDWARE USED:
;       IMSAI MIO BOARD, CASSETTE DATA ON PORT 0,
;       STATUS ON PORT 3,
;       CASSETTE READY JUMPERED TO BIT 2 OF PORT 3.


;       TAPE UTILITY ROUTINE

; WATCH         WAIT FOR TARBELL READY OR CONTROL-C

WATCH:  PUSH B          ;SAVE REGS - CPM STATUS CALL CAN CLOBE
        PUSH D
        PUSH H
        CALL    TSTCC   ;TEST FOR CNTRL-C
        POP H           ;RESTORE REGS IN CPM DEBUGING MODE
        POP D
        POP B
        IN      CASC    ;READ STATUS PORT
        ANI     CFLAG   ;TEST
        JZ      WATCH   ;LOOP TILL REbAADY
        RET


; CASI          CASSETTE INPUT TO A-REGISTER
```

```
CASI:   CALL    WATCH   ;WAIT TIL READY
        IN      CASD    ;READ FROM DATA PORT
        RET


; RECO          WRITE A RECORD TO THE TARBELL.
;               D,E==>TYPE, LENGTH BYTES
;               H,L==>START OF SOURCE
;               RETURNS UPDATED SOURCE POINTER IN DE

RECO:   MOV     A,D     ;TYPE BYTE
        CALL    CASO    ;WRITE IT
        MOV     A,E     ;COUNT
        CALL    CASO    ;WRITE IT
        MOV     B,E     ;COUNT
        XCHG            ;SOURCE NOW IN DE
        LXI     H,0     ;INITIAL CHACKSUM
NCHAR:  LDAX    D       ;FETCH NEXT CHAR
        CALL    CASO    ;WRITE IT
        INX     D       ;PNT NEXT CHAR
        CALL    CKSUM   ;ADD TO CKSUM, PUT ADD IN LIGHTS
        DCR     B       ;REDUCE COUNT
        JNZ     NCHAR   ;LOOP ON COUNT
        DCX     H       ;ADJUST HL FOR COMPLIMENT
        MOV     A,H     ;WRITE CHECKSUM
        CMA
        CALL    CASO
        MOV     A,L
        CMA
        ;WRITE LAST BYTE & RETRUN


; CASO          CASSETTE OUTPUT BYTE FROM A-REGISTER

CASO:   PUSH PSW
        CALL WATCH      ;WAIT TILL READY
        POP PSW
        OUT CASD        ;WRITE TO DATA PORT
        RET


; CKSUM         CALCULATE THE CHECKSUM:
;               ADD A TO HL
;       ALSO OUTPUTS HI ADDR TO SENSE LIGHTS

CKSUM:  ADD     L       ;ADD PREVIOUS LO
        MOV     L,A     ;SAVE NEW LO
        RNC
        INR     H       ;PROPAGATE CARRY


; SENSE         OUTPUT HI ADDR FROM D TO LIGHTS

SENSE:  MOV     A,D
        CMA
        OUT     SWCH
        RET
```

```
; RECI          INPUT A RECORD FROM THE TARBELL
;       TAKES BUFFER POINTER IN HL
;       RETURNS UPDATED POINTER IN DE,
;               RECORD TYPE IN A, RECORD LENGTH IN C
;               CLOBBERS B,H,L

RECI:   CALL    CASI    ;GET TYPE
        PUSH    PSW     ;SAVE TYPE TO RETURN TO CALLER
        CALL    CASI    ;GET LENGTH
        MOV     C,A     ;STORE LEN
        MOV     B,A     ;IN B ALSO
        XCHG            ;PUT DESTINATION PTR IN DE
        LXI     H,0     ;INITIAL CHECKSUM
RECI1:  CALL    CASI    ;INPUT BYTE
        STAX    D       ;STORE IT
        INX     D
        CALL    CKSUM   ;UPDATE CKSUM, PUT ADDR IN LIGHTS
        DCR     B       ;LOOP ON COUNT
        JNZ     RECI1
        PUSH    D       ;SAVE DESTINATION PTR
        CALL    CASI    ;INPUT CHECKSUM
        MOV     D,A
        CALL    CASI
        MOV     E,A
        DAD     D       ;COMPARE
        MOV     A,H
        ORA     L
        JNZ     CKERR   ;BRIF CHECKSUM ERROR
        POP     D       ;RESTORE DEST PTR
        POP     PSW     ;RESTORE RECORD TYPE BYTE
        RET


; CSAVE COMMAND

CSAVE:  RST     1       ;SKIP ANY SPACES
        MVI     A,10H   ;ENABLE WRITE
        OUT     CASC
        PUSH    H       ;SAVE PTR
        MVI     B,255   ;WRITE INITAL 255 NULLS
        XRA     A
NULS:   CALL    CASO
        DCR     B
        JNZ     NULS
        MVI     A,3CH   ;START BYTE
        CALL    CASO
        MVI     B,32    ;32 SYNC BYTES
        MVI     A,0E6H  ;SYNC BYTE VALUE
SYNCS:  CALL    CASO
        DCR     B
        JNZ     SYNCS
        LXI     H,IOBUF ;POINT BUFFER
        MVI     B,5     ;FILE NAME LENGTH
        POP     D       ;RESTORE CMD PTR
FNAME:  MVI     M,20H   ;DEFAULT BLANK
        LDAX    D       ;FETCH FILE NAME
```

```
        ORA     A       ;TEST
        JZ      BLANK
        MOV     M,A     ;STORE CHAR
        INX     D       ;NAME PTR
BLANK:  INX     H       ;BUFFER PTR
        DCR     B       ;COUNT
        JNZ     FNAME

; CALCULATE LGTH OF PROGRAM FILE&WRITE IT ON THE HEADER

        LXI     D,BEGPR ;BEGINNING OF PROGRAM
        LHLD    PROGE   ;END
        MOV     A,L
        SUB     E
        MOV     L,A
        MOV     A,H
        SBB     D
        MOV     H,A
        INX     H       ;PLUS 1 TO GET # OF BYTES INCLUSIVE
        PUSH    H       ;SAVE FOR LATER
        SHLD    IOBUF+5 ;STUFF LENGTH
        LXI     D,407H  ;TYPE AND LEN OF HEADER RECORD
                        ;TYPE 4: BASIC PROG FILE, HEADER RCD
        LXI     H,IOBUF
        CALL    RECO    ;WRITE RECORD

; WRITE PROGRAM FILE

        LXI     H,BEGPR ;POINT START OF PROGRAM
NXTRC:  XTHL            ;GET REMAINING LENGTH
        MOV     A,H     ;GET HI REMAINING
        ORA     L       ;TEST FOR DONE
        JZ      ERITE   ;BRIF IF DONE
        LXI     D,0FF80H;-128
        DAD     D       ;SUBTRACT RECORD LENGTH
        JC      RITE    ;IF CARRY, NOT AT END
        MOV     A,L     ;GET LOW
        ANI     7FH     ;NUMBER BYTES LEFT
        MOV     E,A     ;COUNT
        LXI     H,0     ;REMAINING BYTES
RITE:   XTHL            ;RESTORE H
        MVI     D,084H  ;TYPE BYTE: 80=DATA RECORD (NOT
                        ;FILE HDR), 4=BASIC PROGRAM FILE.
        CALL    RECO    ;WRITE
        XCHG            ;SAVE SOURCE PTR
        JMP     NXTRC
ERITE:  POP     H       ;CLEAN STACK


; BELL          RING USER'S CHIMES

BELL:   MVI     A,7     ;CODE FOR BELL
        CALL    TESTO
        JMP     RDY


        ;PAGE
```

```
; CLOAD          LOAD A PROGRAM FROM THE TARBELL

CLOAD:
NULL1:  MVI     A,60H    ;MIO CONTROL TO READ BY BITS
        OUT     CASC     ;WRITE TO STATUS PORT
NULLS:  CALL    CASI     ;READ LEADING NULLS
        OUT     SWCH     ;PUT IN LIGHTS
        CPI     0E6H     ;WAIT FOR FIRST SYNC BYTE
        JNZ     NULLS
        MVI     A,20H    ;MIO CONTROL TO READ BY BYTES
        OUT     CASC     ;WRITE TO STATUS PORT
        MVI     B,31     ;NUMBER REMAINING SYNC BYTES
SYNC:   CALL    CASI     ;READ PAST SYNC
        OUT     SWCH
        CPI     0E6H
        JNZ     NULL1    ;TRY FOR MORE NULLS
        DCR     B
        JNZ     SYNC
        LXI     H,IOBUF  ;POINT BUFFER
        CALL    RECI     ;READ A RECORD
        CPI     4        ;TEST TYPE BYTE: IS IT BASIC PROGRAM
                         ;..FILE HEADER RECORD?
        JNZ     NULL1    ;NO, START OVER, KEEP LOOKING
        LHLD    IOBUF+5  ;LOAD LENGTH OF PROGRAM FILE
        PUSH    H        ;SAVE
        LXI     H,BEGPR
NXTR:   CALL    RECI     ;READ RECORD
        CPI     84H      ;IS IT BASIC PROG FILE DATA RECOF
        JNZ     CKERR    ;NO, SOMETHING'S WRONG.
        POP     H        ;LENGTH
        ;SUBTRACT 0,C  FROM HL
        MOV     A,L
        SUB     C
        MOV     L,A
        MOV     A,H
        MVI     C,0
        SBB     C
        MOV     H,A
        ORA     L        ;TEST RESULT FOR 0
        XCHG             ;BUFFER ADDR TO HL
        PUSH    D        ;SAVE REMAINING LENGTH
        JNZ     NXTR     ;JIF NOT DONE READING DATA
        POP     D        ;CLEAR STACK
;LOADING DONE. SET POINTER TO END OF PROGRAM.
        XRA     A
        MOV     M,A      ;EXTRA 0 FOR PARANOISA
        DCX     H        ;POINT LAST LOADED BYTE (SHOULD BE 0)
        SHLD    PROGE    ;SAVE END OF PROG FOR EDIT, LIST, &C
        STA     IOBUF+5  ;MARK END OF FILE NAME FOR TYPEOUT
;TYPE FILE NAME
        LDA     IOBUF
        CPI     20H      ;TEST FOR NO NAME
        CNZ     TERMO    ;PRINT NAME IF THERE
        JMP     BELL
        ENDIF
;
```

```
                        ;
1FA8 =                  PEEK    EQU     $
                        ;
                        ; STMT: A=PEEK(X). RETURNS DECIMAL VALUE OF MEMORY ADDRESS X.
                        ;
1FA8 CD351F             CALL    FACDE   ;GET ADDRESS IN D,E
1FAE EB                 XCHG            ;ADDRESS TO H,L
1FAF 110000             LXI     D,0     ;CLEAR D,E
1FB2 5E                 MOV     E,M     ;PUT MEMORY BYTE IN E
1FB3 C3891C             JMP     BINFL   ;CONVERT D,E TO BINARY AND RETURN
                        ;
1FB6 =                  POKE    EQU     $
                        ;
                        ; STMT: POKE <ADDRESS>,<VALUE>.  PUTS VALUE IN MEMORY ADDRESS.
                        ;
1FB6 CD800F             CALL    EXPR    ;EVALUATE ADDRESS EXPRESSION
1FB9 7E                 MOV     A,M     ;LOAD NEXT CHARACTER
1FBA FE2C               CPI     ','     ;TEST
1FBC C20F1C             JNZ     SNERR   ;BRIF ERROR
1FBF 23                 INX     H       ;POINT NEXT
1FC0 E5                 PUSH    H       ;SAVE H,L
1FC1 CD351F             CALL    FACDE   ;PUT ADDRESS IN D,E
1FC4 E1                 POP     H       ;RESTORE H,L
1FC5 D5                 PUSH    D       ;SAVE ADDRESS
1FC6 CD800F             CALL    EXPR    ;EVALUATE VALUE EXPRESSION
1FC9 CD941A             CALL    EOL     ;TEST FOR END OF LINE
1FCC CD661C             CALL    FBIN    ;CONVERT FACC TO A REGISTER VALUE
1FCF E1                 POP     H       ;GET D,E ADDRESS IN H,L
1FD0 77                 MOV     M,A     ;MOVE BYTE
1FD1 C30B02             JMP     RUN     ;CONTINUE
                        ;
                        ;
1FD4 =                  JUMP    EQU     $
                        ;
                        ; STMT: CALL <ADDRESS>. EXECUTES CODE AT MEMORY ADDRESS.
                        ;
1FD4 CD800F             CALL    EXPR    ;EVALUATE ADDRESS EXPRESSION
1FD7 CD941A             CALL    EOL     ;TEST FOR END OF LINE
1FDA CD351F             CALL    FACDE   ;CONVERT FACC TO ADDRESS IN D,E
1FDD 210B02             LXI     H,RUN   ;MAKE INTO SUBROUTINE
1FE0 E5                 PUSH    H
1FE1 E8                 XCHG            ;MOVE ADDRESS TO HL
1FE2 E9                 PCHL            ;EXECUTE USER'S ROUTINE
                        ;
```

```
        IF HUNTER
;
;
BAUD    EQU     $
;
; SOFTWARE BAUD SELECTION ON SIO BOARDS MODIFIED BY
; W. HARTER, COYOTE COMPUTERS, DAVIS, CALIF.
;
; COMMAND 'BAUD <RATE>' WHERE <RATE>=110,300,1200,2400,9600
;
        RST     1       ;SKIP BLANKS
        LXI     D,BAUDS+6       ;POINT BAUD TABLE
        CALL    SEEK    ;GO SEARCH BAUD TABLE
        JZ      CVERR   ;BRIF RATE NOT FOUND
        DCX     H       ;ADJUST POINTER
BAUD1:  INX     H       ;LOOK AT CHARACTER
        CALL    NUMER   ;TEST FOR DIGIT
        JZ      BAUD1   ;LOOP PAST RATE
        CALL    EOL     ;TEST FOR END OF LINE
        XCHG            ;POINT ADDRESS OF CONTROL BYTES
        MOV     E,M     ;LOW BYTE TO E
        INX     H       ;POINT NEXT
        MOV     D,M     ;HIGH BYTE TO D
        LDA     EDSW    ;GET MODE SWITCH
        ORA     A       ;TEST IT
        JNZ     SETIT   ;BRIF IMMEDIATE MODE
        LXI     H,BAUDS ;POINT 'BAUD'
        CALL    TERMM   ;WRITE IT
        PUSH    D       ;SAVE ADDRESS OF CONTROL BYTES
        LXI     H,IOBUF ;POINT BUFFER
        MVI     B,4     ;LOAD COUNT
        CALL    COPYD   ;COPY RATE TO IOBUF
        MVI     M,0     ;TERMINATE MESSAGE
        CALL    TERMO   ;WRITE IT
        POP     D       ;RESTORE CONTROL BYTES
SETIT:  LXI     H,4     ;LOAD OFFSET
        DAD     D       ;POINT 1ST CONTROL BYTE
        MVI     A,40H   ;LOAD RESET
        OUT     TTY+1   ;WRITE IT
        MOV     A,M     ;MODE BYTE
        OUT     TTY+1   ;WRITE IT
        MVI     A,17H   ;ENABLE BYTE
        OUT     TTY+1   ;WRITE IT
        INX     H       ;POINT SPEED BYTE
        MOV     A,M     ;LOAD IT
        OUT     8       ;WRITE IT
BAUD2:  IN      TTY+1   ;READ STATUS
        ANI     2       ;TEST
        JZ      BAUD2   ;WAIT FOR ACKNOWLEDGMENT
        IN      TTY     ;READ AND DISCARD
        LDA     EDSW    ;GET MODE SWITCH
        ORA     A       ;TEST IT
        JZ      RUN     ;BRIF RUN MODE
        JMP     GETCM   ;BRIF IMMEDIATE MODE
BAUDS:  DB      'BAUD ',0FEH    ;BAUD MESSAGE
;
; BAUD TABLE.
;
```

```
B110:   DB      '110 ',0FAH,2,0
        DW      B110
B300:   DB      '300 ',0FBH,0
        DW      B300
B1200:  DB      '1200',0FAH,0
        DW      B1200
B2400:  DB      '2400',0FAH,32,0
        DW      B2400
B9600:  DB      '9600',0FAH,34,0
        DW      B9600
        DB      0       ;END OF BAUD TABLE
;
        ENDIF

        IF      CPM     ;CPM INITIALIZATION STORES
                        ;... BIOS JUMP TABLE HERE
BTSTAT: DS      3       ;JMP TO BIOS CONSOLE STATUS
BTIN:   DS      3       ;JMP TO BIOS CONSOLE INPUT
BTOUT:  DS      3       ;JMP TO BIOS CONSOLE OUTPUT
        ENDIF
;
```

```
1FE2 =          ROMEN   EQU     $-1
                ;
2000                    ORG     8192    ;RAM STARTS OF 8K BOUNDARY
                        IF      LARGE OR CPM    ;ADJUST START OF RAM IF 8+K
                        ORG     2400H   ;RAM STARTS ON 9K BOUNDARY
                        ENDIF
                ;
                ; ALL CODE ABOVE THIS POINT IS READ ONLY AND CAN BE PROM'ED
                ;
                ;
2000 =          RAM     EQU     $

2000 =          BZERO   EQU     $
2000            FORNE:  DS      1       ;# ENTRYS IN TABLE (MUST BE HERE)
2001                    DS      112     ;ROOM FOR 8 NESTS (MUST BE HERE)
2071            TAPES:  DS      1       ;TAPE SWITCH (MUST BE HERE)
2072            DIMSW:  DS      1       ;DIM SWITCH (MUST BE HERE)
2073            OUTSW:  DS      1       ;OUTPUT SWITCH (MUST BE HERE)
2074            ILSW:   DS      1       ;INPUT LINE SWITCH (MUST BE HERE)
2075            RUNSW:  DS      1       ;RUN SWITCH(MUST BE HERE)
2076            EDSW:   DS      1       ;MODE SWITCH(MUST BE HERE)
2077 =          EZERO   EQU     $

2077            LINEN:  DS      5
207C            IMMED:  DS      82      ;IMMEDIATE COMMAND STORAGE AREA
20CE            IOBUF:  DS      82      ;INPUT/OUTPUT BUFFER
2120            STRIN:  DS      256     ;STRING BUFFER AREA
2220            OUTA:   DS      3       ;*** FILLED IN AT RUN TIME
2223            INDX:   DS      2       ;HOLDS VARIABLE NAME OF FOR/NEXT
2225            REL:    DS      1       ;HOLDS THE RELATION IN AN IF STMT
2226            IFTYP:  DS      1       ;HOLDS TYPE CODE OF LEFT SIDE
2227            TVAR1:  DS      4       ;TEMP STORAGE
222B            TVAR2:  DS      4       ;DITTO
222F            TEMP1:  DS      4       ;TEMP STORAGE FOR FUNCTIONS
2233            TEMP2:  DS      4
2237            TEMP3:  DS      4
223B            TEMP4:  DS      4
223F            TEMP5:  DS      4
2243            TEMP6:  DS      4
2247            TEMP7:  DS      4
224B            LINEL:  DS      2       ;HOLDS MIN LINE NUMBER IN LIST
224D            LINEH:  DS      2       ;HOLDS MAX LINE NUMBER IN LIST
224F            PROMP:  DS      1       ;HOLDS PROMPT CHAR
2250            EXPRS:  DS      2       ;HOLDS ADDR OF EXPRESSION
2252            ADDR1:  DS      2       ;HOLDS TEMP ADDRESS
2254            ADDR2:  DS      2       ;HOLDS TEMP ADDRESS
2256            ADDR3:  DS      2       ;HOLDS STMT ADD DURING EXPR EVAL
2258            FACC:   DS      4
225C            FTEMP:  DS      12
2268            PARCT:  DS      1
2269            SPCTR:  DS      2
226B            CMACT:  DS      1       ;COUNT OF COMMAS
226C            FNARG:  DS      4       ;SYMBOLIC ARG & ADDRESS
2270            STMT:   DS      2       ;HOLDS ADDR OF CURRENT STATEMENT
2272            ENDLI:  DS      2       ;HOLDS ADDR OF MULTI STMT PTR
2274            MULTI:  DS      1       ;SWITCH 0=NO, 1=MULTI STMT LINE
2275            DEXP:   DS      1
2276            COLUM:  DS      1       ;CURRENT TTY COLUMN
```

```
2277            RNDX:   DS      2       ;RANDOM VARIABLE STORAGE
2279            RNDY:   DS      2       ;THE RND<X>,TRND<X>,AND RNDSW
227B            RNDZ:   DS      2       ;MUST BE KEPT IN ORDER
227D            RNDS:   DS      2
227F            TRNDX:  DS      2
2281            TRNDY:  DS      2
2283            TRNDZ:  DS      2
2285            TRNDS:  DS      2
2287            RNDSW:  DS      1
2288            FNMOD:  DS      1       ;SWITCH, 0=NOT, <>0 = IN DEF FN
2289            LINE:   DS      2       ;HOLD ADD OF PREV LINE NUM
228B            STACK:  DS      2       ;HOLDS ADDR OF START OF RETURN STACK
228D            PRSW:   DS      1       ;ON=PRINT ENDED WITH , OR ;
228E            NS:     DS      1       ;HOLDS LAST TYPE (NUMERIC/STRING)
228F            DATAP:  DS      2       ;ADDRESS OF CURRENT DATA STMT
2291            DATAB:  DS      2       ;ADDRESS OF DATA POOL
2293            PROGE:  DS      2       ;ADDRESS OF PROGRAM END
                ;
                        IF      CPM
                ; TEMPORARY CODE FOR INITIALIZATION HERE

                INITC:  LHLD    BOOT+1  ;PTR TO BIOS TABLE
                        LXI     D,CSTAT ;OFFSET OF CONSOLE QUERY ENTRY
                        DAD     D       ;POINT INTO BIO JUMP TABLE
                        LXI     D,BTSTAT;POINT INTO BASIC JMP TABLE
                        MVI     B,9     ;COUNT
                        CALL    COPYH   ;MOE BIOS TABLE INTO BASIC
                        MVI     A,0C3H  ;JMP OP CODE
                        LXI     H,RST1! STA 8H! SHLD 9H
                        LXI     H,RST2! STA 10H! SHLD 11H
                        LXI     H,RST3! STA 18H! SHLD 19H
                        LXI     H,RST4! STA 20H! SHLD 21H
                        LXI     H,RST5! STA 28H! SHLD 29H
                        LXI     H,RST6! STA 30H! SHLD 31H
                        LHLD    BDOS+1  ;LOCATE TOP OF RAM
                        JMP     INIT1   ;CONTINUE AS IN NON-CPM VERSION
                        ENDIF


2295                    DS      1       ;DATA STATEMENT FLAG (MUST BE HERE)
                BEGPR:

2296                    END
```