Self-Contained System

Instructions

And Listing

Assembler With Monitor and Editor For
8080 Microcomputer systems

Distributed free of charge by Exatron
Stringy Floppy Owners' Association—ESFOA

```
0000                    ORG     00H
0000 C34000             JMP     INIT    ;DEAD START
0003 C35700             JMP     INITA   ;RESTART AND ENTER MONITOR
                ;
0008                    ORG     08H
0008 C3510D             JMP     BRKP    ;BREAKPOINT RESTART
                ;
0040                    ORG     40H
                ;
                ; THIS ROUTINE INITIALIZES IEE FILE AREA FOR SUBSEQUENT
                ; PROCESSING
                ;
0040 212410     INIT:   LXI     H FILE0
0043 0E4E               MVI     C MAXFIL*FELEN
0045 AF                 XRA     A
0046 77         INIT2:  MOV     M A
0047 23                 INX     H
0048 0D                 DCR     C
0049 C24600             JNZ     INIT2
                ;
                ; CLEAR THE BREAKPOINT TABLE
                ;
004C 0618               MVI     B,NBP*3
004E 210C10             LXI     H BRT
0051 77         INIT3:  MOV     M A
0052 23                 INX     H
0053 05                 DCR     B
0054 C25100             JNZ     INIT3
```

0057 thru 0066: Sixteen ØØ MOP's.

This space can be used for initialization routines or I/O patches.

```
                        ;
                        ; THIS IS THE STARTING POINT OF THE SELF CONTAINED
                        ; SYSTEM ONCE THE SYSTEM HAS BEEN INITIALIZED.  COMMANDS
                        ; ARE READ FROM THE USER, EXECUTED, AND CONTROL RETURNS
                        ; BACK TO THIS POINT TO READ ANOTHER COMMAND
                        ;
0067 31B110     EOR:    LXI     SP,AREA-18
006A CD3A01             CALL    CRLF    ;PRINT CR  LF
006D 00                 NOP             ;PROMPT WAS HERE IN REV 3
006E 00                 NOP             ;.. NOPS INSERTED
006F 00                 NOP             ;.  TO MATCH EXISTING
0070 00                 NOP             ;.. PROMS. REMOVE WHEN
0071 00                 NOP             ;.  A WHOLE NEW EPROM MASTER NEEDED
0072 CD8500             CALL    READ    ;READ INPUT LINE
0075 23                 INX     H
0076 7E                 MOV     A,M     ;FETCH FIRST CHARACTER
0077 FE3A               CPI     '9'+1   ;COMMAND OR LINE NUMBER?
0079 DA1B04             JC      LINE    ;JUMP IF LINE FOR FILE
007C CDA001             CALL    VALC    ;GET COMMAND VALUES
007F CD5601             CALL    COMM    ;CHECK LEGAL COMMANDS
0082 C36700             JMP     EOR
                        ;
                        ; THIS ROUTINE READS IN A LINE FROM THE TTY AND PLACES
                        ; IT IN AN INPUT BUFFER.
                        ; THE FOLLOWING ARE SPECIAL CHARACTERS
                        ;   CR          TERMINATES READ ROUTINE
                        ;   LF          NOT RECOGNIZED BY ROUTINE
                        ;   CTRL-X      DELETE CURRENT LINE
                        ;   DEL         DELETE CHARACTER
                        ; ALL DISPLAYABLE CHARACTERS BETWEEN BLANK & Z AND THE
                        ; ABOVE ARE RECOGNIZED BY THE READ ROUTINE  ALL OTHERS
                        ; ARE SKIPPED OVER.  THE ROUTINE WILL NOT ACCEPT MORE
                        ; CHARACTERS THAN THE INPUT BUFFER WILL HOLD.
                        ;
0085 210610     READ:   LXI     H,IBUF  ;GET INPUT BUFFER ADDRESS
0088 227410             SHLD    ADDS    ;SAVE ADDRESS
008B 1E02               MVI     E,2     ;INITIALIZE CHARACTER COUNT
008D CDF902     NEXT:   CALL    INS     ;READ A LINE character
0090 78                 MOV     A,B     Get
0091 FE18               CPI     24      ;CHECK FOR CTRL X   This compare deletes current line.
0093 CA6700             JZ      EOR
0096 FE0D               CPI     ASCR    ;GET AN ASCII CR
0098 C2B100             JNZ     DEL
009B 7D                 MOV     A,L
009C FEC6               CPI     IBUF AND 0FFH   ;CHECK FOR FIRST CHAR
009E CA6700             JZ      EOR
00A1 360D               MVI     M,ASCR  ;PLACE CR AT END OF LINE
00A3 23                 INX     H
00A4 3601               MVI     M,1     ;PLACE EOF INDICATOR IN LINE
00A6 23                 INX     H
00A7 3E19               MVI     A,IBUF+83 AND 0FFH
00A9 CDF500             CALL    CLER    ;CLEAR REMAINING BUFFER
00AC 210510             LXI     H,IBUF-1
00AF 73                 MOV     M,E     ;SAVE CHARACTER COUNT
00B0 C9                 RET
00B1 FE7F               CPI     127     ;CHECK FOR DELETE CHARACTER  This compare backs
00B3 C2C600             JNZ     CHAR
00B6 3EC6               MVI     A,IBUF AND 0FFH
```

```
00B8 BD                 CMP     L       ;IS THIS 1ST CHARACTER
00B9 CA8D00             JZ      NEXT
00BC 2B                 DCX     H       ;DECREMENT POINTER
00BD 1D                 DCR     E       ;DECREMENT COUNT
00BE 065F       BSPA:   MVI     B,5FH
00C0 CD0601             CALL    OUT8
00C3 C38D00             JMP     NEXT
00C6 FE20       CHAR:   CPI     ' '     ;CHECK FOR LEGAL CHARACTER
00C8 D2D000             JNC     CHAR1   ;JUMP IF NOT CONTROL CHAR
00CB FE09               CPI     09H     ;IS IT A TAB?
00CD C28D00             JNZ     NEXT    ;IGNORE IF NOT
00D0 FE5B       CHAR1:  CPI     'Z'+1
00D2 D28D00             JNC     NEXT
00D5 47                 MOV     B,A
00D6 CD0601             CALL    OUT8    ;ECHO CHARACTER
00D9 70                 MOV     M,B
00DA 3E17               MVI     A,IBUF+81 AND 0FFH
00DC BD                 CMP     L       ;CHECK FOR END OF LINE
00DD CAFF00             JZ      BSPA
00E0 23                 INX     H
00E1 1C                 INR     E       ;INCREMENT CHARACTER COUNT
00E2 C38D00             JMP     NEXT
                ;
                ;
                ;
                ; THIS ROUTINE IS USED TO BLANK OUT A PORTION OF MEMORY
                ;
00E5 BD         CLER:   CMP     L
00E6 C8                 RZ
00E7 3E20               MVI     M,' '   ;PLACE BLANK IN MEMORY
00E9 23                 INX     H
00EA C3E500             JMP     CLER
                ;
                ; SEE IF TTY INPUT READY AND CHECK FOR CTRL X.
                ; RETURN WITH ZERO SET IFF CTRL-X SEEN.
                ;
00EC DB03       INK:    IN      TTS     ;GET TTY STATUS
00EE 2F                 CMA             ;INVERT STATUS
00EF E602               ANI     TTYDA   ;IS DATA AVAILABLE?
00F1 C0                 RNZ             ;RETURN IF NOT
00F2 CDF900             CALL    IN8     ;GET THE CHAR
00F5 FE18               CPI     'X'-40H ;IS IT A CTRL-X?
00F7 C9                 RET
                ;
                ; THIS ROUTINE READS A BYTE OF DATA FROM THE USART
                ;
00F9 DB03       IN8:    IN      TTS     ;READ USART STATUS
00FB E602               ANI     TTYDA
00FD CAF900             JZ      IN8
0100 DB02               IN      TTI     ;READ DATA
0102 E67F               ANI     7FH     ;STRIP OFF PARITY
0104 47                 MOV     B,A
0105 C9                 RET
                ;
                ; THIS ROUTINE OUTPUTS A BYTE OF DATA TO THE USART
                ;
0106 78         OUT8:   MOV     A,B     ;GET CHAR IN A
0107 D60D               SUI     0DH     ;IS IT A CR?
```

```
0109 CA2C01             JZ      OUT81   ;RESET LINE POSITION IF SO
010C C604               ADI     0DH-09H ;WAS IT A TAB?
010E C22101             JNZ     OUT83   ;CONTINUE IF NOT
0111 C5                 PUSH    B       ;SAVE TAB CHAR
0112 0620               MVI     B,' '   ;OUTPUT A SPACE .
0114 CD2901     OUT84:  CALL    OUT82
0117 3A1911             LDA     LPOS    ;GET CURRENT LINE POSITION
011A E607               ANI     7       ;IS IT A TAB STOP?
011C C21401             JNZ     OUT84   ;OUTPUT ANOTHER SPACE IF NOT
011F C1                 POP     B       ;RESTORE TAB CHAR
0120 C9                 RET
0121 78         OUT83:  MOV     A,B     ;GET CHAR IN A
0122 3C                 INR     A       ;IS IT A NON-PRINTING CHAR?
0123 FE21               CPI     ' '+1
0125 DA2F01             JC      OUT80   ;DON'T BUMP LINE POSITION IF SO
0128 3A1911     OUT82:  LDA     LPOS    ;BUMP LINE POSITION...
012B 3C                 INR     A
012C 321911     OUT81:  STA     LPOS
012F DB03       OUT80:  IN      TTS     ;READ STATUS
0131 E601               ANI     TTYTR
0133 CA2F01             JZ      OUT80
0136 78                 MOV     A,B
0137 D302               OUT     TTO     ;TRANSMIT DATA
0139 C9                 RET

                ;
                ; THIS ROUTINE WILL OUTPUT A CARRIAGE RETURN AND
                ; LINE FEED FOLLOWED BY TWO DELETE CHARACTERS WHICH
                ; PROVIDE TIME FOR PRINT HEAD TO RETURN
                ;
013A 060D       CRLF:   MVI     B,13    ;CR
013C CD0601             CALL    OUTC
013F 060A       LF:     MVI     B,10    ;LF
0141 CD0601             CALL    OUTC
0144 06FF               MVI     B,255
0146 CD0601             CALL    OUTC
0149 C30601             JMP     OUTC
                ;
                ; THIS ROUTINE JUMPS TO A LOCATION IN MEMORY GIVEN BY
                ; THE INPUT COMMAND AND BEGINS EXECUTION OF PROGRAM
                ; THERE
                ;
014C CD2703     EXEC:   CALL    VCHK    ;CHECK FOR PARAMETER
014F CD3A01             CALL    CRLF
0152 2A8A10             LHLD    PBUF    ;FETCH ADDRESS
0155 E9                 PCHL            ;JUMP TO PROGRAM
                ;
                ; THIS ROUTINE CHECKS THE INPUT COMMAND AGAINST ALL
                ; LEGAL COMMANDS STORED IN A TABLE.  IF A LEGAL COMMAND
                ; IS FOUND, A JUMP IS MADE TO THAT ROUTINE   OTHERWISE
                ; AN ERROR MESSAGE IS OUTPUT TO THE USER.
                ;
0156 11E402     COMM:   LXI     D,CTAB  ;COMMAND TABLE ADDRESS
0159 3E04               MVI     A,4     ;LENGTH OF COMMAND
015B 329510             STA     NCHR    ;SAVE
015E CD6501             CALL    COMS    ;SEARCH TABLE
0161 C28104             JNZ     WHAT    ;JUMP IF ILLEGAL COMMAND
0164 E9                 PCHL            ;BE HERE NOW
                ;
```

```
                        ; THIS ROUTINE CHECKS TO SEE IF A BASE CHARACTER STRING
                        ; IS EQUAL TO ANY OF THE STRINGS CONTAINED IN A TABLE
                        ; POINTED TO BY D,E.  THE TABLE CONSISTS OF ANY NUMBER
                        ; OF CHARS  WITH 2 BYTES CONTAINING VALUES ASSOCIATED
                        ; WITH IT.  THE END OF THE TABLE IS MARKED WITH A -1.
                        ; THIS ROUTINE CAN BE USED TO SEARCH THROUGH A COMMAND
                        ; OR SYMBOL TABLE.  ON RETURN, IF THE ZERO FLAG IS SET
                        ; A MATCH WAS FOUND; IF NOT, NO MATCH WAS FOUND.  IF
                        ; A MATCH WAS FOUND  D,E POINT TO THE LAST BYTE
                        ; ASSOCIATED WITH THE CHARACTER STRING.  IF NOT  D,E
                        ; POINT TO THE NEXT LOCATION AFTER THE END OF THE TABLE.
                        ;
0165 2A7410     COMS:   LHLD    ADDS    ;FETCH COMPARE ADDRESS
0168 1A                 LDAX    D       ;GET NEXT BYTE
0169 3C                 INR     A       ;IS IT -1?
016A CA7E01             JZ      COMS1   ;ABORT IF SO
016D 3A9510             LDA     NCHR    ;GET LENGTH OF STRING
0170 4F                 MOV     C A
0171 CD8001             CALL    SEAR    ;COMPARE STRINGS
0174 1A                 LDAX    D       ;FETCH VALUE
0175 6F                 MOV     L A
0176 13                 INX     D
0177 1A                 LDAX    D       ;FETCH VALUE
0178 67                 MOV     H A
0179 C8                 RZ
017A 13                 INX     D       ;SET TO NEXT STRING
017B C36501             JMP     COMS
017E 3C         COMS1:  INR     A       ;CLEAR ZERO FLAG
017F C9                 RET
                        ;
                        ; THIS ROUTINE CHECKS TO SEE IF TWO CHARACTER STRINGS IN
                        ; MEMORY ARE EQUAL.  THE STRINGS ARE POINTED TO BY D E
                        ; AND H,L.  ON RETURN, THE ZERO FLAG SET INDICATES A
                        ; MATCH.  REG C INDICATES THE LENGTH OF THE STRINGS  ON
                        ; RETURN, THE POINTERS POINT TO THE NEXT ADDRESS AFTER
                        ; THE CHARACTER STRINGS.
                        ;
0180 1A         SEAR:   LDAX    D       ;FETCH CHARACTER
0181 BE                 CMP     M       ;COMPARE CHARACTERS
0182 C28C01             JNZ     INCA
0185 23                 INX     H
0186 13                 INX     D
0187 0D                 DCR     C       ;DECREMENT CHARACTER COUNT
0188 C28001             JNZ     SEAR
018B C9                 RET
018C 13         INCA:   INX     D
018D 0D                 DCR     C
018E C28C01             JNZ     INCA
0191 0C                 INR     C       ;CLEAR ZERO FLAG
0192 C9                 RET
                        ;
                        ; THIS ROUTINE ZEROES OUT A BUFFER IN MEMORY WHICH IS
                        ; THEN USED BY OTHER SCANNING ROUTINES
                        ;
0193 AF         ZBUF:   XRA     A       ;GET A ZERO
0194 119A10             LXI     D ABUF+12 ;BUFFER ADDRESS
0197 060C               MVI     B 12    ;BUFFER LENGTH
0199 1B         ZBU1:   DCX     D       ;DECREMENT ADDRESS
```

```
019A 12              STAX    D       ;ZERO BUFFER
019B 05              DCR     B
019C C29901          JNZ     ZBU1
019F C9              RET
                ;
                ; THIS ROUTINE CALLS ETRA TO OBTAIN THE INPUT PARAMETER
                ; VALUES AND CALLS AN ERROR ROUTINE IF AN ERROR OCCURRED
                ; IN THAT ROUTINE
                ;
01A0 CDA701     VALC:   CALL    ETRA    ;GET INPUT PARAMETERS
01A3 DA8104             JC      WHAT    ;JUMP IF ERROR
01A6 C9                 RET
                ;
                ; THIS ROUTINE EXTRACTS THE VALUES ASSOCIATED WITH A
                ; COMMAND FROM THE INPUT STREAM AND PLACES THEM IN THE
                ; ASCII BUFFER  ABUF    IT ALSO CALLS A ROUTINE TO
                ; CONVERT THE ASCII HEXADECIMALS TO BINARY AND STORES
                ; THEM IN THE BINARY BUFFER  BBUF    ON RETURN  CARRY
                ; SET INDICATES AN ERROR IN INPUT PARAMETERS.
                ;
01A7 210000     ETRA:   LXI     H,0     ;GET A ZERO
01AA 228C10             SHLD    BBUF+2  ;ZERO VALUE
01AD 227610             SHLD    FBUF    ;SET NO FILE NAME
01B0 CD9301             CALL    ZBUF    ;ZERO BUFFER
01B3 21C50F             LXI     H,IBUF-1
01B6 23         VAL1:   INX     H
01B7 7E                 MOV     A,M     ;FETCH INPUT CHARACTER
01B8 FE20               CPI     ' '     ;LOOK FOR FIRST CHARACTER
01BA 3F                 CMC
01BB D0                 RNC             ;RETURN IF NO CARRY
01BC C2B601             JNZ     VAL1    ;JUMP IF NO BLANK
01BF 229610             SHLD    PNTR    ;SAVE POINTER
01C2 CD3909             CALL    SBLK    ;SCAN TO FIRST PARAMETER
01C5 3F                 CMC
01C6 D0                 RNC             ;RETURN IF CR
01C7 FE2F               CPI     '/'
01C9 C2F1C1             JNZ     VAL5    ;NO FILE NAME
01CC 117610             LXI     D,FBUF  ;NAME FOLLOWS PUT IN FBUF
01CF 0E05               MVI     C,NMLEN
01D1 23         VAL2:   INX     H
01D2 7E                 MOV     A,M
01D3 FE2F               CPI     '/'
01D5 CAE101             JZ      VAL3
01D8 0D                 DCR     C
01D9 FA8104             JM      WHAT
01DC 12                 STAX    D       ;STORE FILE NAME
01DD 13                 INX     D
01DE C3D101             JMP     VAL2
01E1 3E20       VAL3:   MVI     A,' '   ;GET AN ASCII SPACE
01E3 0D         VAL4:   DCR     C
01E4 FAEC01             JM      DONE
01E7 12                 STAX    D       ;FILL IN WITH SPACES
01E8 13                 INX     D
01E9 C3E301             JMP     VAL4
01EC CD4509     DONE:   CALL    SBL2
01EF 3F                 CMC
01F0 D0                 RNC
01F1 117E10     VAL5:   LXI     D,ABUF
```

```
01F4 CD9A0B             CALL    ALPS    ;PLACE PARAMETER IN BUFFER
01F7 78                 MOV     A B     ;GET DIGIT COUNT
01F8 FE05               CPI     5       ;CHECK NUMBER OF DIGITS
01FA 3F                 CMC
01FB D8                 RC              ;RETURN IF TOO MANY DIGITS
01FC 017E10             LXI     B ABUF
01FF CD4802             CALL    AHEX    ;CONVERT VALUE
0202 D8                 RC              ;ILLEGAL CHARACTER
0203 228A10             SHLD    BBUF    ;SAVE   IN BINARY BUFFER
0206 217E10             LXI     H.ABUF
0209 CDE305             CALL    NORM    ;NORMALIZE ASCII VALUE
020C CD3909             CALL    SELK    ;SCAN TO NEXT PARAMETER
020F 3F                 CMC
0210 D0                 RNC             ;RETURN IF CR
0211 118210             LXI     D ABUF-4
0214 CD9A0B             CALL    ALPS    ;PLACE PARAMETER IN BUFFER
0217 78                 MOV     A.B     ;GET DIGIT COUNT
0218 FE05               CPI     5       ;CHECK NUMBER OF DIGITS
021A 3F                 CMC
021B D8                 RC              ;RETURN IF TOO MANY DIGITS
021C 018210             LXI     B ABUF-4
021F CD4802             CALL    AHEX    ;CONVERT VALUE
0222 D8                 RC              ;ILLEGAL VALUE
0223 228C10             SHLD    BBUF-2  ;SAVE IN BINARY BUFFER
0226 218210             LXI     H.ABUF-4
0229 CDE305             CALL    NORM    ;NORMALIZE ASCII VALUE
022C B7                 ORA     A       ;CLEAR CARRY
022D C9                 RET
                ;
                ; THIS ROUTINE FETCHES DIGITS FROM THE BUFFER ADDRESSED
                ; BY B,C AND CONVERTS THE ASCII DECIMAL DIGITS INTO
                ; BINARY.  UP TO A 16-BIT VALUE CAN BE CONVERTED   THE
                ; SCAN STOPS WHEN A BINARY ZERO IS FOUND IN THE BUFFER
                ;
022E 210000     ADEC:   LXI     H 0     ;GEP A 16 BIT ZERO
0231 0A         ADE1:   LDAX    B       ;FETCH ASCII DIGIT
0232 B7                 ORA     A       ;SET ZERO FLAG
0233 C8                 RZ              ;RETURN IFF FINISHED
0234 54                 MOV     D H     ;SAVE CURRENT VALUE
0235 5D                 MOV     E L     ;SAVE CURRENT VALUE
0236 29                 DAD     H       ;TIMES TWO
0237 29                 DAD     H       ;TIMES TWO
0238 19                 DAD     D       ;ADD IN ORIGINAL VALUE
0239 29                 DAD     H       ;TIMES TWO
023A D630               SUI     48      ;ASCII BIAS
023C FE0A               CPI     10      ;CHECK FOR LEGAL VALUE
023E 3F                 CMC
023F D8                 RC              ;RETURN IF ERROR
0240 5F                 MOV     E A
0241 1600               MVI     D 0
0243 19                 DAD     D       ;ADD IN NEXT DIGIT
0244 03                 INX     B       ;INCREMENT POINTER
0245 C33102             JMP     ADE1
                ;
                ;
                ;.THIS ROUTINE FETCHES DIGITS FROM THE BUFFER ADDRESSED
                ; BY B C AND CONVERTS THE ASCII HEXADECIMAL DIGITS INTO
                ; BINARY.  UP TO A 16-BIT VALUE CAN BE CONVERTED   THE
```

```
                        ; SCAN STOPS WEEN A BINARY ZERO IS FOUND IN THE BUFFER.
                        ;
0249 210000     AHEX    LXI     H 0     ;GET A 16 BIT ZERO
024B 0A         AHE1:   LDAX    B       ;FETCH ASCII DIGIT
024C B7                 ORA     A       ;SET ZERO FLAG
024D C8                 RZ              ;RETURN IF DONE
024E 29                 DAD     H       ;LEFT SHIFT
024F 29                 DAD     H       ;LEFT SHIFT
0250 29                 DAD     H       ;LEFT SHIFT
0251 29                 DAD     H       ;LEFT SHIFT
0252 CD5F02             CALL    AHS1    ;CONVERT TO BINARY
0255 FE10               CPI     10H     ;CHECK FOR LEGAL VALUE
0257 3F                 CMC
0259 D8                 RC              ;RETURN IF ERROR
0259 85                 ADD     L
025A 6F                 MOV     L.A
025B 03                 INX     B       ;INCREMENT POINTER
025C C34B02             JMP     AHE1
                        ;
                        ; THIS SUBROUTINE CONVERTS ASCII HEX DIGITS INTO BINARY
                        ;
025F D630       AHS1:   SUI     48      ;ASCII BIAS
0261 FE0A               CPI     10      ;DIGIT 0-10
0263 D8                 RC
0264 D607               SUI     7       ;ALPHA BIAS
0266 C9                 RET
                        ;
                        ;
                        ; THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII
                        ; HEXADECIMAL AND OUTPUTS THE CHARACTERS TO THE TTY.
                        ;
0267 CDAC02     HOUT:   CALL    BINH    ;CONVERT VALUE
026A 217410             LXI     H HCON  ;CONVERSION AREA
026D 46         CHOT:   MOV     B,M     ;FETCH OUTPUT CHARACTER
026E CD0601             CALL    OUT8    ;OUTPUT CHARACTER
0271 23                 INX     H
0272 46                 MOV     B M     ;FETCH CHARACTER
0273 C30601             JMP     OUT8    ;OUTPUT CHARACTER AND RETURN
                        ;
                        ; THIS ROUTINE DOES THE SAME AS ABOVE BUT OUTPUTS A
                        ; BLANK AFTER THE LAST CHARACTER
                        ;
0276 CD6702     HOTB:   CALL    HOUT    ;CONVERT AND OUTPUT
                        ;
                        ; THIS ROUTINE OUTPUTS A BLANK.
                        ;
0279 0620       BLK1:   MVI     B,' '   ;GET A BLANK
027B C30601             JMP     OUT8    ;OUTPUT IT AND RETURN
                        ;
                        ; THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII
                        ; DECIMAL DIGITS AND OUTPUTS THE CHARACTERS TO THE TTY
                        ;
027E CDC902     DOUT:   CALL    BIND    ;CONVERT VALUE
0281 CD6A02             CALL    HOUT 3  ;OUTPUT VALUE  2 DIGITS
0284 23                 INX     H
0285 46                 MOV     B M     ;GET LAST DIGIT
0286 C30601             JMP     OUT8    ;OUTPUT AND RETURN
                        ;
```

```
                ; THIS ROUTINE IS USED BY OTHER ROUTINES TO INCREMENT
                ; THE STARTING ADDRESS IN A COMMAND AND COMPARE IT WITH
                ; THE FINAL ADDRESS IN THE COMMAND.  ON RETURN  THE
                ; CARRY FLAG SET INDICATES THAT THE FINAL ADDRESS HAS
                ; BEEN REACHED.
                ;
02B9 2A8A10     ACHK:   LHLD    BBUF    ;FETCH START ADDRESS
02BC 3A8D10             LDA     BBUF+3  ;STOP ADRESS  HIGH
02BF BC                 CMP     H       ;COMPARE ADDRESSES
02C0 C2CB02             JNZ     ACH1
02C3 3A8C10             LDA     BBUF+2  ;STOP ADDRESS  LOW
02C6 BD                 CMP     L       ;COMPARE ADDRESSES
02C7 C2CB02             JNZ     ACH1
02CA 37                 STC             ;SET CARRY IF EQUAL
02CB 23         ACH1:   INX     H       ;INCREMENT START ADDRESS
02CC 228A10             SHLD    BBUF    ;STORE START ADDRESS
02CF C9                 RET
                ;
                ;
                ; THIS ROUTINE OUTPUTS CHARACTERS OF A STRING
                ; UNTIL A CARRIAGE RETURN IS FOUND.
                ;
02A0 46         SCRN:   MOV     B,M     ;FETCH CHARACTER
02A1 3E0D               MVI     A,13    ;CARRIAGE RETURN
02A3 B8                 CMP     B       ;CHARACTER = CR?
02A4 C8                 RZ
02A5 CD0601             CALL    OUT8    ;OUTPUT CHARACTER
02A8 23                 INX     H       ;INCREMENT ADDRESS
02A9 C3A002             JMP     SCRN
                ;
                ;
                ; THIS ROUTINE CONVERTS THE BINARY VALUE IN REG A INTO
                ; ASCII HEXADECIMAL DIGITS AND STORES THEM IN MEMORY
                ;
02AC 217410     BINE:   LXI     H,HCON  ;CONVERSION
02AF 47                 MOV     B,A     ;SAVE VALUE
02B0 1F                 RAR
02B1 1F                 RAR
02B2 1F                 RAR
02B3 1F                 RAR
02B4 CDBF02             CALL    BIN1
02B7 77                 MOV     M,A
02B8 23                 INX     H
02B9 78                 MOV     A,B
02BA CDBF02             CALL    BIN1    ;CONVERT TO ASCII
02BD 77                 MOV     M,A
02BE C9                 RET
                ;
                ; THIS ROUTINE CONVERTS A VALUE TO HEXADECIMAL
                ;
02BF E60F       BIN1:   ANI     0FH     ;LOW 4 BITS
02C1 C630               ADI     48      ;CONVERT TO ASCII
02C3 FE3A               CPI     58      ;DIGIT 0-9
02C5 D8                 RC
02C6 C607               ADI     7       ;MODIFY FOR A-F
02C8 C9                 RET
                ;
                ;
```

```
                        ; THIS ROUTINE CONVERTS THE BINARY VALUE IN REG A INTO
                        ; ASCII DECIMAL DIGITS AND STORES THEM IN MEMORY
                        ;
02C9 217410     BIN:    LXI     H.HCON  ;CONVERSION ADDRESS
02CC 0664               MVI     B.100
02CE CDDA02             CALL    BID1    ;CONVERT HUNDREDS DIGIT
02D1 060A               MVI     B 10
02D3 CDDA02             CALL    BID1    ;CONVERT TENS DIGIT
02D6 C630               ADI     '0'     ;GET UNITS DIGIT
02D8 77                 MOV     M.A     ;STORE IN MEMORY
02D9 C9                 RET
                        ;
                        ; THIS ROUTINE CONVERTS A VALUE TO DECIMAL
                        ;
02DA 362F       BID1:   MVI     M. 0'-1 ;INITIALIZE DIGIT COUNT
02DC 34                 INR     M
02DD 90                 SUB     B       ;CHECK DIGIT
02DE D2DC02             JNC     BID1 2
02E1 80                 ADD     B       ;RESTORE VALUE
02E2 23                 INX     H
02E3 C9                 RET
                        ;
                        ; LEGAL COMMAND TABLE
                        ;
02E4 44554D50   CTAB:   DB      'DUMP'  ;DUMP COMMAND
02E8 2F03               DW      DUMP    ;COMMAND ADDRESS
02EA 45584543           DB      'EXEC'  ;EXECUTE COMMAND
02EE 4C01               DW      EXEC    ;COMMAND ADDRESS
02F0 454E5452           DB      'ENTR'  ;ENTER COMMAND
02F4 9D04               DW      ENTR
02F6 46494C45           DB      'FILE'  ;FILE COMMAND
02FA 6503               DW      FILE    ;COMMAND ADDRESS
02FC 4C495354           DB      'LIST'  ;LIST COMMAND
0300 F605               DW      LIST    ;COMMAND ADDRESS
0302 44454C54           DB      'DELT'  ;DELETE COMMAND
0306 0D06               DW      DELL    ;COMMAND ADDRESS
0308 4153534D           DB      'ASSM'  ;ASSEMBLE COMMAND
030C 8406               DW      ASSM    ;COMMAND ADDRESS
030E 50414745           DB      'PAGE'  ;PAGE TRANSFER COMMAND
0312 4903               DW      PAGE    ;COMMAND ADDRESS
0314 43555354           DB      'CUST'  ;CUSTOMER COMMAND
0318 0020               DW      2000E   ;COMMAND ADDRESS
031A 42524547           DB      'BREK'  ;BREAKPOINT COMMAND
031E F50C               DW      BREAK   ;COMMAND ADDRESS
0320 50524F43           DB      'PROC'  ;;PROCEED COMMAND
0324 AF0D               DW      PROC    ;COMMAND ADDRESS
0326 FF                 DB      0FFH
                        ;
                        ; THIS ROUTINE CHECKS IF ANY PARAMETERS WERE ENTERED
                        ; WITH THE COMMAND  IF NOT AN ERROR MESSAGE IS ISSUED
                        ;
0327 3A7E10     VCHK:   LDA     ABUF    ;FETCH PARAMETER BYTE
032A B7                 ORA     A       ;SET FLAGS
032B CA8104             JZ      WHAT    ;NO PARAMETER
032E C9                 RET
                        ;
                        ;
                        ; THIS ROUTINE DUMPS OUT THE CONTENTS OF MEMORY FROM
```

SCST Software adds — SYMB RESE ELOAD ESAVE AUTO CATA

```
                        ; THE START TO FINAL ADDRESSES GIVEN IN THE COMMAND.
                        ;
032F CD2703     DUMP:   CALL    VCHK    ;CHECK FOR PARAMETERS
0332 CD3A01     DUMS:   CALL    CRLF    ;START NEW LINE
0335 2A8A10     DUM1:   LHLD    BBUF    ;FETCH MEMORY ADDRESS
0338 7E                 MOV     A,M
0339 CD7602             CALL    HOTB    ;OUTPUT VALUE
033C CD8902             CALL    ACHK    ;CHECK ADDRESS
033F D8                 RC              ;RETURN IF FINISHED
0340 7D                 MOV     A,L     ;IS NEXT ADDRESS
0341 E60F               ANI     0FH     ;  DIVISIBLE BY 16?
0343 C23503             JNZ     DUM1
0346 C33203             JMP     DUMS
                        ;
                        ;
                        ; THIS ROUTINE WILL MOVE 256 BYTES FROM 1ST ADDRESS
                        ; GIVEN IN COMMAND TO 2ND ADDRESS IN COMMAND
                        ;
0349 CD2703     PAGE:   CALL    VCHK    ;CHECK FOR PARAMETER
034C 3A8210             LDA     ABUF-4  ;FETCH 2ND PARAMETER
034F B7                 ORA     A       ;DOES 2ND PARAMETER EXIST?
0350 CA8104             JZ      WHAT
0353 2A8A10             LHLD    BBUF    ;FETCH MOVE TO ADDRESS
0356 EB                 XCHG
0357 2A8C10             LHLD    BBUF-2  ;FETCH MOVE TO ADDRESS
035A 0600               MVI     B,0     ;SET COUNTER
035C 1A         PAG1:   LDAX    D
035D 77                 MOV     M,A
035E 23                 INX     H
035F 13                 INX     D
0360 05                 DCR     B       ;DECREMENT COUNT
0361 C25C03             JNZ     PAG1
0364 C9                 RET
                        ;
                        ;
                        ;
                        ; THIS ROUTINE INITIALIZES THE BEGINNING OF FILE ADDRESS
                        ; AND END OF FILE ADDRESS AS WELL AS THE FILE AREA
                        ; WHEN THE FILE COMMAND IS USED
                        ;
0365 CD3A01     FILE:   CALL    CRLF
                        ; CHECK FOR FILE PARAMETERS
0368 3A7E10             LDA     FBUF
036B B7                 ORA     A
036C CAE003             JZ      FOUT    ;NO - GO LIST
036F CD3F04             CALL    FSEA    ;LOOK UP FILE
0372 EB                 XCHG            ;PNTR IN DE
0373 C28A03             JNZ     TEST
                        ; NO ENTRY
0376 3A7E10             LDA     ABUF    ;CHECK FOR PARAM
0379 B7                 ORA     A
037A CA8404             JZ      WHA1    ;NO?? - ERROR
                        ; CHECK FOR ROOM IN DIRECTORY
037D 3A7D10             LDA     FEF
0380 B7                 ORA     A
0381 C29F03             JNZ     ROOM
0384 219204             LXI     H,EMES1
0387 C38704             JMP     MESS
```

```
                ; ENTRY FOUND ARE THESE PARAMETERS
038A 3A7E10     TEST:   LDA     ABUF
038D B7                 ORA     A
038E CA3203             JZ      SWAPS
0391 2A8A10             LHLD    BBUF
0394 7C                 MOV     A,H
0395 B5                 ORA     L
0396 CAB203             JZ      SWAPS
0399 219704             LXI     H,EMES2 ;NO-NO CAN'T DO
039C C38704             JMP     MESS    ;IT - DELETE FIRST
                ; MOVE FILE NAME TO BLOCK POINTED TO BY FREAD
039F 2A7B10     ROOM:   LHLD    FREAD
03A2 EB                 XCHG
03A3 217610             LXI     H,FBUF  ;FILE NAME POINTER IN H L
03A6 D5                 PUSH    D
03A7 0E05               MVI     C,NMLEN ;NAME LENGTH COUNT
03A9 7E         MOV23:  MOV     A,M
03AA 12                 STAX    D
03AB 13                 INX     D
03AC 0D                 DCR     C       ;TEST COUNT
03AD 23                 INX     H
03AE C2A903             JNZ     MOV23
03B1 D1                 POP     D       ;RESTORE ENTRY POINTER
                ; MAKE FILE POINTED TO BY D,E CURRENT
03B2 212410     SWAPS:  LXI     H,FILE0
03B5 0E0D               MVI     C,FELEN ;ENTRY LENGTH
03B7 1A         SWAP:   LDAX    D
03B8 46                 MOV     B,M
03B9 77                 MOV     M,A     ;EXCHANGE
03BA 78                 MOV     A,B
03BB 12                 STAX    D
03BC 13                 INX     D
03BD 23                 INX     H       ;BUMP POINTERS
03BE 0D                 DCR     C       ;TEST COUNT
03BF C2B703             JNZ     SWAP
                ; CHECK FOR 2ND PARAMETER
03C2 3A7E10             LDA     ABUF
03C5 B7                 ORA     A
03C6 CAEA03             JZ      FOOT    ;NO SECOND PARAMETER
                ; PROCESS 2ND PARAMETER
03C9 2A8A10             LHLD    BBUF    ;GET ADDRESS
03CC 222910             SHLD    BOFP    ;SET BEGIN
03CF 222B10             SHLD    EOFP    ;SET END
03D2 7D                 MOV     A,L     ;IS ADDRESS ZERO?
03D3 B4                 ORA     H
03D4 CAD903             JZ      FIL35   ;YES
03D7 3601       FIL30:  MVI     M,1     ;NON-ZERO - SET EOF
03D9 AF         FIL35:  XRA     A       ;AND MAX LINE #
03DA 322D10             STA     MAXL
03DD C3EA03             JMP     FOOT    ;OUTPUT PARAMETERS
03E0 3ACA10     FOUT:   LDA     IBUF+4
03E3 FE53               CPI     'S'     ;IS COMMAND FILES?
03E5 0E06               MVI     C,MAXFIL
03E7 CAEC03             JZ      FOUL
03EA 0E01       FOOT:   MVI     C,1
                ; OUTPUT THE # OF ENTRIES IN C
03EC 212410     FOUL:   LXI     H,FILE0
03EF 79                 MOV     A,C
```

```
03F0 327D10    FINF:   STA     FOCNT   ;SAVE COUNT
03F3 E5                PUSH    H
03F4 110500            LXI     D,NMLEN
03F7 19                DAD     D
03F8 7E                MOV     A,M
03F9 B7                ORA     A
03FA C20A04            JNZ     FOOD    ;NON ZERO  OK TO OUTPUT
03FD 23                INX     H
03FE 86                ADD     M
03FF 23                INX     H
0400 C20A04            JNZ     FOOD
0403 33                INX     SP
0404 33                INX     SP
0405 23                INX     H
0406 23                INX     H
0407 C31F04            JMP     FEET
                ; HAVE AN ENTRY TO OUTPUT
040A E1         FOOD:  POP     H       ;PTR
040B 0E05              MVI     C,NMLEN
040D 46         FAST:  MOV     B,M     ;LOAD CHARACTER TO B
040E CD0601            CALL    OUTB
0411 0D                DCR     C
0412 23                INX     H
0413 C20D04            JNZ     FAST    ;DO THE REST
                ; NOW OUTPUT BEGIN END PTRS
0416 CD2B04            CALL    FOCL    ;OUTPUT BEGIN
0419 CD2B04            CALL    FOOL    ;OUTPUT END
041C CD3A01            CALL    CRLF    ;AND C R
                ; TEST COUNT  H L POINTS PAST EOFP
041F 110400    FEET:   LXI     D,FELEN-NMLEN-4
0422 19                DAD     D       ;MOVE TO NEXT ENTRY
0423 3A7D10            LDA     FOCNT
0426 3D                DCR     A       ;TEST COUNT
0427 C2F003            JNZ     FINE    ;MORE TO DO
042A C9                RET             ;DONE!
                ; OUTPUT NUMBER POINTED TO BY H L
                ; ON RET, H,L POINT 2 WORDS LATER
042B CD7902    FOOL:   CALL    BLK1    ;SPACE
042E 23                INX     H
042F 7E                MOV     A,M
0430 2B                DCX     H
0431 E5                PUSH    H
0432 CD6702            CALL    HOUT    ;OUTPUT
0435 E1                POP     H
0436 7E                MOV     A,M
0437 23                INX     H
0438 23                INX     H
0439 E5                PUSH    H
043A CD7602            CALL    HOTB    ;OUTPUT
043D E1                POP     H       ;RESTORE H L
043E C9                RET
                ; SEARCH THE FILE DIRECTORY FOR THE FILE
                ; WHOSE NAME IS IN FFUF.
                ; RETURN IF FOUND, ZERO IS OFF, H,L POINT TO
                ; ENTRY WHILE SEARCHING  ON ENTRY FOUND WITH ADDR
                ; ZERO  SET FEF TO >0 AND FREAD TO THE ADDR OF ENTRY
                ;
043F AF         FSEA:  XRA     A
```

```
0440 327D10             STA     FEF     ;CLAIM NO FREE ENTRIES
0443 0606               MVI     B,MAXFIL        ;COUNT OF ENTRIES
0445 112410             LXI     D,FILE0 ;TABLE ADDRESS
0448 217610     FSE10:  LXI     H,FBUF
044B 0E05               MVI     C,NMLEN
044D CD8001             CALL    SEAR    ;TEST STRINGS
0450 F5                 PUSH    PSW     ;SAVE FLAG
0451 D5                 PUSH    D
0452 1A                 LDAX    D       ;GET BOFF
0453 B7                 ORA     A       ;EMPTY ENTRY?
0454 C25704             JNZ     FSE20
0457 13                 INX     D       ;STORE OTHER WORD
0458 1A                 LDAX    D
0459 B7                 ORA     A
045A C25704             JNZ     FSE20   ;NOPE-GO TEST FOR MATCH
045D EB                 XCHG
045E 11FAFF             LXI     D,-NMLEN-1
0461 19                 DAD     D       ;MOV TO BEGINNING
0462 227B10             SHLD    FREAD   ;SAVE ADDR
0465 7A                 MOV     A,D
0466 327D10             STA     FEF     ;SET FREE ENTRY FOUND
0469 E1                 POP     H       ;RESTOR INTERIM PTR
046A F1                 POP     PSW     ;UNJUNK STACK
                ; MOVE TO NEXT ENTRY
046B 110800     FSE15:  LXI     D,FELEN-NMLEN
046E 19                 DAD     D
046F EB                 XCHG            ;NEXT ENTRY ADDR IN DE
0470 05                 DCR     B       ;TEST COUNT
0471 C8                 RZ              ;DONE--NOPE
0472 C34804             JMP     FSE10   ;TRY NEXT
                ;ENTRY WASN'T FREE. TEST FOR MATCH
0475 E1         FSE20:  POP     H
0476 F1                 POP     PSW
0477 C26B04             JNZ     FSE15   ;IF ZERO CLEAR. NO MATCH
                ;ENTRY FOUND
047A 11FBFF             LXI     D,-NMLEN        ;BACKUP
047D 19                 DAD     D       ;H,L POINTS TO ENTRY
047E 7A                 MOV     A,D
047F B7                 ORA     A       ;CLEAR ZERO
0480 C9                 RET             ;THAT'S ALL
                ;
                ;
                ; OUTPUT ERROR MESSAGE FOR ILLEGAL COMMAND
                ;
0481 CD3A01     WHAT:   CALL    CRLF    ;OUT CRLF
0484 219D04     WHA1:   LXI     H,EMES  ;MESSAGE ADDRESS
0487 CDA002     MESS:   CALL    SCRN
048A C36700             JMP     EOR
                ;
048D 5748415420EMES:    DB      'WHAT',13
0492 46554C4C0DEMES1:   DB      'FULL',13
0497 4E4F204F4DEMES2:   DB      'NO NO  13
                ;
                ;
                ; CALL ROUTINE TO ENTER DATA INTO MEMORY
                ; AND CHECK FOR ERROR ON RETURN
                ;
                ; THIS ROUTINE IS USED TO ENTER DATA VALUES INTO MEMORY
```

```
                        ; EACH VALUE IS ONE BYTE AND IS WRITTEN IN HEXADECIMAL
                        ; VALUES GREATER THAT 255 WILL CAUSE CARRY TO BE SET
                        ; AND RETURN TO BE MADE TO CALLING PROGRAM
                        ;
049D CD2703             ENTR:   CALL    VCHK    ;CHECK FOR PARAMETERS
04A0 CDA904                     CALL    ENTS
04A3 DA8104                     JC      WHAT
04A6 C33A01                     JMP     CRLF
                        ;
                        ;
002F =                  FEND    EQU     '/'     ;TERMINATION CHAR
04A9 CD3A01             ENTS:   CALL    CRLF
04AC CD8500                     CALL    READ    ;READ INPUT DATA
04AF 21C610                     LXI     H,IBUF  ;SET LINE POINTER
04B2 229610                     SHLD    PNTR    ;SAVE POINTER
04B5 CD9301             ENT1:   CALL    ZBUF    ;CLEAR BUFFER
04B8 CD3909                     CALL    SBLK    ;CAN TO FIRST VALUE
04BB DAA904                     JC      ENTS    ;JUMP IF CR FOUND
04BE FE2F                       CPI     FEND
04C0 C8                         RZ              ;RETURN CARRY IS ZERO
04C1 CD9A0B                     CALL    ALPS    ;PLACE VALUE IN BUFFER
04C4 78                         MOV     A,B     ;GET DIGIT COUNT
04C5 FE03                       CPI     3       ;CHECK NUM OF DIGITS
04C7 3F                         CMC
04C8 D8                         RC              ;RETURN IF MORE THAN    2 DIGITS
04C9 017E10                     LXI     B,ABUF  ;CONVERSION ADDRESS
04CC CD4902                     CALL    AHEX    ;CONVERT VALUE
04CF D8                         RC              ;ERROR IN HEX CHARACTER
04D0 7D                         MOV     A,L
04D1 2A8A10                     LHLD    BBUF    ;FETCH MEMORY ADDRESS
04D4 77                         MOV     M,A     ;PUT IN MEMORY
04D5 CD9B02                     CALL    ACH1    ;INCREMENT MEMORY LOCATION
04D8 C3B504                     JMP     ENT1
                        ;
                        ;
                        ;
                        ; THIS ROUTINE IS USED TO ENTER LINES INTO THE FILE
                        ; AREA. THE LINE NUMBER ('0000 9999'). NEXT IT IS CHECKED TO SEE
                        ; IF IT IS GREATER THAN THE MAXIMUM CURRENT LINE NUMBER
                        ; IF IT IS  THE NEXT LINE IS INSERTED AT THE END OF THE
                        ; CURRENT FILE AND THE MAXIMUM LINE NUMBER IS UPDATED AS
                        ; WELL AS THE END OF FILE POSITION. LINE NUMBERS THAT
                        ; ALREADY EXIST ARE INSERTED INTO THE FILE AREA AT THE
                        ; APPROPRIATE PLACE AND ANY EXTRA CHARACTERS IN THE OLD
                        ; LINE ARE DELETED.
                        ;
04DB 3A2410             LINE:   LDA     FILE0   ;IS A FILE DEFINED? ..
04DE B7                         ORA     A
04DF CA8104                     JZ      WHAT    ;ABORT IF NOT
04E2 0E04                       MVI     C,4     ;NO OF DIGITS TO CHECK
04E4 21C510                     LXI     H,IBUF-1        ;INITIALIZE ADDRESS
04E7 23                 LICK:   INX     H
04E8 7E                         MOV     A,M     ;FETCH LINE DIGIT
04E9 FE30                       CPI     '0'     ;CHECK FOR VALID NUMBER
04EB DA8104                     JC      WHAT
04EE FE3A                       CPI     '9'+1
04F0 D28104                     JNC     WHAT
```

```
04F3 0D              DCR     C
04F4 C2E704          JNZ     LICK
04F7 227410          SHLD    ADDS    ;FIND ADDRESS
04FA 113010          LXI     D,MAXL-3        ;GET ADDRESS
04FD CDC805          CALL    COM0
0500 D22005          JNC     INSR
                ; GET HERE IF NEW LINE IS GREATER THAN MAXIMUM LINE #
0503 23              INX     H
0504 CDB805          CALL    LODM    ;GET NEW LINE NUMBER
0507 213010          LXI     H,MAXL-3
050A CDC005          CALL    STOM    ;MAKE IT MAXIMUM LINE NUMBER
050D 11C510          LXI     D,IBUF-1
0510 2A2B10          LHLD    EOFP    ;END OF FILE POSITION
0513 0E01            MVI     C,1
0515 CDA605          CALL    LMOV    ;PLACE LINE IN FILE
0518 3601       SEOF:MVI     M,1     ;END OF FILE INDICATOR
051A 222B10          SHLD    EOFP    ;END OF FILE ADDRESS
051D C36700          JMP     EOR
                ; GET HERE IF NEW LINE MUST BE INSERTED INTO ALREADY
                ; EXISTING FILE AREA
0520 CD7805     INSR:CALL    FIN1    ;FIND LINE IN FILE
0523 0E02            MVI     C,2
0525 CA2905          JZ      EQUL
0528 0D              DCR     C       ;NEW LN NOT EQUAL TO SOME OLD LN
0529 46         EQUL:MOV     B,M
052A 2B              DCX     H
052B 3602            MVI     M,2     ;MOVE LINE INDICATOR
052D 227210          SHLD    INSP    ;INSERT LINE POSITION
0530 3AC510          LDA     IBUF-1  ;NEW LN COUNT
0533 0D              DCR     C
0534 CA3E05          JZ      LT      ;NEW LN NOT = OLD LN
0537 90              SUB     B       ;COUNT DIFFERENCE
0538 CA6105          JZ      ZERO    ;LINE LENGTHS EQUAL
053B DA5105          JC      GT
                ; GET HERE IF # OF CHARS IN OLD LINE > # OF CHARS IN
                ; NEW LINE OR NEW LINE # WAS NOT EQUAL TO SOME OLD
                ; LINE #
053E 2A2B10     LT:  LHLD    EOFP    ;END OF FILE ADDRESS
0541 54              MOV     D,H
0542 5D              MOV     E,L
0543 CDA105          CALL    ADR     ;MOVE TO ADDRESS
0546 222B10          SHLD    EOFP    ;NEW END OF FILE ADDRESS
0549 0E02            MVI     C,2
054B CDAF05          CALL    RMOV    ;OPEN UP FILE AREA
054E C36105          JMP     ZERO
                ; GET HERE IF # OF CHARS IN OLD LINE < # OF CHARS IN
                ; NEW LINE.
0551 2F         GT:  CMA
0552 3C              INR     A       ;COUNT DIFFERENCE
0553 54              MOV     D,H
0554 5D              MOV     E,L
0555 CDA105          CALL    ADR
0558 EB              XCHG
0559 CDA605          CALL    LMOV    ;DELETE EXCESS CHAR IN FILE
055C 3601            MVI     M,1     ;E-O-F INDICATOR
055E 222B10          SHLD    EOFP    ;E-O-F ADDRESS
                ; GET HERE TO INSERT CURRENT LINE INTO FILE AREA
0561 2A7210     ZERO:LHLD    INSP    ;INSERT ADDRESS
```

TAPE #3 ← TAPE #4

```
0564 360D               MVI     M ASCR
0566 23                 INX     H
0567 11C510             LXI     D.IBUF-1        ;NEW LINE ADDRESS
056A 0E01               MVI     C 1     ;CHECK VALUE
056C CDA605             CALL    LMOV    ;PLACE LINE IN FILE
056F C36700             JMP     EOR
                ;
                ;
                ; THIS ROUTINE IS USED TO FIND A LN IN THE FILE AREA
                ; WHICH IS GREATER THAN OR EQUAL TO THE CURRENT LINE #
0572 218110     FIND:   LXI     H,ABUF+3        ;BUFFER ADDRESS
0575 227410             SHLD    ADDS    ;SAVE ADDRESS
0578 2A2910     FIN1:   LHLD    BOFP    ;BEGIN FILE ADDRESS
057B 7C                 MOV     A,H     ;RETURN TO MONITOR IF
057C B5                 ORA     L       ;  FILE IS EMPTY...
057D CA6700             JZ      EOR
0580 CD9A05     FI1:    CALL    EO1     ;CHECK FOR END OF FILE
0583 EB                 XCHG
0584 2A7410             LHLD    ADDS    ;FETCH FIND ADDRESS
0587 EB                 XCHG
0588 3E04               MVI     A,4
058A CDA105             CALL    ADR     ;BUMP LINE ADDRESS
058D CDC805             CALL    COM0    ;COMPARE LINE NUMBERS
0590 D8                 RC
0591 C8                 RZ
0592 7E         FI2:    MOV     A,M
0593 CDA105             CALL    ADR     ;NEXT LINE ADDRESS
0596 C38005             JMP     FI1
                ;
                ;
                ; WHEN SEARCHING THROUGH THE FILE AREA  THIS ROUTINE
                ; CHECKS TO SEE IF THE CURRENT ADDRESS IS THE END OF
                ; FILE
                ;
0599 23         EOF:    INX     H
059A 3E01       EO1:    MVI     A 1     ;E-O-F INDICATOR
059C BE                 CMP     M
059D C0                 RNZ
059E C36700             JMP     EOR
                ;
                ;
                ; THIS ROUTINE IS USED TO ADD A VALUE TO AN ADDRESS
                ; CONTAINED IN REGISTER H L
                ;
05A1 85         ADR:    ADD     L
05A2 6F                 MOV     L A
05A3 D0                 RNC
05A4 24                 INR     H
05A5 C9                 RET
                ;
                ;
                ; THIS ROUTINE WILL MOVE CHARACTER STRINGS FROM ONE
                ; LOCATION OF MEMORY TO ANOTHER
                ; CHARACTERS ARE MOVED FROM LOCATION ADDRESSED BY D.E
                ; TO LOCATION ADDRESSED BY H.L.  ADDITIONAL CHARACTERS
                ; ARE MOVED BY BUMPING POINTERS UNTIL THE CHARACTER IN
                ; REG C IS FETCHED.
                ;
```

```
05A6 1A        LMOV:   LDAX    D       ;FETCH CHARACTER
05A7 13                INX     D       ;INCREMENT FETCH ADDRESS
05A8 B9                CMP     C       ;TERMINATION CHARACTER
05A9 C8                RZ
05AA 77                MOV     M,A     ;STORE CHARACTER
05AB 23                INX     H       ;INCREMENT STORE ADDRESS
05AC C3A605            JMP     LMOV
                ;
                ;
                ; THIS ROUTINE IS SIMILAR TO ABOVE EXCEPT THAT THE
                ; CHARACTER ADDRESS IS DECREMENTED AFTER EACH FETCH
                ; AND STORE
                ;
05AF 1A        RMOV:   LDAX    D       ;FETCH CHARACTER
05B0 1B                DCX     D       ;DECREMENT FETCH ADDRESS
05B1 B9                CMP     C       ;TERMINATION CHARACTER
05B2 C8                RZ
05B3 77                MOV     M,A     ;STORE CHARACTER
05B4 2B                DCX     H       ;DECREMENT STORE ADDRESS
05B5 C3AF05            JMP     RMOV
                ;
                ;
                ; THIS ROUTINE IS USED TO LOAD FOUR CHARACTERS FROM
                ; MEMORY INTO REGISTERS
                ;
05B8 46        LODM:   MOV     B,M     ;FETCH CHARACTER
05B9 23                INX     H
05BA 4E                MOV     C,M     ;FETCH CHARACTER
05BB 23                INX     H
05BC 56                MOV     D,M     ;FETCH CHARACTER
05BD 23                INX     H
05BE 5E                MOV     E,M     ;FETCE CHARACTER
05BF C9                RET
                ;
                ;
                ; THIS ROUTINE STORES FOUR CHARACTERS FROM THE REGISTERS
                ; INTO MEMORY
                ;
05C0 73        STOM:   MOV     M,E     ;STORE CHARACTER
05C1 2B                DCX     H
05C2 72                MOV     M,D     ;STORE CHARACTER
05C3 2B                DCX     H
05C4 71                MOV     M,C     ;STORE CHARACTER
05C5 2B                DCX     H
05C6 70                MOV     M,B     ;STORE CHARACTER
05C7 C9                RET
                ;
                ;
                ; THIS ROUTINE IS USED TO COMPARE TWO CHARACTER STRINGS
                ; OF LENGTH 4  ON RETURN ZERO FLAG SET MEANS BOTH
                ; STRINGS ARE EQUAL.  CARRY FLAG =0 MEANS STRING ADDRESSED
                ; BY D E WAS GREATER THAN OR EQUAL TO CHARACTER STRING
                ; ADDRESSED BY H,L
                ;
05C8 0601      COM0:   MVI     B,1     ;EQUAL COUNTER
05CA 0E04              MVI     C,4     ;STRING LENGTH
05CC B7                ORA     A       ;CLEAR CARRY
05CD 1A        CO1:    LDAX    D       ;FETCH CHARACTER
```

```
05CE 9E                 SBB     M       ;COMPARE CHARACTERS
05CF CAD305             JZ      CO2
05D2 04                 INR     B       ;INCREMENT EQUAL COUNTER
05D3 1B         CO2:    DCX     D
05D4 2B                 DCX     H
05D5 0D                 DCR     C
05D6 C2CD05             JNZ     CO1
05D9 05                 DCR     B
05DA C9                 RET
                ;
                ; THIS ROUTINE IS SIMILAR TO THE ABOVE ROUTINE EXCEPT ON
                ; RETURN CARRY FLAG = 0 MEANS THAT CHARACTER STRING
                ; ADDRESSED BY D,E IS ONLY > STRING ADDRESSED BY H L
                ;
05DB 0E04       COM1:   MVI     C 4     ;STRING LENGTH
05DD 1A                 LDAX    D       ;TCH CHARACTER
05DE D601               SUI     1
05E0 C3CE05             JMP     CO1+1
                ;
                ;
                ; THIS ROUTINE WILL TAKE ASCII CHARACTERS AND ADD ANY
                ; NECESSARY ASCII ZEROES SO THE RESULT IS A 4 CHARACTER
                ; ASCII VALUE
                ;
05E3 CDB805     NORM:   CALL    LODM    ;LOAD CHARACTERS
05E6 AF                 XRA     A       ;FETCH A ZERO
05E7 B8                 CMP     B
05E8 C8                 RZ
05E9 BB         NOR1:   CMP     E
05EA C4C005             CNZ     STOM    ;STORE VALUES
05ED C0                 RNZ
05EE 5A                 MOV     E,D     ;NORMALIZE VALUE
05FF 51                 MOV     D,C
05F0 48                 MOV     C B
05F1 0E30               MVI     B,'0'
05F3 C3E905             JMP     NOR1
                ;
                ;
                ; THIS ROUTINE IS USED TO LIST THE CONTENTS OF THE FILE
                ; AREA STARTING AT THE LINE NUMBER GIVEN IN THE COMMAND
                ;
05F6 CD3A01     LIST:   CALL    CRLF
05F9 CD7205             CALL    FIND    ;FIND STARTING LN
05FC 23         LIST0:  INX     H       ;OUTPUT LINE.
05FD CDA002             CALL    SCRN
0600 CD3A01             CALL    CRLF
0603 CD9905             CALL    EOF     ;CHECK FOR END OF FILE
0606 CDED00             CALL    INK     ;CHECK FOR CTRL-X
0609 C2FC05             JNZ     LIST0   ;LOOP IF NO CTRL-X
060C C9                 RET
                ;
                ;
                ; THIS ROUTINE IS USED TO DELETE LINES FROM THE
                ; FILE AREA.  THE REMAINING FILE AREA IS THEN MOVED IN
                ; MEMORY SO THAT THERE IS NO EXCESS SPACE.
                ;
060D CD2703     DELL:   CALL    VCHK    ;CHECK FOR PARAMETER
0610 CD7205             CALL    FIND    ;FIND LINE IN FILE AREA
```

```
0613 227210             SHLD    DELP    ;SAVE DELETE POSITION
0616 218510             LXI     H ABUF-7
0619 7E                 MOV     A,M     ;CHECK FOR 2ND PARAMETER
061A B7                 ORA     A       ;SET FLAGS
061B C22106             JNZ     DEL1
061E 218110             LXI     H ABUF-3        ;USE FIRST PARAMETER
0621 227410     DEL1:   SHLD    ADDS    ;SAVE FIND ADDRESS
0624 EB                 XCHG
0625 213010             LXI     H MAXL-3
0628 CDC805             CALL    COM0    ;COMPARE LINE NUMBERS
062B 2A7210             LHLD    DELP    ;LOAD DELETE POSITION
062E DA6F06             JC      NOVR
                ; GET HERE IF DELETION INVOLVES END OF FILE
0631 222B10             SHLD    EOFP    ;CHANGE E-O-F POSITION
0634 3601               MVI     M,1     ;SET E-O-F INDICATOR
0636 EB                 XCHG
0637 2A2910             LHLD    BOFP    ;GET BEGIN FILE ADDRESS
063A EB                 XCHG
063B 060D               MVI     B,13    ;SET SCAN SWITCH
063D 2B                 DCX     H       ;CHECK FOR BOF
063E 7D         DEL2:   MOV     A,L
063F 93                 SUB     E
0640 7C                 MOV     A,H
0641 9A                 SBB     D
0642 3E0D               MVI     A,ASCR  ;LOOK FOR CR
0644 DA6606             JC      DEL4    ;DECREMENTED PAST BOF
0647 05                 DCR     B
0648 2B                 DCX     H
0649 BE                 CMP     M       ;FIND NEW MAX LN
064A C23E06             JNZ     DEL2
064D 2B                 DCX     H
064E 7D                 MOV     A,L
064F 93                 SUB     E
0650 7C                 MOV     A,H
0651 9A                 SBB     D
0652 DA6706             JC      DEL5
0655 BE                 CMP     M       ;END OF PREVIOUS LINE
0656 23                 INX     H
0657 23                 INX     H
0658 CA5C06             JZ      DEL3
065B 23                 INX     H
065C CDB805     DEL3:   CALL    LODM    ;LOAD NEW MAX LN
065F 213010             LXI     H,MAXL+3        ;SET ADDRESS
0662 CDC005             CALL    STOM    ;STORE NEW MAX LN
0665 C9                 RET
0666 B8         DEL4:   CMP     B       ;CHECK SWITCH
0667 EB         DEL5:   XCHG
0668 C25B06             JNZ     DEL3-1
066B 322D10             STA     MAXL    ;MAKE MAX LN A SMALL NUMBER
066E C9                 RET
                ; GET HERE IF DELETION IS IN MIDDLE OF FILE AREA
066F CD8005     NOVR:   CALL    FI1     ;FIND END OF DELETE AREA
0672 CC9205             CZ      FI2     ;NEXT LINE IF THIS LN EQUAL
0675 EB         NOV1:   XCHG
0676 2A7210             LHLD    DELP    ;CHAR MOVE TO POSITION
0679 0E01               MVI     C,1     ;MOVE TERMINATOR
067B CDA605             CALL    LMOV    ;COMPACT FILE AREA
067F 222B10             SHLD    EOFP    ;SET EOF POSITION
```

```
0681 3601               MVI     M 1     ;SET EOF INDICATOR
0683 C9                 RET
                ;
                ; STARTING HERE IS THE SELF ASSEMBLER PROGRAM
                ; THIS PROGRAM ASSEMBLES PROGRAMS WHICH ARE
                ; IN THE FILE AREA
                ;
0684 CD2703     ASSM:   CALL    VCHK    ;CHECK FOR PARAMETER
0687 3A8210             LDA     ABUF-4  ;GET 2ND PARAMETER
068A B7                 ORA     A       ;CHECK FOR PARAMETERS
068B C29406             JNZ     ASM4
068E 2A8A10             LHLD    BBUF    ;FETCH 1ST PARAMETER
0691 228C10             SHLD    BBUF-2  ;STORE INTO 2ND PARAMETER
0694 3ACA10     ASM4:   LDA     IBUF-4  ;FETCH INPUT CHARACTER
0697 D645               SUI     'E'     ;RESET A IF ERRORS ONLY
0699 328E10             STA     AERR    ;SAVE ERROR FLAG
069C 3EFF               MVI     A,0FFH  ;PUT MARKER IN SYMBOL TABLE...
069E 321A11             STA     SYMT
06A1 AF                 XRA     A       ;GET A ZERO
06A2 329410     ASM3:   STA     PASI    ;SET PASS INDICATOR
06A5 CD3A01             CALL    CRLF    ;INDICATE START OF PASS
06A8 2A8A10             LHLD    BBUF    ;FETCH ORIGIN
06AB 229210             SHLD    ASPC    ;INITIALIZE PC
06AE 2A2910             LHLD    BOFP    ;GET START OF FILE
06B1 227210             SHLD    APNT
06B4 2A7210     ASM1:   LHLD    APNT    ;FETCH LINE POINTER
06B7 31B110             LXI     SP,AREA+18
06BA 7E                 MOV     A,M     ;FETCH CHARACTER
06BB FE01               CPI     1       ;END OF FILE?
06BD CA2D09             JZ      PASS    ;JUMP IF END OF FILE
06C0 EB                 XCHG
06C1 13                 INX     D       ;INCREMENT ADDRESS
06C2 21B110             LXI     H,OBUF  ;BLANK START ADDRESS
06C5 3EC1               MVI     A,IBUF-5 AND 0FFH ;BLANK END ADDRESS
06C7 CDE500             CALL    CLER    ;BLANK OUT BUFFER
06CA 0E0D               MVI     C,ASCR  ;STOP CHARACTER
06CC CDA605             CALL    LMOV    ;MOVE LINE INTO BUFFER
06CF 71                 MOV     M,C     ;PLACE CR IN BUFFER
06D0 EB                 XCHG
06D1 227210             SHLD    APNT    ;SAVE ADDRESS
06D4 3A9410             LDA     PASI    ;FETCH PASS INDICATOR
06D7 B7                 ORA     A       ;SET FLAGS
06D8 C2F106             JNZ     ASM2    ;JUMP IF PASS 2
06DB CD0007             CALL    PAS1
06DE C3B406             JMP     ASM1
06E1 CDC907     ASM2:   CALL    PAS2
06E4 CDEA06             CALL    AOUT    ;OUTPUT LINE
06E7 C3B406             JMP     ASM1
                ;
                ; THIS ROUTINE IS USED TO OUTPUT THE LISTING FOR
                ; AN ASSEMBLY.  IT CHECKS THE ERROR SWITCH TO SEE IF
                ; ALL LINES ARE TO BE PRINTED OR JUST THOSE WITH
                ; ERRORS.
                ;
06EA 3A8E10     AOUT:   LDA     AERR    ;FETCH ERROR SWITCH
06ED B7                 ORA     A       ;SET FLAGS
06EE C2F706             JNZ     AOU1    ;OUTPUT ALL LINES
06F1 3AB110             LDA     OBUF    ;FETCH ERROR INDICATOR
```

```
06F4 FE20               CPI     ' '         ;CHECK FOR AN ERROR
06F6 C8                 RZ                  ;RETURN IF NO ERROR
06F7 21B110     ACU1:   LXI     H,OBUF      ;OUTPUT BUFFER ADDRESS
06FA CDA002             CALL    SCRN        ;OUTPUT LINE. .
06FD C33A01             JMP     CRLF
                ;
                ; PASS1 OF ASSEMBLER. USED TO FORM SYMBOL TABLE
                ;
0700 CD9301     PAS1:   CALL    ZBUF        ;CLEAR BUFFER
0703 329410             STA     PASI        ;SET FOR PASS1
0706 21C610             LXI     H,IBUF      ;INITIALIZE LINE POINTER
0709 CD6107             CALL    PATCH       ;CHECK FOR LABEL OR COMMENT
                ;
                ; PROCESS LABEL
                ;
070C CD500B             CALL    SLAB        ;GET AND CHECK LABEL
070F DA0F0B             JC      OP5         ;ERROR IN LABEL
0712 CAEA0C             JZ      ERRD        ;DUPLICATE LABEL
0715 CD4F07             CALL    LCHK        ;CHECK CHARACTER AFTER LABEL
0718 C20F0B             JNZ     OP5         ;ERROR IF NO BLANK
071B 0E05               MVI     C,LLAB      ;LENGTH OF LABELS
071D 217E10             LXI     H,ABUF      ;SET BUFFER ADDRESS
0720 7E         MLAB:   MOV     A,M         ;FETCH NEXT CHARACTER
0721 12                 STAX    D           ;STORE IN SYMBOL TABLE
0722 13                 INX     D
0723 23                 INX     H
0724 0D                 DCR     C
0725 C22007             JNZ     MLAB
0728 EB                 XCHG
0729 229010             SHLD    TABA        ;SAVE TABLE ADDRESS FOR EQU
072C 3A9310             LDA     ASPC-1      ;FETCH PC (HIGH
072F 77                 MOV     M,A
0730 23                 INX     H
0731 3A9210             LDA     ASPC        ;FETCH PC (LOW
0734 77                 MOV     M,A         ;STORE IN TABLE
0735 23                 INX     H           ;STORE END-OF-TABLE MARKER ..
0736 36FF               MVI     M,0FFH
                ;
                ; PROCESS OPCODE
                ;
0738 CD9301     OPC:    CALL    ZBUF        ;ZERO WORKING BUFFER
073B CD3909             CALL    SBLK        ;SCAN TO OPCODE
073E DA360B             JC      OERR        ;FOUND CARRIAGE RETURN
0741 CD9A0B             CALL    ALPS        ;PLACE OPCODE IN BUFFER
0744 FE20               CPI     ' '         ;CHECK FOR BLANK AFTER OPCODE
0746 DA950A             JC      OPCD        ;CR OR TAB AFTER OPCODE
0749 C2360B             JNZ     OERR        ;ERROR IF NO BLANK
074C C3950A             JMP     OPCD        ;CHECK OPCODE
                ;
                ; THIS ROUTINE CHECKS THE CHARACTER AFTER A LABEL
                ; FOR A BLANK OR A COLON.
                ;
074F 2A9610     LCHK:   LHLD    PNTR
0752 7E                 MOV     A,M         ;GET CHARACTER AFTER LABEL
0753 FE20               CPI     ' '         ;CHECK FOR A BLANK
0755 C8                 RZ                  ;RETURN IF A BLANK
0756 FE09               CPI     09H         ;CHECK FOR A TAB
0758 C8                 RZ                  ;RETURN IF A TAB
```

```
0759 FE3A               CPI     ':'     ;CHECK FOR A COLON
075B C0                 RNZ
075C 23                 INX     H
075D 229610             SHLD    PNTR    ;SAVE POINTER
0760 C9                 RET
                ;
                ; CHECK FOR LABELS OR COMMENTS
                ;
0761 33         PATCH:  INX     SP      ;BUMP SP PAST RETURN ADDRESS ..
0762 33                 INX     SP
0763 229610             SHLD    PNTR    ;SAVE POINTER
0766 7E                 MOV     A,M     ;FETCH CHARACTER
0767 FE20               CPI     ' '     ;CHECK FOR A BLANK
0769 CA3807             JZ      OPC     ;JUMP IF NO LABEL
076C FE09               CPI     09H     ;CHECK FOR A TAB
076E CA3807             JZ      OPC     ;JUMP IF NO LABEL
0771 FE2A               CPI     '*'     ;CHECK FOR COMMENT
0773 C8                 RZ              ;RETURN TO HIGHER LEVEL
0774 FE3B               CPI     ';'     ;ALSO CHECK FOR COMMENT...
0776 C8                 RZ
0777 3B                 DCX     SP      ;POINT SP AT IMMEDIATE RETURN...
0778 3B                 DCX     SP
0779 C9                 RET
                ;
                ; PROCESS ANY PSEUDO OPS THAT NEED TO BE IN PASS 1
                ;
077A CD3909     PSU1:   CALL    SBLK    ;SCAN TO OPERAND
077D 1A                 LDAX    D       ;FETCH VALUE
077E B7                 ORA     A       ;SET FLAGS
077F CA9607             JZ      ORG1    ;ORG OPCODE
0782 FAC607             JM      DAT1    ;DATA STATEMENT
0785 E2AB07             JPO     EQU1    ;EQU OPCODE
0788 FE05               CPI     5
078A DABE07             JC      RES1    ;RES OPCODE
078D C22D09             JNZ     PASS    ;JUMP IF END
                ; DO DW PSEUDO-OP
0790 0E02       ACO1:   MVI     C,2     ;2 BYTE INSTRUCTION
0792 AF                 XRA     A       ;GET A ZERO
0793 C3250F             JMP     OCN1    ;ADD VALUE TO PROGRAM CNTR
                ; DO ORG PSEUDO-OP
0796 CDBC0B     ORG1:   CALL    ASCN    ;GET OPERAND
0799 3AB110             LDA     OBUF    ;FETCH ERROR INDICATOR
079C FE20               CPI     ' '     ;CHECK FOR AN ERROR
079E C0                 RNZ
079F 229210             SHLD    ASPC    ;STORE NEW ORIGIN
07A2 3AC610             LDA     IBUF    ;GET FIRST CHARACTER
07A5 FE21               CPI     ' '+1   ;CHECK FOR LABEL
07A7 D8                 RC              ;NO LABEL
07A8 C3B607             JMP     EQUS    ;CHANGE LABEL VALUE
                ; DO EQU PSEUDO-OP
07AB CDBC0B     EQU1:   CALL    ASCN    ;GET OPERAND
07AE 3AC610             LDA     IBUF    ;FETCH 1ST CHARACTER
07B1 FE21               CPI     ' '-1   ;CHECK FOR LABEL
07B3 DAC30C             JC      ERRM    ;MISSING LABEL
07B6 EB         EQUS:   XCHG
07B7 2A9010             LHLD    TABA    ;SYMBOL TABLE ADDRESS
07BA 72                 MOV     M,D     ;STORE LABEL VALUE
07BB 23                 INX     H
```

```
07BC 73                 MOV     M,E
07BD C9                 RET
                ; DO DS PSEUDO-OP
07BE CDBC0B     RES1:   CALL    ASCN    ;GET OPERAND
07C1 44                 MOV     B,H
07C2 4D                 MOV     C,L
07C3 C31A08             JMP     RES21   ;ADD VALUE TO PROGRAM COUNTER
                ;
                ; DO DB PSEUDO-OP
                ;
07C6 C32109     DAT1:   JMP     DAT2A
                ;
                ; PERFORM PASS 2 OF THE ASSEMBLER
                ;
07C9 21E310     PAS2:   LXI     H,OBUF-2 ;SET OUTPUT BUFFER ADDRESS
07CC 3A9310             LDA     ASPC-1  ;FETCH PC HIGH
07CF CDAF02             CALL    BINB-3  ;CONVERT FOR OUTPUT
07D2 23                 INX     H
07D3 3A9210             LDA     ASPC    ;FETCH PC LOW
07D6 CDAF02             CALL    BINB-3  ;CONVERT FOR OUTPUT
07D9 23                 INX     H
07DA 229D10             SHLD    OIND    ;SAVE OUTPUT ADDRESS
07DD CD9301             CALL    ZBUF    ;CLEAR BUFFER
07E0 21C610             LXI     H,IBUF  ;INITIALIZE LINE POINTER
07E3 CD6107     PAS1:   CALL    PATCH
07E6 CD5003             CALL    SLAB    ;SCAN OFF LABEL
07E9 DAE50C             JC      ERRL    ;ERROR IN LABEL
07EC CD4F07             CALL    LCHK    ;CHECK FOR A BLANK OR COLON
07EF C2E50C             JNZ     ERRL    ;ERROR IF NOT A BLANK
07F2 C33807             JMP     OPC
                ;
                ;
                ;
                ; PROCESS PSEUDO OPS FOR PASS2
07F5 1A         PSU2:   LDAX    D
07F6 B7                 ORA     A       ;SET FLAGS
07F7 CA3908             JZ      ORG2    ;ORG OPCODE
07FA FA1E08             JM      DAT2    ;DATA OPCODE
07FD E22708             JPO     EQU2    ;EQUATE PSEUDE-OP
0800 FE05               CPI     5
0802 DA0E08             JC      RES2    ;RES OPCODE
0805 C22D09             JNZ     EASS    ;END OPCODE
                ; DO DW PSEUDO-OP
0808 CD0D09     ACC2:   CALL    TYS6    ;GET VALUE
080B C39007             JMP     ACC1
                ; DO DS PSEUDO-OP
080E CDB90B     RES2:   CALL    ASBL    ;GET OPERAND
0811 44                 MOV     B,H
0812 4D                 MOV     C,L
0813 2A8C10             LHLD    BBUF 2  ;FETCH STORAGE COUNTER
0816 09                 DAD     B       ;ADD VALUE
0817 228C10             SHLD    BBUF-2
081A AF         RES21:  XRA     A       ;GET A ZERO
081B C3280B             JMP     OCN2
                ; DO DB PSEUDO-OP
081E CDCC08     DAT2:   CALL    TYS5    ;GET OPERAND
0821 AF         DAT2A:  XRA     A       ;MAKE A ZERO
0822 0E01               MVI     C 1     ;BYTE COUNT
```

```
0824 C3250B             JMP     OCN1
                        ;
                        ; HANDLE EQUATES ON 2ND PASS.
                        ;
0827 CDB90B     EQU2:   CALL    ASBL    ;GET OPERAND INTO HL AND
                                        ;  FALL INTO NEXT ROUTINE
                        ;
                        ; STORE CONTENTS OF HL AS HEX ASCII AT OBUF 2.
                        ;   ON RETURN  DE HOLDS VALUE WHICH WAS IN HL
                        ;
082A EB         BINAD:  XCHG            ;PUT VALUE INTO DE
082B 21B310             LXI     H,OBUF-2 ;POINTER TO ADDR IN OBUF
082E 7A                 MOV     A,D     ;STORE HI BYTE....
082F CDAF02             CALL    BINH-3
0832 23                 INX     H
0833 7B                 MOV     A,E     ;STORE LO BYTE...
0834 CDAF02             CALL    BINH-3
0837 23                 INX     H
0838 C9                 RET
                ; DO ORG PSEUDO-OP
0839 CDB90B     ORG2:   CALL    ASBL    ;GET NEW ORIGIN
083C 3AB110             LDA     OBUF    ;GET ERROR INDICATOR
083F FE20               CPI     ' '     ;CHECK FOR AN ERROR
0841 C0                 RNZ             ;DON'T MODIFY PC IF ERROR
0842 CD2A08             CALL    BINAD   ;STORE NEW ADDR IN OBUF
0845 2A9210             LHLD    ASPC    ;FETCH PC
0848 EB                 XCHG
0849 229210             SHLD    ASPC    ;STORE NEW PC
084C 7D                 MOV     A,L
084D 93                 SUB     E       ;FORM DIFFERENCE OF ORIGINS
084E 5F                 MOV     E,A
084F 7C                 MOV     A,H
0850 9A                 SBB     D
0851 57                 MOV     D,A
0852 2A8C10             LHLD    BBUF-2  ;FETCH STORAGE POINTER
0855 19                 DAD     D       ;MODIFY
0856 228C10             SHLD    BBUF-2  ;SAVE
0859 C9                 RET
                        ;
                        ; PROCESS 1 BYTE INSTRUCTIONS WITHOUT OPERANDS
                        ;
085A C31A09     TYP1:   JMP     ASTO    ;STORE VALUE IN MEMORY AND RETURN
                        ;
                        ; PROCESS STAX AND LDAX INSTRUCTIONS
                        ;
085D CDB90B     TYP2:   CALL    ASBL    ;FETCH OPERAND
0860 C4A50C             CNZ     ERRR    ;ILLEGAL REGISTER
0863 7D                 MOV     A,L     ;GET LOW ORDER OPERAND
0864 B7                 ORA     A       ;SET FLAGS
0865 CA8108             JZ      TY31    ;OPERAND = 0
0868 FE02               CPI     2       ;OPERAND = 2
086A C4A50C             CNZ     ERRR    ;ILLEGAL REGISTER
086D C38108             JMP     TY31
                        ;
                        ; PROCESS PUSH,POP,INX,DCX,DAD INSTRUCTIONS
                        ;
0870 CDB90B     TYP3:   CALL    ASBL    ;FETCH OPERAND
0873 C4A50C             CNZ     ERRR    ;ILLEGAL REGISTER
```

```
0876 7D                 MOV     A L     ;GET LOW ORDER OPERAND
0877 0F                 RRC             ;CHECK LOW ORDER BIT
0878 DCA50C             CC      ERRR    ;ILLEGAL REGISTER
087B 17                 RAL             ;RESTORE
087C FE08               CPI     8
087E D4A50C             CNC     ERRR    ;ILLEGAL REGISTER
0881 07         TY31:   RLC             ;MULTIPLY BY 8
0882 17                 RAL
0883 17                 RAL
0884 47         TY32:   MOV     B,A
0885 1A                 LDAX    D       ;FETCH OPCODE BASE
0886 80                 ADD     B       ;FORM OPCODE
0887 FE76               CPI     118     ;CHECK FOR MOV M,M
0889 CCA50C             CZ      ERRR    ;ILLEGAL REGISTER
088C C35A08             JMP     TYP1
                ;
                ; PROCESS ACCUMULATOR  INR DCR,MOV.RST INSTRUCTIONS
                ;
088F CDB90B     TYP4:   CALL    ASBI    ;FETCH OPERAND
0892 C4A50C             CNZ     ERRR    ;ILLEGAL REGISTER
0895 7D                 MOV     A,L     ;GET LOW ORDER OPERAND
0896 FE08               CPI     8
0898 D4A50C             CNC     ERRR    ;ILLEGAL REGISTER
089B 1A                 LDAX    D       ;FETCH OPCODE BASE
089C FE40               CPI     64      ;CHECK FOR MOV INSTRUCTION
089E CAAD08             JZ      TY41
08A1 FEC7               CPI     199
08A3 7D                 MOV     A,L
08A4 CA8108             JZ      TY31    ;RST INSRUCTION
08A7 FA8408             JM      TY32    ;ACCUMULATOR INSTRUCTION
08AA C38108             JMP     TY31    ;INR,DCR
                ; PROCESS MOV INSTRUCTION
08AD 29         TY41:   DAD     H       ;MULTIPLY OPERAND BY 8
08AE 29                 DAD     H
08AF 29                 DAD     H
08B0 85                 ADD     L       ;FORM OPCODE
08B1 12                 STAX    D       ;SAVE OPCODE
08B2 CDEB08             CALL    MPNT
08B5 CDFC0B             CALL    ASCN
08B8 C4A50C             CNZ     ERRR    ;INCREMENT POINTER
08BB 7D                 MOV     A,L     ;FETCH LOW ORDER OPERAND
08BC FE08               CPI     8
08BE D4A50C             CNC     ERRR    ;ILLEGAL REGISTER
08C1 C38408             JMP     TY32
                ;
                ; PROCESS IMMEDIATE INSTRUCTIONS
                ; IMMEDIATE BYTE CAN BETWEEN -256 AND +255
                ; MVI INSTRUCTION IS A SPECIAL CASE AND CONTAINS
                ; 2 ARGUMENTS IN OPERAND
08C4 FE06       TYP5:   CPI     6       ;CHECK FOR MVI INSTRUCTION
08C6 CCD908             CZ      TY56
08C9 CD1A09             CALL    ASTO    ;STORE OBJECT BYTE
08CC CDB90B     TY55:   CALL    ASBL    ;GET IMMEDIATE ARGUMENT
08CF 3C                 INR     A
08D0 FE02               CPI     2       ;CHECK OPERAND FOR RANGE
08D2 D4BE0C             CNC     ERRV    ;OPERAND OUT OF RANGE
08D5 7D                 MOV     A,L
08D6 C35A08             JMP     TYP1
```

```
                        ;
                        ; FETCH 1ST ARG FOR MVI AND LXI INSTRUCTIONS
                        ;
08D9 CDE90B     TY56:   CALL    ASBL    ;FETCH ARG
08DC C4A50C             CNZ     ERRR    ;ILLEGAL REGISTER
08DF 7D                 MOV     A,L     ;GET LOW ORDER ARGUMENT
08E0 FE08               CPI     8
08E2 D4A50C             CNC     ERRR    ;ILLEGAL REGISTER
08E5 29                 DAD     H
08E6 29                 DAD     H
08E7 29                 DAD     H
08E8 1A                 LDAX    D       ;FETCH OPCODE BASE
08E9 85                 ADD     L       ;FOR OPCODE
08EA 5F                 MOV     E,A     ;SAVE OBJECT BYTE
08EB 2A9610     MPNT:   LHLD    PNTR    ;FETCH POINTER
08EE 7E                 MOV     A,M     ;FETCH CHARACTER
08EF FE2C               CPI     ','     ;CHECK FOR COMMA
08F1 23                 INX     H       ;INCREMENT POINTER
08F2 229610             SHLD    PNTR
08F5 C2AE0C             JNZ     ERR5    ;SYNTAX ERROR IF NO COMMA
08F8 7B                 MOV     A,E
08F9 C9                 RET
                        ;
                        ; PROCESS 3 BYTE INSTRUCTIONS
                        ; LXI INSTRUCTION IS A SPECIAL CASE
                        ;
08FA FE01       TYP6:   CPI     1       ;CHECK FOR LXI INSTRUCTION
08FC C20A09             JNZ     TY6     ;JUMP IF NOT LXI
08FF CDD908             CALL    TY56    ;GET REGISTER
0902 E608               ANI     08H     ;CHECK FOR ILLEGAL REGISTER
0904 C4A50C             CNZ     ERRR    ;REGISTER ERROR
0907 7B                 MOV     A,E     ;GET OPCODE
0908 E6F7               ANI     0F7H    ;CLEAR BIT IN ERROR
090A CD1A09     TY6:    CALL    ASTO    ;STORE OBJECT BYTE
090D CDB90B     TY56:   CALL    ASBL    ;FETCH OPERAND
0910 7D                 MOV     A,L
0911 54                 MOV     D,H
0912 CD1A09             CALL    ASTO    ;STORE 2ND BYTE
0915 7A                 MOV     A,D
0916 C35A09             JMP     TYP1
0919 C9                 RET
                        ;
                        ; THIS ROUTINE IS USED TO STORE OBJECT CODE PRODUCED
                        ; BY THE ASSEMBLER DURING PASS 2 INTO MEMORY
                        ;
091A 2A8C10     ASTO:   LHLD    BBUF+2  ;FETCH STORAGE ADDRESS
091D 77                 MOV     M,A     ;STORE OBJECT BYTE
091E 23                 INX     H       ;INCREMENT LOCATION
091F 228C10             SHLD    BBUF+2
0922 2A9D10             LHLD    OIND    ;FETCH OUTPUT ADDRESS
0925 23                 INX     H
0926 CDAF02             CALL    BINH+3  ;CONVERT OBJECT BYTE
0929 229D10             SHLD    OIND
092C C9                 RET
                        ;
                        ; GET HERE WHEN END PSEUDO-OP IS FOUND OR WHEN
                        ; END-OF-FILE OCCURS IN SOURCE FILE. CONTROL IS SET
                        ; FOR EITHER PASS 2 OR ASSEMBLY TERMINATOR IF FINISHED.
```

```
                        ;
092D 3A9410     PASS:   LDA     PASI    ;FETCH PASS INDICATOR
0930 B7                 ORA     A       ;SET FLAGS
0931 C26700             JNZ     EOR     ;JUMP IF FINISHED
0934 3E01               MVI     A,1     ;PASS INDICATOR FOR 2ND PASS
0936 C3A206             JMP     ASM3    ;DO 2ND PASS
                        ;
                        ; THIS ROUTINE SCANS THROUGH A CHARACTER STRING UNTIL
                        ; THE FIRST NON-BLANK CHARACTER IS FOUND
                        ;
                        ; ON RETURN  CARRY SET INDICATES A CARRIAGE RETURN
                        ; AS FIRST NON-BLANK CHARACTER.
                        ;
0939 2A9610     SBLK:   LHLD    PNTR    ;FETCH ADDRESS
093C 7E         SBL1:   MOV     A,M     ;FETCH CHARACTER
093D FE09               CPI     09H     ;IS IT A TAB?
093F CA4509             JZ      SBL2    ;TREAT LIKE A BLANK
0942 FE20               CPI     ' '     ;CHECK FOR A BLANK
0944 C0                 RNZ             ;RETURN IF NON-BLANK
0945 23         SBL2:   INX     H       ;INCREMENT
0946 229610             SHLD    PNTR    ;SAVE POINTER
0949 C33C09             JMP     SBL1
                        ;
                        ;
                        ; THIS ROUTINE IS USED TO CHECK THE CONDITION
                        ; CODE MNEMONICS FOR CONDITIONAL JUMPS. CALLS
                        ; AND RETURNS.
                        ;
094C 217F10     COND:   LXI     H,ABUF-1
094F 227410             SHLD    ADDS
0952 0602               MVI     B,2     ;2 CHARACTERS
0954 C3900A             JMP     COPC
                        ;
                        ;
                        ; THE FOLLOWING IS THE OPCODE TABLE
                        ;
                        ;
0957 4F5247     OTAB:   DB      'ORG'
095A 00                 DB      0
095B 00                 DB      0
095C 455155             DB      'EQU'
095F 00                 DB      0
0960 01                 DB      1
0961 4442               DB      'DB'
0963 00                 DB      0
0964 00                 DB      0
0965 FF                 DB      255
0966 4453               DB      'DS'
0968 00                 DB      0
0969 00                 DB      0
096A 03                 DB      3
096B 4457               DB      'DW'
096D 00                 DB      0
096E 00                 DB      0
096F 05                 DB      5
0970 454E44             DB      'END'
0973 00                 DB      0
0974 06                 DB      6
```

```
0975 00             DB      0
0976 484C54         DB      'HLT'
0979 76             DB      118
097A 524C43         DB      'RLC'
097D 07             DB      7
097F 525243         DB      'RRC'
0981 0F             DB      15
0982 52414C         DB      'RAL'
0985 17             DB      23
0986 524152         DB      'RAR'
0989 1F             DB      31
098A 524554         DB      'RET'
098D C9             DB      201
098E 434D41         DB      'CMA'
0991 2F             DB      47
0992 535443         DB      'STC'
0995 37             DB      55
0996 444141         DB      'DAA'
0999 27             DB      39
099A 434D43         DB      'CMC'
099D 3F             DB      63
099E 4549           DB      'EI'
09A0 00             DB      0
09A1 FB             DB      251
09A2 4449           DB      'DI'
09A4 00             DB      0
09A5 F3             DB      243
09A6 4E4F50         DB      'NOP'
09A9 00             DB      0
09AA 00             DB      0
09AB 58434847       DB      'XCHG'
09AF EB             DB      235
09B0 5854484C       DB      'XTHL'
09B4 E3             DB      227
09B5 5350484C       DB      'SPHL'
09B9 F9             DB      249
09BA 5043484C       DB      'PCHL'
09BE E9             DB      233
09BF 00             DB      0
09C0 53544158       DB      'STAX'
09C4 02             DB      2
09C5 4C444158       DB      'LDAX'
09C9 0A             DB      10
09CA 00             DB      0
09CB 50555348       DB      'PUSH'
09CF C5             DB      197
09D0 504F50         DB      'POP'
09D3 00             DB      0
09D4 C1             DB      193
09D5 494E58         DB      'INX'
09D8 00             DB      0
09D9 03             DB      3
09DA 444358         DB      'DCX'
09DD 00             DB      0
09DE 0B             DB      11
09DF 444144         DB      'DAD'
09E2 00             DB      0
09E3 09             DB      9
```

```
09E4 00             DB      0
09E5 494E52         DB      'INR'
09E8 04             DB      4
09E9 444352         DB      'DCR'
09EC 05             DB      5
09ED 4D4F56         DB      'MOV'
09F0 40             DB      64
09F1 414444         DB      'ADD'
09F4 80             DB      128
09F5 414443         DB      'ADC'
09F8 88             DB      136
09F9 535542         DB      'SUB'
09FC 90             DB      144
09FD 534242         DB      'SBB'
0A00 98             DB      152
0A01 414E41         DB      'ANA'
0A04 A0             DB      160
0A05 585241         DB      'XRA'
0A08 A8             DB      168
0A09 4F5241         DB      'ORA'
0A0C B0             DB      176
0A0D 434D50         DB      'CMP'
0A10 B8             DB      184
0A11 525354         DB      'RST'
0A14 C7             DB      199
0A15 00             DB      0
0A16 414449         DB      'ADI'
0A19 C6             DB      198
0A1A 414349         DB      'ACI'
0A1D CE             DB      206
0A1E 535549         DB      'SUI'
0A21 D6             DB      214
0A22 534249         DB      'SBI'
0A25 DE             DB      222
0A26 414E49         DB      'ANI'
0A29 E6             DB      230
0A2A 585249         DB      'XRI'
0A2D EE             DB      238
0A2E 4F5249         DB      'ORI'
0A31 F6             DB      246
0A32 435049         DB      'CPI'
0A35 FE             DB      254
0A36 494E           DB      'IN'
0A39 00             DB      0
0A39 DB             DB      219
0A3A 4F5554         DB      'OUT'
0A3D D3             DB      211
0A3E 4D5649         DB      'MVI'
0A41 06             DB      6
0A42 00             DB      0
0A43 4A4D50         DB      'JMP'
0A46 00             DB      0
0A47 C3             DB      195
0A48 43414C4C       DB      'CALL'
0A4C CD             DB      205
0A4D 4C5849         DB      'LXI'
0A50 00             DB      0
0A51 01             DB      1
```

```
0A52 4C4441             DB      'LDA'
0A55 00                 DB      0
0A56 3A                 DB      58
0A57 535441             DB      'STA'
0A5A 00                 DB      0
0A5B 32                 DB      50
0A5C 53484C44           DB      'SHLD'
0A60 22                 DB      34
0A61 4C484C44           DB      'LHLD'
0A65 2A                 DB      42
0A66 00                 DB      0
                ; CONDITION CODE TABLE
0A67 4E5A               DB      'NZ'
0A69 00                 DB      0
0A6A 5A                 DB      'Z'
0A6B 00                 DB      0
0A6C 08                 DB      8
0A6D 4E43               DB      'NC'
0A6F 10                 DB      16
0A70 43                 DB      'C'
0A71 00                 DB      0
0A72 18                 DB      24
0A73 504F               DB      'PO'
0A75 20                 DB      32
0A76 5045               DB      'PE'
0A78 28                 DB      40
0A79 50                 DB      'P'
0A7A 00                 DB      0
0A7B 30                 DB      48
0A7C 4D                 DB      'M'
0A7D 00                 DB      0
0A7E 38                 DB      56
0A7F 00                 DB      0
                ;
                ; THIS ROUTINE IS USED TO CHECK A GIVEN OPCODE
                ; AGAINST THE LEGAL OPCODES IN THE OPCODE TABLE
                ;
0A80 2A7410     COPC:   LHLD    ADDS
0A83 1A                 LDAX    D       ;FETCH CHARACTER
0A84 B7                 ORA     A       ;SET FLAGS
0A85 CA920A             JZ      COP1    ;JUMP IF TERMINATION CHARACTER
0A88 48                 MOV     C,B
0A89 CD8001             CALL    SEAR
0A8C 1A                 LDAX    D
0A8D C8                 RZ              ;RETURN IF MATCH
0A8E 13                 INX     D       ; NEXT STRING
0A8F C3800A             JMP     COPC    ;CONTINUE SEARCH
0A92 3C         COP1:   INR     A       ;CLEAR ZERO FLAG
0A93 13                 INX     D       ;INCREMENT ADDRESS
0A94 C9                 RET
                ;
                ; THIS ROUTINE CHECKS THE LEGAL OPCODES IN BOTH PASS 1
                ; AND PASS 2.  IN PASS 1 THE PROGRAM COUNTER IS INCRE-
                ; MENTED BY THE CORRECT NUMBER OF BYTES.  AN ADDRESS IS
                ; ALSO SET SO THAT AN INDEXED JUMP CAN BE MADE TO
                ; PROCESS THE OPCODE FOR PASS 2.
                ;
                ;
```

```
0A95 217E10    OPCD:   LXI     H.ABUF  ;GET ADDRESS
0A98 227410            SHLD    ADDS
0A9B 115709            LXI     D.OTAB  ;OPCODE TABLE ADDRESS
0A9E 0E04              MVI     B 4     ;CHARACTER COUNT
0AA0 CD800A            CALL    COPC    ;CHECK OPCODES
0AA3 CA3E0B            JZ      PSEU    ;JUMP IF A PSEUDO-OP
0AA6 05                DCR     B       ;3 CHARACTER OPCODES
0AA7 CD800A            CALL    COPC
0AAA CAB10A            JZ      OP1
0AAD 04                INR     B       ;4 CHARACTER  OPCODES
0AAE CD800A            CALL    COPC
0AB1 215A08    OP1:    LXI     H TYP1  ;TYPE 1 INSTRUCTIONS
0AB4 0E01      OP2:    MVI     C.1     ;1 BYTE INSTRUCTIONS
0AB6 CA110B            JZ      OCNT
               ;
0AB9 CD800A    OPC2:   CALL    COPC    ;CHECK FOR STAX LDAX
0ABC 215D08            LXI     H.TYP2
0ABF CAB40A            JZ      OP2
0AC2 CD800A            CALL    COPC    ;CHECK FOR PUSH.POP INX
               ;                       ;DCX AND DAD
0AC5 217008            LXI     H.TYP3
0AC8 CAB40A            JZ      OP2
0ACB 05                DCR     B       ;3 CHAR OPCODES
0ACC CD800A            CALL    COPC    ;ACCUMULATOR INSTRUCTIONS
               ;                       ;INR DCR.MOV.RST
0ACF 218F08            LXI     H.TYP4
0AD2 CAB40A            JZ      OP2
               ;
0AD5 CD800A    OPC3:   CALL    COPC    ;IMMEDIATE INSTRUCTIONS
0AD8 21C408            LXI     H TYP5
0ADB 0E02              MVI     C.2     ;2 BYTE INSTRUCTIONS
0ADD CA110B            JZ      OCNT
0AE0 04                INR     B       ;4 CHAR OPCODES
0AE1 CD800A            CALL    COPC    ;JMP CALL.LXI LDA STA
               ;                       ;LHLD,SHLD OPCODES
0AE4 CA0C0B            JZ      OP4
0AE7 CD4C09            CALL    COND    ;CONDITIONAL INSTRUCTIONS
0AEA C2360B            JNZ     OERR    ;ILLEGAL OPCODE
0AED C6C0              ADI     192     ;ADD BASE VALUE TO RETURN
0AEF 57                MOV     D.A
0AF0 0E03              MVI     B.3     ;3 CHARACTER OPCODES
0AF2 3A7E10            LDA     ABUF    ;FETCH FIRST CHARACTER
0AF5 4F                MOV     C A     ;SAVE CHARACTER
0AF6 FE52              CPI     'R'     ;CONDITIONAL RETURN
0AF8 7A                MOV     A,D
0AF9 CAB10A            JZ      OP1
0AFC 79                MOV     A,C
0AFD 14                INR     D       ;FORM CONDITIONAL JUMP
0AFE 14                INR     D
0AFF FE4A              CPI     'J'     ;CONDITIONAL JUMP
0B01 CA0B0B            JZ      OPAD
0B04 FE43              CPI     'C'     ;CONDITIONAL CALL
0B06 C2360B            JNZ     OERR    ;ILLEGAL OPCODE
0B09 14                INR     D       ;FORM CONDIITIONAL CALL
0B0A 14                INR     D
0B0B 7A        OPAD:   MOV     A D     ;GET OPCODE
0B0C 21FA08    OP4:    LXI     H.TYPE
0B0F 0E03      OP5:    MVI     C 3     ;3 BYTE INSTRUCTION
```

```
0B11 329C10     CCNT:   STA     TEMP    ;SAVE OPCODE
                ;
                ; CHECK FOR OPCODE ONLY CONTAINING THE CORRECT NUMBER OF
                ; CHARACTERS. THUS ADD0, SAY, WOULD GIVE AN ERROR
                ;
0B14 3E7E               MVI     A,ABUF AND 0FFH ;LOAD BUFFER ADDRESS
0B16 80                 ADD     B       ;ADD LENGTH OF OPCODE
0B17 5F                 MOV     E,A
0B18 3E10               MVI     A,ABUF/256
0B1A CE00               ACI     0       ;GET HIGH ORDER ADDRESS
0B1C 57                 MOV     D,A
0B1D 1A                 LDAX    D       ;FETCH CHARACTER AFTER OPCODE
0B1E B7                 ORA     A       ;IT SHOULD BE ZERO
0B1F C2360B             JNZ     OERR    ;OPCODE ERROR
0B22 3A9410             LDA     PASI    ;FETCH PASS INDICATOR
0B25 0600       OCN1:   MVI     B,0
0B27 EB                 XCHG
0B28 2A9210     OCN2:   LHLD    ASPC    ;FETCH PROGRAM COUNTER
0B2B 09                 DAD     B       ;ADD IN BYTE COUNT
0B2C 229210             SHLD    ASPC    ;STORE PC
0B2F B7                 ORA     A       ;WHICH PASS?
0B30 C8                 RZ              ;RETURN IF PASS 1
0B31 3A9C10             LDA     TEMP    ;FETCH OPCODE
0B34 EB                 XCHG
0B35 E9                 PCHL
                ;
0B36 21D00C     OERR:   LXI     H,ERRC  ;SET ERROR ADDRESS
0B39 0E03               MVI     C,3     ;LEAVE 3 BYTES FOR PATCH
0B3B C3220B             JMP     OCN1-3
                ;
0B3E 218210     PSEU:   LXI     H,ABUF-4        ;SET BUFFER ADDRESS
0B41 7E                 MOV     A,M     ;FETCH CHARACTER AFTER OPCODE
0B42 B7                 ORA     A       ;SHOULD BE A ZERO
0B43 C2360B             JNZ     OERR
0B46 3A9410             LDA     PASI    ;FETCH PASS INDICATOR
0B49 B7                 ORA     A
0B4A CA7A07             JZ      PSU1
0B4D C3F507             JMP     PSU2
                ;
                ;
                ; THIS ROUTINE IS USED TO PROCESS LABELS.
                ; IT CHECKS TO SEE IF A LABEL IS IN THE SYMBOL TABLE
                ; OR NOT.  ON RETURN  Z=1 INDICATES A MATCH WAS FOUND
                ; AND H L CONTAIN THE VALUE ASSOCIATED WITH THE LABEL.
                ; THE REGISTER NAMES A, B, C, D, E, H  L  P  AND S ARE
                ; PRE-DEFINED AND NEED NOT BE ENTERED BY THE USER
                ; ON RETURN, C=1 INDICATES A LABEL ERROR.
                ;
0B50 FE41       SLAB:   CPI     'A'     ;CHECK FOR LEGAL CHAR
0B52 D8                 RC
0B53 FE5B               CPI     'Z'+1   ;CHECK FOR ILLEGAL CHAR
0B55 3F                 CMC
0B56 D8                 RC              ;RETURN IF ILLEGAL CHAR
0B57 CD9A0B             CALL    ALPS    ;PLACE SYMBOL IN BUFFER
0B5A 217E10             LXI     H,ABUF  ;SET BUFFER ADDRESS
0B5D 227410             SHLD    ADIS    ;SAVE ADDRESS
0B60 05                 DCR     B       ;CHECK IF ONE CHARACTER
0B61 C2740B             JNZ     SLA1
```

```
                        ; CHECK IF PREDEFINED REGISTER NAME
0B64 04                         INR     B       ;SET B=1
0B65 11850B                     LXI     D,RTAB  ;REGISTER TABLE ADDRESS
0B68 CD800A                     CALL    COPC    ;CHECK NAME OF REGISTER
0B6B C2740B                     JNZ     SLA1    ;NOT A PREDEFINED REGISTER
0B6E 6F                         MOV     L,A     ;SET VALUE HIGH
0B6F 2600                       MVI     H,0
0B71 C3820B                     JMP     SLA2
0B74 111A11             SLA1:   LXI     D,SYMT  ;SET SYMBOL TABLE ADDRESS
0B77 3E05                       MVI     A,LLAB  ;FETCH LENGTH OF LABEL
0B79 329510                     STA     NCHR
0B7C CD6501                     CALL    COMS    ;CHECK TABLE
0B7F 4C                         MOV     C,H     ;SWAP H AND L
0B80 65                         MOV     H,L
0B81 69                         MOV     L,C
0B82 37                 SLA2:   STC             ;SET CARRY
0B83 3F                         CMC             ;CLEAR CARRY
0B84 C9                         RET             ;RETURN
                        ;
                        ; PREDEFINE REGISTER VALUES IN THIS TABLE
                        ;
0B85 41                 RTAB:   DB      'A'
0B86 07                         DB      7
0B87 42                         DB      'B'
0B88 00                         DB      0
0B89 43                         DB      'C'
0B8A 01                         DB      1
0B8B 44                         DB      'D'
0B8C 02                         DB      2
0B8D 45                         DB      'E'
0B8E 03                         DB      3
0B8F 48                         DB      'H'
0B90 04                         DB      4
0B91 4C                         DB      'L'
0B92 05                         DB      5
0B93 4D                         DB      'M'
0B94 06                         DB      6
0B95 50                         DB      'P'
0B96 06                         DB      6
0B97 53                         DB      'S'
0B98 06                         DB      6
0B99 00                         DB      0       ;END OF TABLE INDICATOR
                        ;
                        ; THIS ROUTINE SCANS THE INPUT LINE AND PLACES THE
                        ; OPCODES AND LABELS IN THE BUFFER.  THE SCAN TERMINATES
                        ; WHEN A CHARACTER OTHER THAN 0-9 OR A-Z IS FOUND.
                        ;
0B9A 0600               ALPS:   MVI     B,0     ;SET COUNT
0B9C 12                 ALP1:   STAX    D       ;STORE CHARACTER IN BUFFER
0B9D 04                         INR     B       ;INCREMENT COUNT
0B9E 78                         MOV     A,B     ;FETCH COUNT
0B9F FE0B                       CPI     11      ;MAXIMUM BUFFER SIZE
0BA1 D0                         RNC             ;RETURN IF BUFFER FILLED
0BA2 13                         INX     D       ;INCREMENT BUFFER
0BA3 23                         INX     H       ;INCREMENT INPUT POINTER
0BA4 229610                     SHLD    PNTR    ;SAVE LINE POINTER
0BA7 7E                         MOV     A,M     ;FETCH CHARACTER
0BA8 FE30                       CPI     '0'     ;CHECK FOR LEGAL CHARACTERS
```

```
0BAA D8                 RC
0BAB FE3A               CPI     '9'-1
0BAD DA9C0B             JC      ALP1
0BB0 FE41               CPI     'A'
0BB2 D8                 RC
0BB3 FE5B               CPI     'Z'+1
0BB5 DA9C0B             JC      ALP1
0BB8 C9                 RET
                ; THIS ROUTINE IS USED TO SCAN THROUGH THE INPUT LINE
                ; TO FETCH THE VALUE OF THE OPERAND FIELD   ON RETURN
                ; THE VALUE OF THE OPERAND IS CONTAINED IN REG'S H L
                ;
0BB9 CD3909     ASBL:   CALL    SBLK    ;GET FIRST ARGUMENT
0BBC 210000     ASCN:   LXI     H,0     ;GET A ZERO
0BBF 229910             SHLD    OPRD    ;INITIALIZE OPERAND
0BC2 24                 INR     H
0BC3 229A10             SHLD    OPRI-1  ;INITIALIZE OPERAND INDICATOR
0BC6 2A9610     NXT1:   LHLD    PNTR    ;FETCH SCAN POINTER
0BC9 2B                 DCX     H
0BCA CD9301             CALL    ZBUF    ;CLEAR BUFFER
0BCD 329910             STA     SIGN    ;ZERO SIGN INDICATOR
0BD0 23         NXT2:   INX     H       ;INCREMENT POINTER
0BD1 7E                 MOV     A,M     ;FETCH NEXT CHARACTER
0BD2 FE21               CPI     ' '+1
0BD4 DA780C             JC      SEND    ;JUMP IF CR OR BLANK
0BD7 FE2C               CPI     ','     ;FIELD SEPARATOR
0BD9 CA780C             JZ      SEND
                ; CHECK FOR OPERATORS
0BDC FE2B               CPI     '+'     ;CHECK FOR PLUS
0BDE CAE90B             JZ      ASC1
0BE1 FE2D               CPI     '-'     ;CHECK FOR MINUS
0BE3 C2F90B             JNZ     ASC2
0BE6 329910             STA     SIGN
0BE9 3A9B10     ASC1:   LDA     OPRI    ;FETCH OPERAND INDICATOR
0BEC FE02               CPI     2       ;CHECK FOR TWO OPERATORS
0BEE CAAE0C             JZ      ERRS    ;SYNTAX ERROR
0BF1 3E02               MVI     A,2
0BF3 329B10             STA     OPRI    ;SET INDICATOR
0BF6 C3D00B             JMP     NXT2
                ;CHECK FOR OPERANDS
0BF9 4F         ASC2:   MOV     C,A     ;SAVE CHARACTER
0BFA 3A9B10             LDA     OPRI    ;GET INDICATOR
0BFD B7                 ORA     A       ;CHECK FOR TWO OPERANDS
0BFE CAAE0C             JZ      ERRS    ;SYNTAX ERROR
0C01 79                 MOV     A,C
0C02 FE24               CPI     '$'     ;LC EXPRESSION
0C04 C2110C             JNZ     ASC3
0C07 23                 INX     H       ;INCREMENT POINTER
0C08 229610             SHLD    PNTR    ;SAVE POINTER
0C0B 2A9210             LHLD    ASPC    ;FETCH LOCATION COUNTER
0C0E C34D0C             JMP     AVAL
                ;CHECK FOR ASCII CHARACTERS
0C11 FE27       ASC3:   CPI     27H     ;CHECK FOR SINGLE QUOTE
0C13 C23D0C             JNZ     ASC5    ;JUMP IF NOT QUOTE
0C16 110000             LXI     D,0     ;GET A ZERO
0C19 0E03               MVI     C,3     ;CHARACTER COUNT
0C1B 23         ASC4:   INX     H       ;BUMP POINTER
0C1C 229610             SHLD    PNTR    ;SAVE
```

```
0C1F 7E                 MOV     A,M     ;FETCH NEXT CHARACTER
0C20 FE0D               CPI     ASCR    ;IS IT A CR?
0C22 CACB0C             JZ      ERRA    ;ARGUMENT ERROR
0C25 FE27               CPI     27H     ;IS IT QUOTE
0C27 C2340C             JNZ     SSTR
0C2A 23                 INX     H       ;INCREMENT POINTER
0C2B 229610             SHLD    PNTR    ;SAVE
0C2E 7E                 MOV     A,M     ;FETCH NEXT CHAR
0C2F FE27               CPI     27H     ;CHECK FOR 2 QUOTES IN A ROW
0C31 C24E0C             JNZ     AVAL+1  ;TERMINAL QUOTE
0C34 0D         SSTR:   DCR     C       ;CHECK COUNT
0C35 CACB0C             JZ      ERRA    ;TOO MANY CHARACTERS
0C38 53                 MOV     D,E
0C39 5F                 MOV     E,A     ;SET CHARACTER IN BUFFER
0C3A C31B0C             JMP     ASC4
0C3D FE30       ASC5:   CPI     '0'     ;CHECK FOR NUMERIC
0C3F DACB0C             JC      ERRA    ;ILLEGAL CHARACTER
0C42 FE3A               CPI     '9'+1
0C44 D26C0C             JNC     ALAB
0C47 CD880C             CALL    NUMS    ;GET NUMERIC VALUE
0C4A DACB0C             JC      ERRA    ;ARGUMENT ERROR
0C4D EB         AVAL:   XCHG
0C4E 2A9910             LHLD    OPRD    ;FETCH OPERAND
0C51 AF                 XRA     A       ;GET A ZERO
0C52 329B10             STA     OPRI    ;STOR IN OPERAND INDICATOR
0C55 3A9810             LDA     SIGN    ;GET SIGN INDICATOR
0C58 B7                 ORA     A       ;SET FLAGS
0C59 C2630C             JNZ     ASUB
0C5C 19                 DAD     D       ;FORM RESULT
0C5D 229910     ASC7:   SHLD    OPRD    ;SAVE RESULT
0C60 C3C60F             JMP     NXT1
0C63 7D         ASUB:   MOV     A,L
0C64 93                 SUB     E
0C65 6F                 MOV     L,A
0C66 7C                 MOV     A,H
0C67 9A                 SBB     D
0C68 67                 MOV     H,A
0C69 C35D0C             JMP     ASC7
0C6C CD500B     ALAB:   CALL    SLAB
0C6F CA4D0C             JZ      AVAL
0C72 DACB0C             JC      ERRA    ;ILLEGAL SYMBOL
0C75 C3B90C             JMP     ERRU    ;UNDEFINED SYMBOL
                ;
                ; GET HERE WHEN TERMINATING CHARACTER IS FOUND.
                ; CHECK FOR LEADING FIELD SEPARATOR.
                ;
0C78 3A9B10     SEND:   LDA     OPRI    ;FETCH OPERAND INDICATOR
0C7B B7                 ORA     A       ;SET FLAGS
0C7C C2AE0C             JNZ     ERRS    ;SYNTAX ERROR
0C7F 2A9910             LHLD    OPRD
0C82 7C         SEN1:   MOV     A,H     ;GET HIGH ORDER BYTE
0C83 119C10             LXI     D,TEMP  ;GET ADDRESS
0C86 B7                 ORA     A       ;SET FLAGS
0C87 C9                 RET
                ;
                ; GET A NUMERIC VALUE WHICH IS EITHER HEXADECIMAL OR
                ; DECIMAL   ON RETURN. CARRY SET INDICATES AN ERROR
                ;
```

```
0C88 CD9A0B     NUMS:   CALL    ALPS    ;GET NUMERIC
0C8B 1B                 DCX     D
0C8C 1A                 LDAX    D       ;GET LAST CHARACTER
0C8D 017E10             LXI     B,ABUF  ;SET BUFFER ADDRESS
0C90 FE48               CPI     'H'     ;IS IT HEXADECIMAL?
0C92 CAA00C             JZ      NUM2
0C95 FE44               CPI     'D'     ;IS IT DECIMAL
0C97 C29C0C             JNZ     NUM1
0C9A AF                 XRA     A       ;GET A ZERO
0C9B 12                 STAX    D       ;CLEAR D FROM BUFFER
0C9C CD2E02     NUM1:   CALL    ADEC    ;CONVERT DECIMAL VALUE
0C9F C9                 RET
0CA0 AF         NUM2:   XRA     A       ;GET A ZERO
0CA1 12                 STAX    D       ;CLEAR H FROM BUFFER
0CA2 C34802             JMP     AHEX
                ;
                ; PROCESS REGISTER ERROR
0CA5 3E52       ERRR:   MVI     A,'R'   ;GET INDICATOR
0CA7 210000             LXI     H 0     ;GET A 0
0CAA 32B110             STA     OBUF    ;SET IN OUTPUT BUFFER
0CAD C9                 RET
                ;PROCESS SYNTAX ERROR
0CAE 3E53       ERRS:   MVI     A,'S'   ;GET INDICATOR
0CB0 32B110             STA     OBUF    ;STORE IN OUTPUT BUFFER
0CB3 210000             LXI     H,0
0CB6 C3820C             JMP     SEN1
                ;PROCESS UNDEFINED SYMBOL ERROR
0CB9 3E55       ERRU:   MVI     A,'U'   ;GET INDICATOR
0CBB C3B00C             JMP     ERRS+2
                ;PROCESS VALUE ERROR
0CBE 3E56       ERRV:   MVI     A,'V'   ;GET INDICATOR
0CC0 C3A70C             JMP     ERRR-2
                ;PROCESS MISSING LABEL ERROR
0CC3 3E4D       ERRM:   MVI     A,'M'   ;GET INDICATOR
0CC5 32B110             STA     OBUF    ;STORE IN OUTPUT BUFFER
0CC8 C3F706             JMP     AOU1    ;DISPLAY ERROR AND RETURN
                ;PROCESS ARGUMENT ERROR
0CCB 3E41       ERRA:   MVI     A 'A'   ;GET INDICATOR
0CCD C3B00C             JMP     ERRS 2
                ; PROCESS OPCODE ERROR
                ; STORE 3 BYTES OF ZERO IN OBJECT CODE TO PROVIDE
                ; FOR A PATCH.
0CD0 3E4F       ERRO:   MVI     A,'O'   ;GET INDICATOR
0CD2 32B110             STA     OBUF    ;STORE IN OUTPUT BUFFER
0CD5 3A9410             LDA     PASI    ;FETCH PASS INDICATOR
0CD8 B7                 ORA     A       ;WHICH PASS
0CD9 C8                 RZ              ;RETURN IF PASS1
0CDA 0E03               MVI     C,3     ;NEED 3 BYTES
0CDC AF         ERO1:   XRA     A       ;GET A ZERO
0CDD CD1A09             CALL    ASTO    ;PUT IN LISTING AND MEMORY
0CE0 0D                 DCR     C
0CE1 C2DC0C             JNZ     ERO1
0CE4 C9                 RET
                ; PROCESS LABEL ERROR
0CE5 3E4C       ERRL:   MVI     A,'L'   ;GET INDICATOR
0CE7 C3D20C             JMP     ERRO+2
                ;PROCESS DUPLICATE LABEL ERROR
0CEA 3E44       ERRD:   MVI     A,'D'   ;GET ERROR INDICATOR
```

```
0CEC 32B110          STA     OBUF    ;STORE IN OUTPUT BUFFER
0CEF CDEA06          CALL    AOUT    ;DISPLAY ERROR
0CF2 C33807          JMP     OPC     ;PROCESS OPCODE
                     ;
                     ;
                     ; THIS ROUTINE SETS OR CLEARS BREAKPOINTS
                     ;
0CF5 3A7E10  BREAK:  LDA     ABUF    ;CHECK FOR AN ARG
0CF8 B7              ORA     A
0CF9 CA370D          JZ      CLRB    ;IF NO ARG, GO CLEAR BREAKPOINTS
0CFC 1608            MVI     D,NBR   ;ELSE  GET NUMBER OF BREAKPOINTS
0CFE 210C10          LXI     H,BRT   ;AND ADDR OF TABLE
0D01 7E      B1:     MOV     A,M     ;GET HI BYTE OF ENTRY
0D02 23              INX     H
0D03 46              MOV     B,M     ;GET LO BYTE OF ENTRY
0D04 B0              ORA     B       ;CHECK FOR EMPTY ENTRY
0D05 CA110D          JZ      B2      ;BRANCH IF EMPTY
0D08 23              INX     H       ;ELSE GO ON TO NEXT ENTRY
0D09 23              INX     H
0D0A 15              DCR     D       ;BUMP COUNT
0D0B C2010D          JNZ     B1      ;AND TRY AGAIN
0D0E C38104          JMP     WHAT    ;OOPS, NO ROOM
0D11 2B      B2:     DCX     H
0D12 EB              XCHG
0D13 2A8A10          LHLD    BBUF    ;GET ADDRESS
0D16 EB              XCHG            ;IN D,E
0D17 7A              MOV     A,D     ;CHECK FOR ADDR   11D
0D18 B7              ORA     A
0D19 C2220D          JNZ     B3
0D1C 7B              MOV     A,E
0D1D FE0B            CPI     11
0D1F DA8104          JC      WHAT    ;OOPS. TOO LOW
0D22 72      B3:     MOV     M,D     ;SAVE ADDRESS
0D23 23              INX     H
0D24 73              MOV     M,E
0D25 23              INX     H
0D26 1A              LDAX    D       ;PICK UP INSTRUCTION
0D27 77              MOV     M,A     ;SAVE IT
0D28 3ECF            MVI     A,0CFH  ;REPLACE IT WITH A
0D2A 12              STAX    D       ;RESTART INSTRUCTION
0D2B 3EC3            MVI     A,0C3H  ;SET UP LO MEMORY
0D2D 320800          STA     8       ;WITH A JUMP TO BRKP
0D30 21510D          LXI     H,BRKP
0D33 220900          SHLD    9
0D36 C9              RET             ;THEN RETURN
                     ;
                     ; THIS ROUTINE CLEARS ALL BREAKPOINTS
                     ;
0D37 210C10  CLRB:   LXI     H,BRT   ;GET TABLE ADDRESS
0D3A 0E08            MVI     C,NBR   ;GET NUMBER OF BREAKPOINTS
0D3C AF      CLRL:   XRA     A       ;GET A ZERO
0D3D 56              MOV     D,M     ;GET HI-BYTE OF ENTRY
0D3E 77              MOV     M,A
0D3F 23              INX     H
0D40 5E              MOV     E,M     ;GET LO-BYTE OF ENTRY
0D41 77              MOV     M,A
0D42 23              INX     H
0D43 46              MOV     B,M     ;GET INST BYTE
```

```
0D44 23                 INX     H
0D45 7A                 MOV     A,D     ;WAS THIS A NULL ENTRY
0D46 B3                 ORA     E
0D47 CA4C0D             JZ      CL2     ;BRANCH IF IT WAS
0D4A 78                 MOV     A B
0D4B 12                 STAX    D       ;ELSE  PLUG INST BACK IN
0D4C 0D         CL2:    DCR     C       ;BUMP COUNT
0D4D C23C0D             JNZ     CLBL    ;GO DO NEXT ONE
0D50 C9                 RET             ;RETURN WHEN DONE
                ;
                ; COME HERE WHEN WE HIT A BREAKPOINT
                ;
0D51 220910     BRKP:   SHLD    HOLD-8  ;SAVE H L
0D54 E1                 POP     H       ;GET PC
0D55 2B                 DCX     H       ;ADJUST IT
0D56 220A10             SHLD    HOLD-10 ;SAVE IT
0D59 F5                 PUSH    PSW     ;SAVE FLAGS
0D5A E1                 POP     H       ;GET THEM INTO HL
0D5B 220010             SHLD    HOLD    ;NOW STORE THEM FOR USER
0D5E 210000             LXI     H,0
0D61 39                 DAD     SP      ;GET STACK POINTER
0D62 310810             LXI     SP,HOLD-8 ;SET NEW SP
0D65 E5                 PUSH    H       ;SAVE OLD SP
0D66 D5                 PUSH    D       ;SAVE D E
0D67 C5                 PUSH    B       ;SAVE B C
0D68 2F                 CMA             ;COMPLEMENT ACC
0D69 D3FF               OUT     0FFH    ;DISPLAY IT IN THE LIGHTS
0D6B 31B110             LXI     SP AREA+18 ;SET SP AGAIN
0D6E 2A0A10             LHLD    HOLD-10 ;GET PC
0D71 EB                 XCHG            ;INTO D,E
0D72 210C10             LXI     H,BRT   ;GET ADDR OF TABLE
0D75 0608               MVI     B,NBR   ;AND NUMBER OF ENTRIES
0D77 7E         BL1:    MOV     A M     ;GET AN ENTRY FROM THE TABLE
0D78 23                 INX     H
0D79 BA                 CMP     D       ;DOES IT MATCH
0D7A C2820D             JNZ     BL2     ;BRANCH IF NOT
0D7D 7E                 MOV     A,M     ;ELSE GET NEXT BYTE
0D7E BB                 CMP     E       ;CHECK IT
0D7F CA8B0D             JZ      BL3     ;IT MATCHES!
0D82 23         BL2:    INX     H       ;BUMP AROUND THIS ENTRY
0D83 23                 INX     H
0D84 05                 DCR     B       ;BUMP COUNT
0D85 CA8104             JZ      WHAT    ;NOT IN OUR TABLE!
0D88 C3770D             JMP     BL1
                ;
0D8B 23         BL3:    INX     H
0D8C 7E                 MOV     A,M     ;GET INSTR BYTE
0D8D 12                 STAX    D       ;PUT IT BACK
0D8E AF                 XRA     A       ;CLEAR ENTRY IN TABLE
0D8F 2B                 DCX     H
0D90 77                 MOV     M A
0D91 2B                 DCX     H
0D92 77                 MOV     M,A
0D93 CD3A01             CALL    CRLF    ;RESTORE THE CARRIAGE
0D96 3A0E10             LDA     HOLD-11 ;GET HI-BYTE OF PC
0D99 CD6702             CALL    HOUT    ;TYPE IT
0D9C 3A0A10             LDA     HOLD-10 ;GET LO-BYTE OF PC
0D9F CD6702             CALL    HOUT    ;TYEP IT
```

```
0DA2 21A80D            LXI     H,BMES  ;TELL USER WHAT IT IS
0DA5 C38704            JMP     MESS    ;GO BACK TO COMMAND LEVEL
                       ;
0DA8 20425245414BMES:  DB      ' BREAK',13
                       ;
                       ; THIS ROUTINE PROCEEDS FROM A BREAKPOINT
                       ;
0DAF 3A7F10    PROC:   LDA     ABUF    ;CHECK FOR ARG
0DB2 B7                ORA     A
0DB3 CABC0D            JZ      P1      ;JMP IF NO ARG
0DB6 2A8A10            LHLD    BBUF    ;ELSE  GET ARG
0DB9 220A10            SHLD    HOLD+10 ;PLUG IT INTO PC SLOT
0DBC 310010    P1:     LXI     SP,HOLD ;SET SP TO POINT AT REG'S
0DBF F1                POP     PSW     ;RESTORE PSW
0DC0 C1                POP     B       ;RESTORE B,C
0DC1 D1                POP     D       ;RESTORE D,E
0DC2 E1                POP     H       ;GET OLD SP
0DC3 F9                SPHL            ;RESTORE IT
0DC4 2A0A10            LHLD    HOLD+10 ;GET PC
0DC7 E5                PUSH    H       ;PUT IT ON STACK
0DC8 2A0810            LHLD    HOLD+8  ;RESTORE H,L
0DCB C9                RET             ;AND PROCEED
                       ;
                       ; SYSTEM RAM
                       ;
1000                   ORG     1000H
                       ;
                       ; DEFINE BREAKPOINT REGION
                       ;
0008 =         NBR     EQU     8       ;NUMBER OF BREAKPOINTS
1000           HOLD:   DS      12      ;REGISTER HOLD AREA
100C           BRT:    DS      3*NBR   ;BREAKPOINT TABLE
                       ;
                       ; FILE AREA PARAMETERS
0006 =         MAXFIL  EQU     6       ;MAX # OF FILES
0005 =         NMLEN   EQU     5       ;NAME LENGTH
000D =         FELEN   EQU     NMLEN+8 ;DIRECTORY ENTRY LENGTH
1024           FILE0:  DS      NMLEN
1029           BOFP:   DS      2
102B           EOFP:   DS      2
102D           MAXL:   DS      4
1031           FILTB:  DS      (MAXFIL-1)*FELEN
1072           INSP:   DS      2       ;INSERT LINE POSITION
1072 =         DELP    EQU     INSP    ;DELETE LINE POSITION
000D =         ASCR    EQU     13      ;ASCII CARRIAGE RETURN VALUE
1074           HCON:   DS      2
1074 =         ADDS    EQU     HCON    ;FIND ADDRESS
1076           FBUF:   DS      NMLEN   ;FILE NAME BUFFER
107B           FREAD:  DS      2       ;FREE ADDRESS IN DIRECTORY
107D           FEF:    DS      1       ;FREE ENTRY FOUND FLAG
107D =         FOCNT   EQU     FEF     ;OUTPUT COUNTER
107E           ABUF:   DS      12      ;ASCII BUFFER
108A           BBUF:   DS      4       ;BINARY BUFFER
108E           SCNT:   DS      1
108F           DCNT:   DS      1       ;DUMP ROUTINE COUNTER
1090           TABA:   DS      2       ;SYMBOL TABLE END ADDRESS
1092           ASPC:   DS      2       ;ASSEMBLER PROGRAM COUNTER
1094           PASI:   DS      1       ;PASS INDICATOR
```

```
1095            NCHR:   DS      1       ;LENGTH OF STRING FOR COMPARE
1096            PNTR:   DS      2       ;LINE POINTER STORAGE
1098            SIGN:   DS      1       ;SIGN STORAGE FOR SCAN
1099            OPRD:   DS      2       ;OPERAND STORAGE
109B            OPRI:   DS      1       ;OPERAND FOUND INDICADOR
109C            TEMP:   DS      1
1072 =          APNT    EQU     INSP    ;ASSEMBLE LINE POINTER
109E =          AERR    EQU     SCNT    ;ASSEMBLER ERROR PRINT SWITCH
109D            OINT:   DS      2       ;OUTPUT ADDRESS
0005 =          LLAB    EQU     5       ;LENGTH OF LABELS
109F            AREA:   DS      18
10B1            OBUF:   DS      16      ;OUTPUT BUFFER AREA
10C1                    DS      5
10C6            IBUF:   DS      83
1119            LPOS:   DS      1       ;TELETYPE LINE POSITION
111A =          SYMT    EQU     $       ;START OF SYMBOL TABLE
                ;
                ; TELETYPE PARAMETERS
                ;
0003 =          TTS     EQU     3       ;TTY STATUS PORT
0002 =          TTI     EQU     2       ;TTY DATA IN PORT
0002 =          TTO     EQU     2       ;TTY DATA OUT PORT
0002 =          TTYDA   EQU     2       ;TTY DATA AVAILABLE BIT
0001 =          TTYTR   EQU     1       ;TTY XMTR READY BIT
00FF =          SWCH    EQU     0FFH    ;SWITCH REGISTER
                ;
111A                    END
```