

# DIGITAL RESEARCH

Post Office Box 579, Pacific Grove, California 93950, (408) 373-3403

CP/M ASSEMBLER (ASM)

CP/M VERSION \_\_\_\_\_

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

# ASM 1.0

```

; COMMON DATA FOR CP/M ASSEMBLER MODULE
0100   ORG      100H
20F0 = ENDA   EQU      20F0H ;END OF ASSEMBLER PROGRAM
0085 = BDOS   EQU      5H   ;ENTRY TO DOS, USED TO COMPUTE END MEMORY
0103 310002 LXI     SP,ENDMOD
0103 2A2600 LHLD    BDOS+1
0105 22CD01 SHLD   SYMAX ;COMPUTE END OF MEMORY
0109 C33002 JMP     ENDMOD

;
; PRINT BUFFER AND PRINT BUFFER POINTER
0078 = PBMAX EQU    120   ;MAX PRINT BUFFER
010C  PBUFF: DS    PBMAX
0134  PBP:   DS     1     ;PRINT BUFFER POINTER

;
; SCANNER PARAMETERS
0105  TOKEN: DS     1     ;CURRENT TOKEN
0106  VALUE: DS     2     ;BINARY VALUE FOR NUMBERS
0108  ACCLEN: DS    1     ;ACCUMULATOR LENGTH
0048 = ACMAX EQU    64   ;LENGTH OF ACCUMULATOR
0109  ACCUM: DS    ACMAX ;ACCUMULATOR (MUST FOLLOW ACCLEN)

;
; OPERAND EXPRESSION EVALUATOR PARAMETERS
0103  EVALUE: DS     2     ;VALUE OF EXPRESSION AFTER EVALUATION

;
; SYMBOL TABLE MODULE PARAMETERS
010B F020 SYTOP: DW    ENDA ;FIRST LOCATION AVAILABLE FOR SYMBOL TABLE
010D  SYMAX: DS     2     ;LAST AVAILABLE LOCATION FOR SYMBOL TABLE

;
; MISCELLANEOUS DATA AREAS
010F  PASS: DS     1     ;PASS # 0,1
0108  FPC:  DS     2     ;FILL ADDRESS FOR NEXT HEX RECORD
0102  ASPC: DS     2     ;ASSEMBLER'S PSEUDO PC
0104 F022 SYBAS: DW    ENDA ;SYMBOL TABLE BASE
0106  SYADR: DS     2     ;CURRENT SYMBOL BASE
0203 = ENDMOD EQU    ($ AND 0FF00H)+100H
0108  END

```

```

; I/O MODULE FOR CP/M ASSEMBLER
;
0200   ORG      200H
0000 = BOOT EQU    000H ;REBOOT LOCATION
; I/O MODULE ENTRY POINTS
0200 C3E00C JMP     INIT ;INITIALIZE, START ASSEMBLER
0203 C3A10D JMP     SETUP ;FILE SETUP
0206 C3CA0D JMP     GNC   ;GET NEXT CHARACTER
0209 C3340E JMP     PNC   ;PUT NEXT OUTPUT CHARACTER
020C C3AA0E JMP     PNB   ;PUT NEXT HEX BYTE
020F C3DE0E JMP     PCHAR ;PRINT CONSOLE CHARACTER
0212 C3BC0C JMP     PCON  ;PRINT CONSOLE BUFFER TO CRLF
0215 C3E00F JMP     WOBUFF ;WRITE OUTBUFFER
0218 C32F0F JMP     PERR  ;PLACE ERROR CHARACTER INTO PBUFF
021B C34C10 JMP     DHEX  ;PLACE HEX BYTE INTO OUTPUT BUFFER
021E C3190F JMP     EOR   ;END OF ASSEMBLY

; DATA FOR I/O MODULE
0221  BPC:  DS     2     ;BASE PC FOR CURRENT HEX RECORD
0223  DRL:  DS     1     ;HEX BUFFER LENGTH
0224  DBUFF: DS    16   ;HEX BUFFER

;
; DISK NAMES
0234  CDISK: DS     1     ;CURRENTLY SELECTED DISK
0235  ADISK: DS     1     ;.ASM DISK
0236  PDISK: DS     1     ;.PRN DISK
0237  HDISK: DS     1     ;.HEX DISK

;
; COMMON EQUATES
0078 = OPMAX EQU    120   ;MAX PRINT SIZE
010C = QBUFF EQU    100H  ;PRINT BUFFER
0104 = QBP EQU     QBUFF+OPMAX ;PRINT BUFFER POINTER

;
0105 = TOKEN EQU    QBP+1 ;CURRENT TOKEN UNDER SCAN
0106 = VALUE EQU   TOKEN+1 ;VALUE OF NUMBER IN BINARY
0108 = ACCLEN EQU  VALUE+2 ;ACCUMULATOR LENGTH
0040 = ACMAX EQU   64    ;MAX ACCUMULATOR LENGTH
0109 = ACCUM EQU   ACCLEN+1

;
0109 = EVALUE EQU   ACCUM+ACMAX ;VALUE FROM EXPRESSION ANALYSIS

;
010B = SYTOP EQU    EVALUE+2 ;CURRENT SYMBOL TOP
010D = SYMAX EQU    SYTOP+2  ;MAX ADDRESS+1

;
010F = PASS EQU     SYMAX+2 ;CURRENT PASS NUMBER
0108 = FPC EQU     PASS+1   ;FILL ADDRESS FOR DHEX ROUTINE
0102 = ASPC EQU    FPC+2   ;ASSEMBLER'S PSEUDO PC

```

CP/M VERSION 1.3  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # ASM 1.0

```

;
CR EQU 0DH ;CARRIAGE RETURN
000A = LF EQU 0AH ;LINE FEED
001A = EOP EQU 1AH ;END OF FILE MARK
;
;
; DOS ENTRY POINTS
0005 = BDOS EQU 5H ;DOS ENTRY POINT
0001 = READC EQU 1 ;READ CONSOLE DEVICE
0002 = WRITC EQU 2 ;WRITE CONSOLE DEVICE
0003 = REDYC EQU 11 ;CONSOLE CHARACTER READY
000E = SELECT EQU 14 ;SELECT DISK SPECIFIED BY REGISTER E
000F = OPENF EQU 15 ;OPEN FILE
0010 = CLOSF EQU 16 ;CLOSE FILE
0013 = DELEF EQU 19 ;DELETE FILE
0014 = READP EQU 20 ;READ FILE
0015 = WRITP EQU 21 ;WRITE FILE
0016 = MAKEP EQU 22 ;MAKE A FILE
0019 = CSEL EQU 25 ;RETURN CURRENTLY SELECTED DISK
001A = SETDM EQU 26 ;SET DMA ADDRESS
;
; FILE AND BUFFERING PARAMETERS
0008 = NSB EQU 8 ;NUMBER OF SOURCE BUFFERS
0006 = NPB EQU 6 ;NUMBER OF PRINT BUFFERS
0005 = NHB EQU 6 ;NUMBER OF HEX BUFFERS
;
0400 = SSIZE EQU NSB*128
0300 = PFSIZE EQU NPB*128
0300 = HSIZE EQU NHB*128
;
; FILE CONTROL BLOCKS
0238 SCB: DS 9 ;FILE NAME
0241 41534D DB 'ASM' ;FILE TYPE
0244 SCBR: DS 1 ;REEL NUMBER (ZEROED IN SETUP)
0245 DS 19 ;MISC AND DISK MAP
0258 SCBCR: DS 1 ;CURRENT RECORD (ZEROED IN SETUP)
;
0259 PCB: DS 9
0262 50524E00 DB 'PRN',0
0266 DS 19
0279 00 DB 0 ;RECORD TO WRITE NEXT
;
027A HCB: DS 9
0283 48455800 DB 'HEX',0
0287 DS 19
029A 00 DB 0
;
; POINTERS AND BUFFERS
029B 0004 SBP: DW SSIZE ;NEXT CHARACTER POSITION TO READ
029D SBUFF: DS SSIZE

```

CP/M VERSION \_\_\_\_\_

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

069D 0000  
069F

099F 0000  
09A1  
005C =  
0001 =  
0009 =  
0080 =

0CA1 213402  
0CA4 BE  
0CA5 C8  
0CA6 77  
0CA7 5F  
0CAB 0E0E  
0CAA CD0500  
0CAD C9

0CAE 23  
0CAF 7E  
0CB0 FE20  
0CB2 CAB80C  
0CB5 DE41  
0CB7 C9  
0CB8 3A3402  
0CBB C9

0CBC 7E  
0CB0 CD0E0E  
0CC0 7E  
0CC1 23  
0CC2 FE0D  
0CC4 C2BC0C  
0CC7 3E0A  
0CC9 CD0E0E  
0CCC C9

0CCD 115C00  
0CD0 0609  
0CD2 1A  
0CD3 FE3F  
0CD5 CAB80D  
0CD8 77  
0CD9 23

; PBP: DW 0  
PBUFF: DS PSIZE

; HBP: DW 0  
HBUFF: DS HSIZE  
; PCB EQU 5CH ;FILE CONTROL BLOCK ADDRESS  
; PNM EQU 1 ;POSITION OF FILE NAME  
; FLN EQU 9 ;FILE NAME LENGTH  
; BUFF EQU 80H ;INPUT DISK BUFFER ADDRESS

; SEL: ;SELECT DISK IN REG-A  
LXI H,CDISK  
CMP M ;SAME?  
RZ  
MOV M,A ;CHANGE CURRENT DISK  
MOV E,A  
MVI C,SELECT  
CALL BDOS  
RET

; SCNP: ;SCAN THE NEXT PARAMETER  
INX H  
MOV A,M  
CPI  
JZ SCNP0  
SBI 'A' ;NORMALIZE  
RET  
SCNP0: LDA CDISK  
RET

; PCON: ;PRINT MESSAGE AT H,L TO CONSOLE DEVICE  
MOV A,M  
CALL PCHAR  
MOV A,M  
INX H  
CPI CR  
JNZ PCON  
MVI A,LF  
CALL PCHAR  
RET

; FNAME: ;FILL NAME FROM DEFAULT FILE CONTROL BLOCK  
LXI D,FCB  
MVI B,FLN  
FNAME0: LDAX D ;GET NEXT FILE CHARACTER  
CPI '?'  
JZ FNERR ;FILE NAME ERROR  
MOV M,A ;STORE TO FILE CNTRL BLOCK  
INX H

CP/M VERSION \_\_\_\_\_

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

```

00DA 13      INX      D
00DB 05      DCR      B
00DC C2D20C  JNZ      FNAM0 ;FOR NEXT CHARACTER
00DF C9      RET

```

```

;
INIT: ;SET UP STACK AND FILES, START ASSEMBLER
      LXI      H,TITL
00E0 21A00F  CALL     PCON
00E3 C0C0C0  CALL     PCON
00E6 C33F0D  JMP      SET0

```

```

;
OPEN: ;OPEN FILE ADDRESSED BY D,E
      MVI      C,OPENF
00E9 0E0F  CALL     BDOS
00EB C03500  CPI      255
00EE FEFF  RNZ
00F0 C0      OPEN ERROR
00F1 21B90F  LXI      H,ERR0R
00F4 C0C0C0  CALL     PCON
00F7 C30000  JMP      BOOT

```

```

;
CLOSE: ;CLOSE FILE ADDRESSED BY D,E
      MVI      C,CLOSF
00FA 0E10  CALL     BDOS
00FC C03500  CPI      255
00FF FEFF  RNZ ;CLOSE OK
0101 C0      LXI      H,ERRCL
0102 212910  CALL     PCON
0105 C0C0C0  JMP      BOOT
0108 C30000

```

```

;
DELETE: ;DELETE FILE ADDRESSED BY D,E
      MVI      C,DELEF
010B 0E13  JMP      BDOS
010D C30500

```

```

;
MAKE: ;MAKE FILE ADDRESSED BY D,E
      MVI      C,MAKEF
0110 0E16  CALL     BDOS
0112 C03500  CPI      255
0115 FEFF  RNZ
0117 C0      MAKE ERROR
0118 21D00F  LXI      H,ERRMA
011B C03C0C  CALL     PCON
011E C30000  JMP      BOOT

```

```

;
SELA: LDA      ADISK
0221 3A3502  CALL     SEL
0224 CDA10C  RET
0227 C9

```

```

;
NPR: ;RETURN ZERO FLAG IF NO PRINT FILE
      LDA      PDISK
0228 3A3602  CPI      'Z'-'A'
022B FE19

```

```

022D C8      RZ
022E FE17  CPI      'X'-'A' ;CONSOLE:
0230 C9      RET

```

```

;
SELP: LDA      PDISK
0231 3A3602  CALL     SEL
0234 CDA10C  RET
0237 C9

```

```

;
SELH: LDA      HDISK
0238 3A3702  CALL     SEL
023B CDA10C  RET
023E C9

```

```

;
SET0: ;SET UP FILES FOR INPUT AND OUTPUT
      LDA      FCB ;GET FIRST CHARACTER
023F 3A5C00  CPI      FNERR ;MAY HAVE FORGOTTEN NAME
0242 FE20  JZ      FNERR ;FILE NAME ERROR
0244 CABB0D  MVI      C,CSEL ;CURRENT DISK?
0247 0E19  CALL     BDOS ;GET IT TO REG
0249 C08500  STA      CDISK
024C 323402

```

```

;
SCAN PARAMETERS
      LXI      H,FCB+FLN-1
024F 216400  CALL     SCNP
0252 CDAE0C  STA      ADISK
0255 323502  CALL     SCNP
0258 CDAE0C  STA      HDISK
025B 323702  CALL     SCNP
025E CDAE0C  STA      PDISK
0261 323602

```

```

;
      LXI      H,SCB ;ADDRESS SOURCE FILE CONTROL BLOCK
0264 213802  CALL     FNAME ;FILE NAME OBTAINED FROM DEFAULT FCB
0267 C0CD0C

```

```

;
      CALL     NPR ;Z OR X?
026A C0280D  JZ      NOPR
026D C0830D  LXI      H,PCB ;ADDRESS PRINT FILE CONTROL BLOCK
0270 215902  PUSH     H ;SAVE A COPY FOR OPEN
0273 E5      PUSH     H ;SAVE A COPY FOR DELETE
0274 E5      CALL     FNAME ;FILL PCB
0275 C0CD0C  CALL     SELP
0278 CD310D  POP      D ;FCB ADDRESS
027B D1      CALL     DELETE
027C C08B0D  POP      D ;FCB ADDRESS
027F D1      CALL     MAKE
0280 CD100D

```

```

;
NOPR: ;TEST FOR HEX FILE
      LDA      HDISK
0283 3A3702  CPI      'Z'-'A'
0286 FE19  JZ      NOHEX
0288 CA9E0D  LXI      H,HCB
028B 217A02  PUSH     H
028E E5

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

CPI/M VERSION \_\_\_\_\_

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

```
0D8P E5      PUSH  H
0D93 CDCD8C  CALL  FNAME
0D93 CD388D  CALL  SELH
0D96 D1      POP   D
0D97 CD888D  CALL  DELETE
0D9A D1      POP   D
0D9B CD188D  CALL  MAKE
```

```
;
; FILES SET UP, CALL ASSEMBLER
0D9E C30011  NOHEX: JMP  ENDMOD
```

```
;
; SETUP: ;SETUP INPUT FILE FOR SOURCE PROGRAM
0DA1 218804  LXI   H,SSIZE
0DA4 229802  SHLD  SBP ;CAUSE IMMEDIATE READ
0DA7 AF     XRA   A ;ZERO VALUE
0DAB 324402  STA   SCBR ;CLEAR REEL NUMBER
0DAB 325802  STA   SCBCR ;CLEAR CURRENT RECORD
0DAE 322302  STA   DBL ;CLEAR HEX BUFFER LENGTH
0DB1 CD218D  CALL  SELA
0DB4 113802  LXI   D,SCB
0DB7 CDE98C  CALL  OPEN
```

```
;
0DBA C9     RET
```

```
;
; FNERR: ;FILE NAME ERROR
0DBB 21E38F  LXI   H,ERRFN
0DBE CDB88C  CALL  PCON
0DC1 C38888  JMP   BOOT
```

```
;
; GCOMP: ;COMPARE D,E AGAINST H,L
0DC4 7A     MOV   A,D
0DC5 BC     CMP   B
0DC6 C8     RNZ
0DC7 7B     MOV   A,E
0DC8 8D     CMP   L
0DC9 C9     RET
```

```
;
; GNC: ;GET NEXT CHARACTER FROM SOURCE BUFFER
0DCA C5     PUSH  B
0DCB D5     PUSH  D
0DCC E5     PUSH  H ;ENVIRONMENT SAVED
0DCC 2A9382  LHLD  SBP
0DD3 118884  LXI   D,SSIZE
0DD3 CDC48D  CALL  GCOMP
0DD6 C2198E  JNZ   GNC2
```

```
;
; READ ANOTHER BUFFER
0DD9 CD218D  CALL  SELA
0DDC 218888  LXI   H,8
```

```
0DDF 229B02  SHLD  SBP
0DE2 8688  MVI   B,NS3 ;NUMBER OF SOURCE BUFFERS
0DE4 219D02  LXI   H,SBUFF
```

```
GNC0: ;READ 128 BYTES
      PUSH  B ;SAVE COUNT
      PUSH  H ;SAVE BUFFER ADDRESS
      MVI   C,READP
      LXI   D,SCB
      CALL  BDOS ;PERFORM THE READ
      POP   H ;RESTORE BUFFER ADDRESS
      POP   B ;RESTORE BUFFER COUNT
      ORA   A ;SET FLAGS
```

```
0DF2 C1
0DF3 87
0DF4 8E88
0DF6 C28D8E
```

```
;
; NORMAL READ OCCURRED
      LXI   D,BUFF ;SOURCE BUFFER ADDRESS
      MVI   C,128
```

```
MOV0: LDAX  D ;GET CHARACTER
      MOV   M,A ;STORE CHARACTER
```

```
0DF9 118800
0DFC 8E88
0DFE 1A
0DF7 77
0E00 13
0E01 23
0E02 8D
0E03 C2FE8D
```

```
;
; DCR  B
; JNZ  GNC0
; JMP  GNC2
;
; BUFFER LOADED, TRY NEXT BUFFER
```

```
0E06 05
0E07 C2E78D
0E0A C3198E
```

```
;
; DCR  B
; JNZ  GNC0
; JMP  GNC2
```

```
0E0D FE83
0E0F D2288E
0E12 361A
0E14 23
0E15 8D
0E16 C2128E
```

```
;
; GNC1: ;EOF OR ERROR
      CPI   3 ;ALLOW 0,1,2
      JNC  FRERR ;FILE READ ERROR
GNC2: MVI   M,EOF ;STORE AND END OF FILE CHARACTER
      INX  H
      DCR  C
      JNZ  GNCE ;FILL CURRENT BUFFER WITH EOF'S
```

```
0E19 119D02
0E1C 2A9B82
0E1F E5
0E20 23
0E21 229B02
0E24 E1
0E25 19
0E26 7E
0E27 E1
0E28 D1
0E29 C1
0E2A C9
```

```
;
; GNC2: ;GET CHARACTER TO ACCUMULATOR AND RETURN
      LXI   D,SBUFF
      LHLD  SBP
      PUSH  H ;SAVE CURRENT SBP
      INX  H ;READY FOR NEXT READ
      SHLD  SBP
      POP   H ;RESTORE PREVIOUS SBP
      DAD  D ;ABSOLUTE ADDRESS OF CHARACTER
      MOV   A,M ;GET IT
      POP  H
      POP  D
      POP  B
      RET
```

CPI/M VERSION \_\_\_\_\_

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

```

0E28 21FA0F ;FRERR: LXI H,ERRFR
0E2E CDBC0C CALL PCON ;PRINT READ ERROR MESSAGE
0E31 C30000 JMP BOOT

```

```

;
;PNC: ;SAME AT PNCF, BUT ENVIRONMENT IS SAVED FIRST
;      PUSH B
;      CHECK FOR CONSOLE OUTPUT / NO OUTPUT
0E34 C5 MOV B,A ;SAVE CHARACTER
0E35 47 LDA PDISK ;Z OR X?
0E36 3A3602 CPI 'Z'-'A' ;Z NO OUTPUT
0E39 FE19 JZ PNRET
0E3B CA510E ;

```

```

0E3E FE17 CPI 'X'-'A'
0E40 78 MOV A,B ;RECOVER CHAR FOR CON OUT
0E41 C24A0E JNZ PNGO
0E44 CDDE0E CALL PCHAR
0E47 C3510E JMP PNRET

```

```

;
;      NOT X OR Z, SO PRINT IT
0E4A D5 PNGO: PUSH D
0E4B E5 PUSH H
0E4C CD530E CALL PNCF
0E4F E1 POP H
0E50 D1 POP D
0E51 C1 PNRET: POP B
0E52 C9 RET

```

```

;
;PNCF: ;PRINT NEXT CHARACTER
0E53 2A9D06 LHLD PBP
0E56 EB XCHG
0E57 219F06 LXI H,PBUFF
0E5A 19 DAD D
0E5B 77 MOV M,A ;CHARACTER STORED AT PBP IN PBUFF
0E5C EB XCHG ;PBP TO H,L
0E5D 23 INX H ;POINT TO NEXT CHARACTER
0E5E 229D06 SHLD PBP ;REPLACE IT
0E61 EB XCHG
0E62 210003 LXI H,PSIZE
0E65 CDC40D CALL GCOMP ;AT END OF BUFFER?
0E68 C8 RNZ ;RETURN IF NOT

```

```

;
;OVERFLOW, WRITE BUFFER
0E69 CD310D CALL SELP
0E6C 210200 LXI H,0
0E6F 229D06 SHLD PBP
0E72 219F06 LXI H,PBUFF
0E75 115902 LXI D,PCB ;D,E ADDRESS FILE CONTROL BLOCK
0E78 0606 MVI B,NPB ;NUMBER OF BUFFERS TO B
;      (DROP THROUGH TO WBUFF)

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

0E7A 7E
0E7B FE1A
0E7D C8

```

```

0E7E C5
0E7F D5
0E80 0E80
0E82 118000

```

```

0E85 7E
0E86 12
0E87 23
0E88 13
0E89 0D
0E8A C2850E

```

```

0E8D D1
0E8E D5
0E8F E5
0E90 0E15
0E92 CD0500
0E95 E1
0E96 D1
0E97 C1
0E98 B7
0E99 C2A10E

```

```

0E9C 05
0E9D C8
0E9E C37A0E

```

```

0EA1 211110
0EA4 CDBC0C
0EA7 C3770F

```

```

0EAA C5
0EAB D5
0EAC E5
0EAD CDB40E
0EB0 E1
0EB1 D1
0EB2 C1

```

```

;WBUF: ;WRITE BUFFERS STARTING AT H,L FOR B BUFFERS
;      CHECK FOR EOF'S
;      MOV A,M
;      CPI EOP
;      RZ ;DON'T DO THE WRITE

```

```

;
;      PUSH B ;SAVE NUMBER OF BUFFERS
;      PUSH D ;SAVE PCB ADDRESS
;      MVI C,128 ;READY FOR MOVE
;      LXI D,BUFF
WBUF0: ;MOVE TO BUFFER
;      MOV A,M ;GET CHARACTER
;      STAX D ;PUT CHARACTER
;      INX H
;      INX D
;      DCR C
;      JNZ WBUF0

```

```

;
;      WRITE BUFFER
;      POP D ;RECOVER PCB ADDRESS
;      PUSH D ;SAVE IT AGAIN FOR LATER
;      PUSH H ;SAVE BUFFER ADDRESS
;      MVI C,WRITE ;DOS WRITE FUNCTION
;      CALL BDOS
;      POP H ;RECOVER BUFFER ADDRESS
;      POP D ;RECOVER PCB ADDRESS
;      POP B ;RECOVER BUFFER COUNT
;      ORA A ;SET ERROR RETURN FLAGS
;      JNZ FWERR

```

```

;
;      WRITE OK
;      DCR B
;      RZ ;RETURN IF NO MORE BUFFERS TO WRITE
;      JMP WBUF

```

```

;FWERR: ;ERROR IN WRITE
;      LXI H,ERRFW
;      CALL PCON ;ERROR MESSAGE OUT
;      JMP EORC ;TO CLOSE AND REBOOT

```

```

;
;PNB: ;PUT NEXT HEX BYTE
;      PUSH B
;      PUSH D
;      PUSH H
;      CALL PNBFF
;      POP H
;      POP D
;      POP B

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

02B3 C9          RET
;
; PNBPF: ;PUT NEXT BYTE
; (SIMILAR TO THE PNCPL SUBROUTINE)
02B4 2A9F09     LHLD  HBP
02B7 EB        XCHG
02B8 21A109     LXI   H,HBUFF
02BB 19        DAD   D
02BC 77        MOV   M,A ;CHARACTER STORED AT HBP IN HBUFF
02BD EB        XCHG
02BE 23        INX   H ;HBP INCREMENTED
02BF 229F09     SHLD  HBP
02C2 EB        XCHG ;BACK TO D,E
02C3 210223     LXI   H,HSIZE
02C6 CDC42D     CALL  GCOMP ;EQUAL?
02C9 C8        RNZ

;
; OVERFLOW, WRITE BUFFERS
02CA CD382D     CALL  SELH
02CD 210200     LXI   H,0
02D0 229F09     SHLD  HBP
02D3 21A109     LXI   H,HBUFF
02D6 117A02     LXI   D,HCB ;FILE CONTROL BLOCK FOR HEX FILE
02D9 0636     MVI   B,NHB
02DB C37A0E     JMP   WBUFF ;WRITE BUFFERS

;
; PCHAR: ;PRINT CHARACTER IN REGISTER A
02DE C5        PUSH  B
02DF D5        PUSH  D
02E0 E5        PUSH  H
02E1 0E02     MVI   C,WRITC
02E3 5F        MOV   E,A
02E4 CD0508     CALL  BDOS
02E7 E1        POP   H
02E8 D1        POP   D
02E9 C1        POP   B
02EA C9        RET

;
; WOCHAR: ;WRITE CHARACTER IN REG-A WITH REFLECT AT CONSOLE IF ERROR
02EB 4F        MOV   C,A ;SAVE THE CHAR
02EC CD340E     CALL  PNC ;PRINT CHAR
02EF 3A0C01     LDA   QBUFF
02F2 FE20     CPI
02F4 C8        RZ
;
; ERROR IN LINE
02F5 3A3602     LDA   PDISK
02F8 FE17     CPI   'X'-A
02FA C8        RZ ;ALREADY PRINTED IF 'X'

;
02FB 79        MOV   A,C ;RECOVER CHARACTER

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

02FC CDDE0E     CALL  PCHAR ;PRINT IT
02FF C9        RET

;
; WOBUFF: ;WRITE THE OUTPUT BUFFER TO THE PRINT FILE
0300 3A8401     LDA   CBP ;GET CHARACTER COUNT
0303 210C01     LXI   H,QBUFF ;BASE OF BUFFER
0306 B7        ORA   A ;ZERO COUNT?
0307 CA150F     JZ   WOB0
;
; NOT END, SAVE COUNT AND GET CHARACTER
030A 47        MOV   B,A ;SAVE COUNT
030B 7E        MOV   A,M
030C CDEB0E     CALL  WOCHAR ;WRITE CHARACTER
030F 23        INX   H ;ADDRESS NEXT CHARACTER OF BUFFER
0310 78        MOV   A,B ;GET COUNT
0311 3D        DCR   A
0312 C3060F     JMP   WOB0

;
; WOB0: ;END OF PRINT - ZERO CBP
0315 328401     STA   CBP
; FOLLOW BY CR LF
0318 3E0D     MVI   A,CR
031A CDEB0E     CALL  WOCHAR
031D 3E0A     MVI   A,LF
031F CDEB0E     CALL  WOCHAR
0322 210C01     LXI   H,QBUFF
0325 3E78     MVI   A,QBMAX ;READY TO BLANK OUT
0327 3620     MVI   M,
0329 23        INX   H
032A 3D        DCR   A
032B C2270F     JNZ   WOB2
032E C9        RET

;
; PERR: ;FILL QBUFF ERROR MESSAGE POSITION
032F 47        MOV   B,A ;SAVE CHARACTER
0330 210C01     LXI   H,QBUFF
0333 7E        MOV   A,M
0334 FE20     CPI
0336 C0        RNZ ;DON'T CHANGE IT IF ALREADY SET
0337 78        MOV   M,B ;STORE ERROR CHARACTER
0338 C9        RET

;
; EOR: ;END OF ASSEMBLER
0339 CD280D     CALL  NPR ;2 OR A?
033C CA4F0F     JZ   EOPR
;
; EOR2: FILL OUTPUT FILES WITH EOF'S
033F 2A9D06     LHLD  PBP
0342 7D        MOV   A,L
0343 B4        ORA   H ;VALUE ZERO?
0344 CA4F0F     JZ   EOPR

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

0F47 3E1A      MVI    A,EOP    ;CTL-Z IS END OF FILE
0F49 CD340E    CALL   PNC      ;PUT ENDFILES IN PRINT BUFFER
0F4C C33F0F    JMP    EOR2     ;EVENTUALLY BUFFER IS WRITTEN
;
EOPR: ;END OF PRINT FILE, CHECK HEX
      LDA    HDISK
      CPI    'Z'-'A'
      JZ     EORC
EOR0: ;WRITE TERMINATING RECORD INTO HEX FILE
      LLA    DBL    ;MAY BE ZERO ALREADY
      ORA    A
      CNZ    WHEX   ;WRITE HEX BUFFER IF NOT ZERO
      LHLD   FPC    ;GET CURRENT FPC AS LAST ADDRESS
      SHLD   BPC    ;RECORD LENGTH ZERO, BASE ADDRESS 0000
      CALL   WHEX   ;WRITE HEX BUFFER
;
; NOW CLEAR OUTPUT BUFFER FOR HEX FILE
EOR1: LHLD   HBP
      MOV    A,L
      ORA    H
      JZ     EORC
      MVI   A,EOP
      CALL  PNC
      JMP   EOR1
;
; CLOSE FILES AND TERMINATE
EORC: CALL   NPR
      JZ    EORPC
      CALL SELP
      LXI  D,PCB
      CALL CLOSE
EORPC: LDA    HDISK
      CPI    'Z'-'A'
      JZ     EORHC
      CALL  SELH
      LXI  D,HCB
      CALL  CLOSE
;
EORHC: LXI  H,ENDA
      CALL PCON
      JMP  EOOT
;
0FA0 43502F4D20TITL: DB 'CP/M ASSEMBLER - VER 1.0',CR
0FB9 4E4F20534FERR0P: DB 'NO SOURCE FILE PRESENT',CR
0FD0 4E4F204449ERRMA: DB 'NO DIRECTORY SPACE',CR
0FE3 534F555243ERRFN: DB 'SOURCE FILE NAME ERROR',CR
0FFA 534F555243ERRFR: DB 'SOURCE FILE READ ERROR',CR

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

1011 4F55545055ERRFW: DB 'OUTPUT FILE WRITE ERROR',CR
1029 43414E4E4FERRCL: DB 'CANNOT CLOSE FILES',CR
103C 454E44204FENDA: DB 'END OF ASSEMBLY',CR
;
DHEX: ;DATA TO HEX BUFFER (BYTE IN REG-A)
      PUSH  B
      MOV   B,A    ;HOLD CHARACTER FOR 'Z' TEST
      LDA   HDISK
      CPI   'Z'-'A'
      MOV   A,B    ;RECOVER CHARACTER
      JZ    DHRER
      PUSH D      ;ENVIRONMENT SAVED
      PUSH PSW   ;SAVE DATA BYTE
      LXI  H,DBL  ;CURRENT LENGTH
      MOV  A,M    ;TO ACCUM
      ORA  A      ;ZERO?
      JZ    DHEX3
;
; LENGTH NOT ZERO, MAY BE FULL BUFFER
      CPI    16
      JC    DHEX1 ;BR IF LESS THAN 16 BYTES
; BUFFER FULL, DUMP IT
      CALL  WHEX ;DBL = 0 UPON RETURN
      JMP   DHEX3 ;SET BPC AND DATA BYTE
;
DHEX1: ;PARTIAL BUFFER IN PROGRESS, CHECK FOR SEQUENTIAL BYTE LOAD
      LHLD  FPC
      XCHG
      LHLD  BPC ;BASE PC IN H,L
      MOV   C,A ;CURRENT LENGTH OF BUFFER
      MVI  B,0  ;IS IN B,C
      DAD  B    ;BPC+DBL TO H,L
      MOV  A,E  ;READY FOR COMPARE
      CMP  L    ;EQUAL?
      JNZ  DHEX2 ;BR IF NOT
      MOV  A,D  ;CHECK HO BYTE
      CMP  H
      JZ   DHEX4 ;BR IF SAME ADDRESS
;
DHEX2: ;NON SEQUENTIAL ADDRESS, DUMP AND CHANGE BASE ADDRESS
      CALL WHEX
DHEX3: ;SET NEW BASE
      LHLD FPC
      SHLD BPC
;
DHEX4: ;STORE DATA BYTE AND INC DBL
      LXI  H,DBL
      MOV  E,M ;LENGTH TO REG-E
      INR  M   ;DBL=DBL+1
      MVI D,0 ;HIGH ORDER ZERO FOR DOUBLE ADD

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```



```

1891 212402 LXI H,DBUFF
1894 19 D ;DBUFF+DBL TO H,L
1895 F1 POP PSW ;RESTORE DATA BYTE
1896 77 MOV M,A ;INTO DATA BUFFER
1897 D1 POP D
1898 C1 DHRET: POP B ;ENVIRONMENT RESTORED
1899 C9 RET

```

```

;
;WRC: ;WRITE CHARACTER WITH CHECK SUM IN D
      PUSH PSW
189A F5 RRC
189B 8F RRC
189C 8F RRC
189D 8F RRC
189E 8F RRC
189F E68F ANI 8FH
18A1 CD9A10 CALL HEXC ;OUTPUT HEX CHARACTER
18A4 F1 POP PSW ;RESTORE BYTE
18A5 F5 PUSH PSW ;SAVE A VERSION
18A6 E68F ANI 8FH
18A8 CD9A10 CALL HEXC ;WRITE LOW NIBBLE
18AB F1 POP PSW ;RESTORE BYTE
18AC 82 ADD D ;COMPUTE CHECKSUM
18AD 57 MOV D,A ;SAVE CS
18AE C9 RET

```

```

;
;HEXC: ;WRITE CHARACTER
      ADI 90H
18AF C690 DAA
18B1 27 ACI 40H
18B2 CE40 DAA
18B4 27 JMP PNB ;PUT BYTE
18B5 C3AA0E

```

```

;
;WHEX: ;WRITE CURRENT HEX BUFFER
      MVI A, ;RECORD HEADER
18B8 3E3A CALL PNB ;PUT BYTE
18BA CDAA0E LXI H,DBL ;RECORD LENGTH ADDRESS
18BD 212302 MOV E,M ;LENGTH TO REG-E
18C0 5E XRA A ;ZERO TO REG-A
18C1 AF MOV D,A ;CLEAR CHECKSUM
18C2 57 MOV M,A ;LENGTH IS ZEROED FOR NEXT WRITE
18C3 77 LHLD BPC ;BASE ADDRESS FOR RECORD
18C4 2A2102 MOV A,E ;LENGTH TO A
18C7 78 CALL WRC ;WRITE HEX VALUE
18C8 CD9A10 MOV A,H ;HIGH ORDER BASE ADDR
18CB 7C CALL WRC ;WRITE HO BYTE
18CC CD9A10 MOV A,L ;LOW ORDER BASE ADDR
18CF 7D CALL WRC ;WRITE LO BYTE
18D0 CD9A10 XRA A ;ZERO TO A
18D3 AF CALL WRC ;WRITE RECORD TYPE 00
18D4 CD9A10 MOV A,E ;CHECK FOR LENGTH 0
18D7 78

```

```

18D8 B7 ORA A
18D9 CA810 JZ WHEX1

```

```

;
; NON - ZERO, WRITE DATA BYTES
      LXI H,DBUFF
18DC 212402 WHEX0: MOV A,M ;GET BYTE
18DF 7E INX H
18E0 23 CALL WRC ;WRITE DATA BYTE
18E1 CD9A10 DCR E ;END OF BUFFER?
18E4 1D JNZ WHEX0
18E5 C2DF10

```

```

;
; END OF DATA BYTES, WRITE CHECK SUM
18E8 AF WHEX1: XRA A
18E9 92 SUB D ;COMPUTE CHECKSUM
18EA CD9A10 CALL WRC

```

```

;
; SEND CRLF AT END OF RECORD
18ED 3E0D MVI A,CR
18EF CDAA0E CALL PNB
18F2 3E0A MVI A,LF
18F4 CDAA0E CALL PNB
18F7 C9 RET

```

```

1100 = ENDMOD EQU ($ AND 0FFE0H)+20H
18F8 END

```

CP/M VER 1.01  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

1100  
1100 C30014  
1103 C33211  
1106 C3C011

```

ORG      1100H
JMP      ENDMOD ;END OF THIS MODULE
JMP      INITS  ;INITIALIZE THE SCANNER
JMP      SCAN   ;CALL THE SCANNER

```

```

;
;
; ENTRY POINTS IN I/O MODULE
0200 = IOMOD EQU 200H
0206 = GNCP EQU IOMOD+6H
0215 = WOBUFF EQU IOMOD+15H
0218 = PERR EQU IOMOD+18H

```

```

;
LASTC: DS 1 ;LAST CHAR SCANNED
NEXTC: DS 1 ;LOOK AHEAD CHAR
STYPE: DS 1 ;RADIX INDICATOR

```

```

; COMMON EQUATES
0078 = PBMAX EQU 120 ;MAX PRINT SIZE
010C = PBUFF EQU 10CH ;PRINT BUFFER
0184 = PBP EQU PBUFF+PBMAX ;PRINT BUFFER POINTER

```

```

;
0185 = TOKEN EQU PBP+1 ;CURRENT TOKEN UDER SCAN
0186 = VALUE EQU TOKEN+1 ;VALUE OF NUMBER IN BINARY
0188 = ACCLEN EQU VALUE+2 ;ACCUMULATOR LENGTH
0340 = ACMAX EQU 64 ;MAX ACCUMULATOR LENGTH
0189 = ACCUM EQU ACCLEN+1

```

```

;
01C9 = EVALUE EQU ACCUM+ACMAX ;VALUE FROM EXPRESSION ANALYSIS

```

```

;
01CB = SYTOP EQU EVALUE+2 ;CURRENT SYMBOL TOP
01CD = SYMAX EQU SYTOP+2 ;MAX ADDRESS+1

```

```

;
01CF = PASS EQU SYMAX+2 ;CURRENT PASS NUMBER
01D0 = FPC EQU PASS+1 ;FILL ADDRESS FOR NEXT HEX BYTE
01D2 = ASPC EQU FPC+2 ;ASSEMBLER'S PSEUDO PC

```

```

; GLOBAL EQUATES
0301 = IDEN EQU 1 ;IDENTIFIER
0302 = NUMB EQU 2 ;NUMBER
0303 = STRNG EQU 3 ;STRING
0304 = SPECL EQU 4 ;SPECIAL CHARACTER

```

```

;
0301 = PLABT EQU 0001B ;PROGRAM LABEL
0302 = DLABT EQU 0010B ;DATA LABEL
0304 = EQUT EQU 0100B ;EQUATE
0305 = SETT EQU 0101B ;SET
0306 = MACT EQU 0110B ;MACRO

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

0008 =  
000B =  
002C =

0032 =  
0038 =  
000A =  
0010 =  
000D =  
000A =  
001A =  
0009 =

110C CD0602  
110F F5  
1110 FE0D  
1112 CA3011  
1115 FE0A  
1117 CA3011

111A 3A8401  
111D FE78  
111F D23011

1122 5F  
1123 1600  
1125 3C  
1126 328401  
1129 210C01  
112C 19  
112D F1  
112E 77  
112F C9

1130 F1  
1131 C9

1132 C04911  
1135 320A11  
1138 328401  
113B 3E0A  
113D 323911  
1140 CD1502  
1143 3E10  
1145 328401

```

EXTT EQU 1000B ;EXTERNAL
REFT EQU 1011B ;REFER
GLBT EQU 1100B ;GLOBAL

```

```

;
BINV EQU 2
OCTV EQU 8
DECV EQU 10
HEXV EQU 16
CR EQU 0DH
LF EQU 0AH
EOF EQU 1AH
TAB EQU 09H ;TAB CHARACTER

```

```

;
; UTILITY SUBROUTINES
GNC: ;GET NEXT CHARACTER AND ECHO TO PRINT FILE

```

```

CALL GNCF
PUSH PSW
CPI CR
JZ GNC0
CPI LF ;IF LF THEN DUMP CURRENT BUFFER
JZ GNC0

```

```

;
; NOT A CR OR LF, PLACE INTO BUFFER IF THERE IS ENOUGH ROOM
LDA PBP
CPI PBMAX
JNC GNC0
ENOUGH ROOM, PLACE INTO BUFFER

```

```

MOV E,A
MVI D,0 ;DOUBLE PRECISION PBP IN D,E
INR A
STA PBP ;INCREMENTED PBP IN MEMORY
LXI H,PBUFF
DAD D ;PBUFF(PBP)
POP PSW
MOV M,A ;PBUFF(PBP) = CHAR
RET

```

```

GNC0: ;CHAR NOT PLACED INTO BUFFER
POP PSW
RET

```

```

;
INITS: ;INITIALIZE THE SCANNER

```

```

CALL ZERO
STA NEXTC ;CLEAR NEXT CHARACTER
STA PBP
MVI A,LF ;SET LAST CHAR TO LF
STA LASTC
CALL WOBUFF ;CLEAR BUFFER
MVI A,16 ;START OF PRINT LINE
STA PBP

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

1148 C9
1149 AF
114A 328501
114D 320B11
1152 C9
1151 218801
1154 7E
1155 FE40
1157 DA5F11
115A 3580
115C CD1E13
115F 5E
1160 1600
1162 34
1163 23
1164 19
1165 3A0A11
1168 77
1169 C9
116A 7E
116B FE24
116D C8
116E AF
116F 77
1170 C9
1171 3A0A11
1174 D630
1176 FE3A
1178 17
1179 E601
117B C9
117C CD7111
117F C8
1183 3A2A11
1183 D641
1185 FE86
1187 17
1188 E601
118A C9
  
```

```

;
ZERO: XRA A
      STA ACCLN
      STA STYPE
      RET

;
SAVER: ;STORE THE NEXT CHARACTER INTO THE ACCUMULATOR AND UPDATE ACCLN
      LXI H,ACCLN
      MOV A,M
      CPI ACMAX
      JC SAVI ;JUMP IF NOT UP TO LAST POSITION
      MVI M,0
      CALL ERRO
SAV1: MOV E,M ;D,E WILL HOLD INDEX
      MVI D,0
      INR M ;ACCLN INCREMENTED
      INX H ;ADDRESS ACCUMULATOR
      DAD D ;ADD INDEX TO ACCUMULATOR
      LDA NEXTC ;GET CHARACTER
      MOV M,A ;INTO ACCUMULATOR
      RET

;
TDOLL: ;TEST FOR DOLLAR SIGN, ASSUMING H,L ADDRESS NEXTC
      MOV A,M
      CPI 'S'
      RNZ
      XRA A ;TO GET A ZERO
      MOV M,A ;CLEARS NEXTC
      RET ;WITH ZERO FLAG SET

;
NUMERIC: ;CHECK NEXTC FOR NUMERIC, RETURN ZERO FLAG IF NOT NUMERIC
      LDA NEXTC
      SUI 0
      CPI 10
      CARRY RESET IF NUMERIC
      RAL
      ANI 1B ;ZERO IF NOT NUMERIC
      RET

;
HEX: ;RETURN ZERO FLAG IF NEXTC IS NOT HEXADECIMAL
      CALL NUMERIC
      RNZ ;RETURNS IF 0-9
      LDA NEXTC
      SUI 'A'
      CPI 6
      CARRY SET IF OUT OF RANGE
      RAL
      ANI 1B
      RET
  
```

```

118B 3A0A11
118E D641
1190 FE1A
1192 17
1193 E601
1195 C9
1196 CD8B11
1199 C8
119A CD7111
119D C9
119E 3A0A11
11A1 FE61
11A3 D8
11A4 FE7B
11A6 D8
11A7 E65F
11A9 320A11
11AC C9
11AD CD0C11
11B0 320A11
11B3 CD9E11
11B6 C9
11B7 FE0D
11B9 C8
11BA FE1A
11BC C8
11BD FE21
11BF C9
11C0 AF
11C1 328501
11C4 CD4911
11C7 3A0A11
11CA FE09
11CC CAF411
11CF FE3B
11D1 CAE111
  
```

```

;
LETTER: ;RETURN ZERO FLAG IF NEXTC IS NOT A LETTER
      LDA NEXTC
      SUI 'A'
      CPI 26
      RAL
      ANI 1B
      RET

;
ALNUM: ;RETURN ZERO FLAG IF NOT ALPHANUMERIC
      CALL LETTER
      RNZ
      CALL NUMERIC
      RET

;
TRANS: ;TRANSLATE TO UPPER CASE
      LDA NEXTC
      CPI 'A' OR 1100000B ;LOWER CASE A
      RC ;CARRY IF LESS THAN LOWER A
      CPI ('Z' OR 1100000B)+1 ;LOWER CASE Z
      RNC ;NO CARRY IF GREATER THAN LOWER A
      ANI 1011111B ;CONVERT TO UPPER CASE
      STA NEXTC
      RET

;
GNCN: ;GET CHARACTER AND STORE TO NEXTC
      CALL GNC
      STA NEXTC
      CALL TRANS ;TRANSLATE TO UPPER CASE
      RET

;
EOLT: ;END OF LINE TEST FOR COMMENT SCAN
      CPI CR
      RZ
      CPI EOF
      RZ
      CPI '!'
      CPI '!'
      RET

;
SCAN: ;FIND NEXT TOKEN IN INPUT STREAM
      XRA A
      STA TOKEN
      CALL ZERO

;
DEBL: DEBLANK
      LDA NEXTC
      CPI TAB ;TAB CHARACTER TREATED AS BLANK OUTSIDE STRING
      JZ DEB0
      CPI '!' ;MAY BE A COMMENT
      JZ DEB1 ;DEBLANK THROUGH COMMENT
  
```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

11D4 FE2A      CPI      * *      ;PROCESSOR TECH COMMENT
11D6 C2ED11   JNZ      DEB2      ;NOT *
11D9 3A8911   LDA      LASTC     ;
11DC FE2A      CPI      LF        ;LAST LINE FEED?
11DE C2ED11   JNZ      DEB2      ;NOT LF*
; COMMENT FOUND, REMOVE IT
DEB1: CALL    GNCN
;
11E1 CDAD11   CALL    EOLT      ;CR, EOF, OR I
11E4 CDB711   JZ      FINDL     ;HANDLE END OF LINE
11E7 CAF111   JMP     DEB1      ;OTHERWISE CONTINUE SCAN
11EA C3E111   ORI     .        ;MAY BE ZERO
11ED F620     ORI     .
11EF FE20     CPI
11F1 C2FA11   JNZ     FINDL
11F4 CDAD11   DEB0: CALL GNCN    ;GET NEXT AND STORE TO NEXTC
11F7 C3C711   JMP     DEBL

;
; LINE DEBLANKED, FIND TOKEN TYPE
FINDL: ;LOOK FOR LETTER, DECIMAL DIGIT, OR STRING QUOTE
CALL    LETTER
JZ      FIND0
MVI     A, IDEN
JMP     STOKEN

;
FIND0: CALL    NUMERIC
JZ      FIND1
MVI     A, NUMB
JMP     STOKEN

;
FIND1: LDA     NEXTC
CPI     ' '
JNZ     FIND2
XRA     A
STA     NEXTC ;DON'T STORE THE QUOTE
MVI     A, STRNG
JMP     STOKEN

;
FIND2: ;ASSUME IT IS A SPECIAL CHARACTER
CPI     LF
JNZ     FIND3
LF FOUND
LDA     PASS
ORA     A
CNZ     WOBUFF
LXI     H, PBUFF ;CLEAR ERROR CHAR ON BOTH PASSES
MVI     H,
MVI     A, 16
STA     PBP ;START NEW LINE
FIND3: MVI     A, SPECL

;
STOKEN: STA     TOKEN

```

```

11FA CD8B11
11FD CA8512
1203 3E01
1202 C33912

```

```

1205 CD7111
1208 CA1812
120B 3E02
120D C33912

```

```

1210 3A8A11
1213 FE27
1215 C22112
1218 AF
1219 320A11
121C 3E03
121E C33912

```

```

1221 FE0A
1223 C23712

```

```

1226 3ACF01
1229 B7
122A C41502
122D 210C01
1230 3620
1232 3E10
1234 328401
1237 3E04

```

```

1239 328501

```

```

123C 3A0A11
123F 320911
1242 B7
1243 C45111
1246 CDAD11
1249 3A8501
124C FE04
124E C8
124F FE03
1251 C49E11
1254 210A11
1257 3A8501

```

```

125A FE01
125C C26C12

```

```

125F CD6A11
1262 CA3C12
1265 CD9611
1268 C8

```

```

1269 C33C12

```

```

126C FE02
126E C20213

```

```

1271 CD6A11
1274 CA3C12
1277 CD7C11
127A C23C12

```

```

127D 3A0A11
1280 FE4F
1282 C8A12
1285 FE51
1287 C28F12

```

```

128A 3E08
128C C39612

```

```

128F FE48
1291 C2A012

```

```

;
;
; LOOP WHILE CURRENT ITEM IS ACCUMULATING
SCTOK: LDA     NEXTC
STA     LASTC ;SAVE LAST CHARACTER
ORA     A
CNZ     SAVER ;STORE CHARACTER INTO ACCUM IF NOT ZERO
CALL    GNCN ;GET NEXT TO NEXTC
LDA     TOKEN
CPI     SPECL
RZ
;RETURN IF SPECIAL CHARACTER
CPI     STRNG
CNZ     TRANS ;TRANSLATE TO UPPER CASE IF NOT IN STRING
LXI     H, NEXTC
LDA     TOKEN

;
CPI     IDEN
JNZ     SCT2

;
; ACCUMULATING AN IDENTIFIER
CALL    TDOLL ;S?
JZ      SCTOK ;IF SO, SKIP IT
CALL    ALNUM ;ALPHA NUMERIC?
RZ
;RETURN IF END
; NOT END OF THE IDENTIFIER
JMP     SCTOK

;
SCT2: ;NOT SPECIAL OR IDENT, CHECK NUMBER
CPI     NUMB
JNZ     SCT3

;
; ACCUMULATING A NUMBER, CHECK FOR $
CALL    TDOLL
JZ      SCTOK ;SKIP IF FOUND
CALL    HEX ;HEX CHARACTER?
JNZ     SCTOK ;STORE IT IF FOUND
END OF NUMBER, LOOK FOR RADIX INDICATOR

;
;
LDA     NEXTC
CPI     'O' ;OCTAL INDICATOR
JZ      NOCT
CPI     'Q' ;OCTAL INDICATOR
JNZ     NUM2

;
NOCT: ;OCTAL
MVI     A, OCTV
JMP     SSTYP

;
NUM2: CPI     'H'
JNZ     NUM3

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

1294 3E18      MVI    A,HEXV
1296 328B11   SSTYP: STA    STYPE
1299 AP       XRA    A
129A 328A11   STA    NEXTC ;CLEARS THE LOOKAHEAD CHARACTER
129D C38B12   JMP    NCON
;
;          RADIX MUST COME FROM ACCUM
12A8 3A8911   NUM3: LDA    LASTC
12A3 FE42     CPI    "B"
12A5 C2AD12   JNZ    NUM4
12A6 3E82     MVI    A,BINV
12AA C3B412   JMP    SSTY1
;
12AD FE44     NUM4: CPI    "D"
12AF 3E8A     MVI    A,DECV
12B1 C28612   JNZ    SSTY2
12B4 218801   SSTY1: LXI   H,ACCLN
12B7 35       DCR    M ;ACCLN DECREMENTED TO REMOVE RADIX INDICATOR
12B8 328B11   SSTY2: STA    STYPE
;
NCON: ;NUMERIC CONVERSION OCCURS HERE
12B9 218800   LXI    H,0
12BE 228601   SHLD  VALUE ;VALUE ACCUMULATES BINARY EQUIVALENT
12C1 218801   LXI    H,ACCLN
12C4 4E       MOV    C,M ;C=ACCLN
12C5 23       INX    H ;ADDRESSES ACCUM
CLOP: ;NEXT DIGIT IS PROCESSED HERE
12C6 7E       MOV    A,M
12C7 23       INX    H ;READY FOR NEXT LOOP
12C8 FE41     CPI    "A"
12CA D2D212   JNC    CLOP1 ;NOT HEX A-F
12CD D638     SUI    "0" ;NORMALIZE
12CF C3D412   JMP    CLOP2
;
CLOP1: ;HEX A-F
12D2 D637     SUI    "A"-10
CLOP2: ;CHECK SIZE AGAINST RADIX
12D4 E5       PUSH  H ;SAVE ACCUM ADDR
12D5 C5       PUSH  B ;SAVE CURRENT POSITION
12D6 4F       MOV    C,A
12D7 218B11   LXI    H,STY
12DA BE       CMP    M
12DB D41813   CNC    ERRV ;VALUE ERROR IF DIGIT>=RADIX
12DE 8682     MVI    B,8 ;DOUBLE PRECISION DIGIT
12E8 7E       MOV    A,M ;RADIX TO ACCUMULATOR
12E1 2A8601   LHLD  VALUE
12E4 EB       XCHG  ;VALUE TO D,E - ACCUMULATE RESULT IN H,L
12E5 218800   LXI    H,0 ;ZERO ACCUMULATOR
CLOP3: ;LOOP UNTIL RADIX GOES TO ZERO
12E8 B7       ORA    A

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

12E9 CAF712   JZ     CLOP4
12EC 1F       RAR
12ED D2F112   JNC    TTWO ;TEST LSB
12F0 19       DAD    D ;SKIP SUMMING OPERATION IF LSB=0
;
TTWO: ;MULTIPLY VALUE * 2 FOR SHL OPERATION
12F1 EB       XCHG  ;ADD IN VALUE
12F2 29       DAD    D
12F3 EB       XCHG  ;MULTIPLY VALUE * 2 FOR SHL OPERATION
12F4 C3E812   JMP    CLOP3
;
;
CLOP4: ;END OF NUMBER CONVERSION
12F7 09       DAD    B ;DIGIT ADDED IN
12F8 228601   SHLD  VALUE
12FB C1       POP    B
12FC E1       POP    H
12FD 8D       DCR    C ;MORE DIGITS?
12FE C2C612   JNZ    CLOP
1301 C9       RET    ;DONE WITH THE NUMBER
;
SCT3: ;MUST BE A STRING
1302 3A8A11   LDA    NEXTC
1305 FE8D     CPI    CR ;END OF LINE?
1307 CA1E13   JZ     ERRO ;AND RETURN
130A FE27     CPI
130C C23C12   JNZ    SCTOK
130F CDAD11   CALL  GNCN
1312 FE27     CPI
1314 C8       JMP    SCTOK ;RETURN IF SINGLE QUOTE ENCOUNTERED
;
;          OTHERWISE TREAT AS ONE QUOTE
;
;          END OF SCANNER
;
;          ERROR MESSAGE ROUTINES
ERRV: ;"V" VALUE ERROR
1318 F5       PUSH  PSW
1319 3E56     MVI    A,"V"
131B C32413   JMP    ERR
;
ERRO: ;"O" OVERFLOW ERROR
131E F5       PUSH  PSW
131F 3E4F     MVI    A,"O"
1321 C32413   JMP    ERR
;
ERR: ;PRINT ERROR MESSAGE
1324 C5       PUSH  B
1325 E5       PUSH  H
1326 CD1802   CALL  PERR
1329 E1       POP   H
132A C1       POP   B

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

132B F1  
132C C9  
1488 =  
132D

POP  
RET  
ENDMOD EQU  
END

PSW  
(\$ AND 0FF00H) + 100H

```
CP/M VERSION _____  
COPYRIGHT © 1976 1340  
DIGITAL RESEARCH 0200 =  
P. O. BOX 579 0212 =  
PACIFIC GROVE, CA. 93950 021E =  
SER. # _____ 1340 C3A015  
1343 C35C14  
1346 C39E14  
1349 C39814  
134C C3E814  
134F C36815  
1352 C37215  
1355 C38D15  
1358 C39615  
0078 =  
010C =  
0184 =  
0185 =  
0186 =  
0188 =  
0040 =  
0189 =  
01C9 =  
01CB =  
01CD =  
01CF =  
01D0 =  
01D2 =  
01D4 =  
01D6 =  
0001 =  
0002 =  
0003 =  
0004 =  
0001 =
```

```
SYMBOL TABLE MANIPULATION MODULE  
ORG 1340H  
EQU 200H ;IO MODULE ENTRY POINT  
PCON EQU IOMOD+12H  
EOR EQU IOMOD+1EH  
ENTRY POINTS TO SYMBOL TABLE MODULE  
JMP ENDMOD  
JMP INISY  
JMP LOOKUP  
JMP FOUND  
JMP ENTER  
JMP SETTY  
JMP GETTY  
JMP SETVAL  
JMP GETVAL  
COMMON EQUATES  
PBMAX EQU 120 ;MAX PRINT SIZE  
PBUFF EQU 10CH ;PRINT BUFFER  
PBP EQU PBUFF+PBMAX ;PRINT BUFFER POINTER  
TOKEN EQU PBP+1 ;CURRENT TOKEN UNDER SCAN  
VALUE EQU TOKEN+1 ;VALUE OF NUMBER IN BINARY  
ACCLEN EQU VALUE+2 ;ACCUMULATOR LENGTH  
ACMAX EQU 64 ;MAX ACCUMULATOR LENGTH  
ACCUM EQU ACCLEN+1  
EVALUE EQU ACCUM+ACMAX ;VALUE FROM EXPRESSION ANALYSIS  
SYTOP EQU EVALUE+2 ;CURRENT SYMBOL TOP  
SYMAY EQU SYTOP+2 ;MAX ADDRESS+1  
PASS EQU SYMAX+2 ;CURRENT PASS NUMBER  
FPC EQU PASS+1 ;FILL ADDRESS FOR NEXT HEX BYTE  
ASPC EQU FPC+2 ;ASSEMBLER'S PSEUDO PC  
SYBAS EQU ASPC+2 ;BASE OF SYMBOL TABLE  
SYADR EQU SYBAS+2 ;CURRENT SYMBOL BEING ACCESSED  
GLOBAL EQUATES  
IDEN EQU 1 ;IDENTIFIER  
NUMB EQU 2 ;NUMBER  
STRNG EQU 3 ;STRING  
SPECL EQU 4 ;SPECIAL CHARACTER  
PLABT EQU 0001B ;PROGRAM LABEL
```

```
CP/M VERSION _____  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # _____
```

```

0022 = DLABT EQU 0010B ;DATA LABEL
0024 = EOUT EQU 0100B ;EQUATE
0025 = SETT EQU 0101B ;SET
0026 = MACT EQU 0110B ;MACRO
;
0008 = EXTT EQU 1000B ;EXTERNAL
000B = REPT EQU 1011B ;REFER
000C = GLBT EQU 1100B ;GLOBAL
;
;

```

```

000D = CR EQU 0DH
;
;

```

```

; DATA AREAS
; SYMBOL TABLE BEGINS AT THE END OF THIS MODULE
0005 = FIXD EQU 5 ;5 BYTES OVERHEAD WITH EACH SYMBOL ENTRY
; 2BY COLLISION, 1BY TYPE/LEN, 2BY VALUE
0008 = HSIZE EQU 128 ;HASH TABLE SIZE
007F = HMASK EQU HSIZE-1 ;HASH MASK FOR CODING
135B = HASHT: DS HSIZE*2 ;HASH TABLE
145B = HASHC: DS 1 ;HASH CODE AFTER CALL ON LOOKUP
;
;

```

SYMBOL TABLE ENTRY FORMAT IS

```

-----
: HIGH VAL BYTE :
-----
: LOW VAL BYTE :
-----
: CHARACTER N :
-----
: ... :
-----
: CHARACTER 1 :
-----
: TYPE : LENG :
-----
: HIGH COLLISION:
-----
SYADR= : LOW COLLISION :
-----

```

WHERE THE LOW/HIGH COLLISION FIELD ADDRESSES ANOTHER ENTRY WITH THE SAME HASH CODE (OR ZERO IF THE END OF CHAIN), TYPE DESCRIBES THE ENTRY TYPE (GIVEN BELOW), LENG IS THE NUMBER OF CHARACTERS IN THE SYMBOL PRINTNAME -1 (I.E., LENG=0 IS A SINGLE CHARACTER PRINTNAME, WHILE LENG=15 INDICATES A 16 CHARACTER NAME). CHARACTER 1 THROUGH N GIVE THE PRINTNAME CHARACTERS IN ASCII UPPER CASE (ALL LOWER CASE NAMES ARE TRANSLATED ON INPUT), AND THE LOW/HIGH VALUE GIVE THE PARTICULAR ADDRESS OR CONSTANT VALUE ASSOCIATED WITH THE NAME. THE REPRESENTATION OF MACROS DIFFERS IN THE FIELDS WHICH FOLLOW THE VALUE FIELD (MACROS ARE NOT CURRENTLY IMPLEMENTED).

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

0001 = PLABT EQU 0001B ;PROGRAM LABEL
0002 = DLABT EQU 0010B ;DATA LABEL
0004 = EOUT EQU 0100B ;EQUATE
0005 = SETT EQU 0101B ;SET
0026 = MACT EQU 0110B ;MACRO
;
0008 = EXTT EQU 1000B ;EXTERNAL ATTRIBUTE
000B = REPT EQU 1011B ;REFER
000C = GLBT EQU 1100B ;GLOBAL ATTRIBUTE
;
145C 215B13 LXI H,HASHT ;ZERO THE HASH TABLE
145F 0680 MVI B,HSIZE
1461 AF XRA A ;CLEAR ACCUM
;
INIG: MOV M,A
INX H
MOV M,A ;CLEAR DOUBLE WORD
INX H
DCR B
JNZ INIG
;
SET SYMBOL TABLE POINTERS
LXI H,0
SHLD SYADR

```

THE TYPE FIELD CONSISTS OF FOUR BITS WHICH ARE ASSIGNED AS FOLLOWS:

```

0000 UNDEFINED SYMBOL
0001 LOCAL LABELLED PROGRAM
0010 LOCAL LABELLED DATA
0011 (UNUSED)
0100 EQUATE
0101 SET
0110 MACRO
0111 (UNUSED)
1000 (UNUSED)
1001 EXTERN LABELLED PROGRAM
1010 EXTERN LABELLED DATA
1011 REFERENCE TO MODULE
1100 (UNUSED)
1101 GLOBAL UNDEFINED SYMBOL
1110 GLOBAL LABELLED PROGRAM
1111 (UNUSED)

```

TYPE DEFINITIONS

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

1470 C9      RET
;
; CHASH: ;COMPUTE HASH CODE FOR CURRENT ACCUMULATOR
1471 218801 LXI    H,ACCLN
1474 46      MOV    B,M      ;GET ACCUM LENGTH
1475 A7      XRA    A      ;CLEAR ACCUMULATOR
1476 23      CH0: INX    H      ;MOVE TO FIRST/NEXT CHARACTER POSITION
1477 86      ADD    M      ;ADD WITH OVERFLOW
1478 85      DCR    B
1479 C27614 JNZ    CH0
147C E57F    ANI    HMASK ;MASK BITS FOR MODULO HZISE
147E 325B14 STA    HASHC ;FILL HASHC WITH RESULT
1481 C9      RET

;
; SETLN: ;SET THE LENGTH FIELD OF THE CURRENT SYMBOL
1482 47      MOV    B,A      ;SAVE LENGTH IN B
1483 2AD601  LHLD  SYADR
1486 23      INX    H
1487 23      INX    H
1488 7E      MOV    A,M      ;GET TYPE/LENGTH FIELD
1489 E6F0    ANI    0F0H ;MASK OUT TYPE FIELD
148B B0      ORA    B      ;MASK IN LENGTH
148C 77      MOV    M,A
148D C9      RET

;
; GETLN: ;GET THE LENGTH FIELD TO REG-A
148E 2AD601  LHLD  SYADR
1491 23      INX    H
1492 23      INX    H
1493 7E      MOV    A,M
1494 E60F    ANI    0FH
1496 3C      INR    A      ;LENGTH IS STORED AS VALUE - 1
1497 C9      RET

;
; FOUND: ;FOUND RETURNS TRUE IF SYADR IS NOT ZERO (TRUE IS NZ FLAG HERE)
1498 2AD601  LHLD  SYADR
1493 7D      MOV    A,L
149C B4      ORA    H
149D C9      RET

;
; LOOKUP: ;LOOK FOR SYMBOL IN ACCUMULATOR
149E CD7114 CALL  CHASH ;COMPUTE HASH CODE
;
; NORMALIZE IDENTIFIER TO 16 CHARACTERS
14A1 218801 LXI    H,ACCLN
14A4 7E      MOV    A,M
14A5 FE11    CPI    17
14A7 DAAC14 JC     LENOK
14AA 3610    MVI    M,16

LENOK:

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

;
; LOOK FOR SYMBOL THROUGH HASH TABLE
14AC 215B14 LXI    H,HASHC
14AF 5E      MOV    E,M
14B0 1600    MVI    D,0 ;DOUBLE HASH CODE IN D,ESER.#
14B2 215B13 LXI    H,HASHT ;BASE OF HASH TABLE
14B5 19      DAD    D
14B6 19      DAD    D ;HASHT(HASHC)
14B7 5E      MOV    E,M ;LOW ORDER ADDRESS
14B8 23      INX    H
14B9 66      MOV    H,M
14BA 68      MOV    L,E ;HEADER TO LIST OF SYMBOLS IS IN H,L
LOOK0: SHLD  SYADR
CALL    FOUND
RZ      ;RETURN IF SYADR BECOMES ZERO

;
; OTHERWISE EXAMINE CHARACTER STRING FOR MATCH
14C2 CD8E14 CALL  GETLN ;GET LENGTH TO REG-A
14C5 218801 LXI    H,ACCLN
14C8 BE      CMP    M
14C9 C2E114 JNZ    LCOMP

;
; LENGTH MATCH, TRY TO MATCH CHARACTERS
14CC 47      MOV    B,A ;STRING LENGTH IN B
14CD 23      INX    H ;HL ADDRESSES ACCUM
14CE EB      XCHG ;TO D,E
14CF 2AD601  LHLD  SYADR
14D2 23      INX    H
14D3 23      INX    H
14D4 23      INX    H ;ADDRESSES CHARACTERS
LOOK1: LDAX  D ;NEXT CHARACTER FROM ACCUM
CMP    M ;NEXT CHARACTER IN SYMBOL TABLE
JNZ    LCOMP

;
; CHARACTER MATCHED, INCREMENT TO NEXT
14DA 13      INX    D
14DB 23      INX    H
14DC 05      DCR    B
14DD C2D514 JNZ    LOOK1

;
; COMPLETE MATCH AT CURRENT SYMBOL, SYADR IS SET
;
; RET
;
; LCOMP: ;NOT FOUND, MOVE SYADR DOWN ONE COLLISION ADDRESS
14E1 2AD601  LHLD  SYADR
14E4 5E      MOV    E,M
14E5 23      INX    H
14E6 56      MOV    D,M ;COLLISION ADDRESS IN D,E
14E7 EB      XCHG
14E8 C3BB14 JMP    LOOK0
;
;

```



```

ENTER: ;ENTER SYMBOL IN ACCUMULATOR
; ENSURE THERE IS ENOUGH SPACE IN THE TABLE
14E9 218801 LXI H,ACCLN
14EE 5E MOV E,M
14EF 1600 MVI D,0 ;DOUBLE PRECISION ACCLN IN D,E
14F1 2ACB01 LHLD SYTOP
14F4 22D601 SHLD SYADR ;NEXT SYMBOL LOCATION
14F7 19 DAD D ;SYTOP+ACCLN
14F8 110500 LXI D,FXD ;FIXED DATA/SYMBOL
14FB 19 DAD D ;HL HAS NEXT TABLE LOCATION FOR SYMBOL
14FC EB XCHG ;NEW SYTOP IN D,E
14FD 2ACD01 LHLD SYMAX ;MAXIMUM SYTOP VALUE
1500 7B MOV A,E
1501 95 SUB L ;COMPUTE 16-BIT DIFFERENCE
1502 7A MOV A,D
1503 9C SBB H
1504 EB XCHG ;NEW SYTOP IN H,L
1505 D24115 JNC OVERER ;OVERFLOW IN TABLE

; OTHERWISE NO ERROR
1508 22C801 SHLD SYTOP ;SET NEW TABLE TOP
153B 2AD601 LHLD SYADR ;SET COLLISION FIELD
150E EB XCHG ;CURRENT SYMBOL ADDRESS TO D,E
150F 215B14 LXI H,HASHC ;HASH CODE FOR CURRENT SYMBOL TO H,L
1512 4E MOV C,M ;LOW BYTE
1513 0600 MVI B,0 ;DOUBLE PRECISION VALUE IN B,C
1515 215B13 LXI H,HASHT ;BASE OF HASH TABLE
1518 09 DAD B
1519 09 DAD B ;HASHT(HASHC) IN H,L
; D,E ADDRESSES CURRENT SYMBOL - CHANGE LINKS
151A 4E MOV C,M ;LOW ORDER OLD HEADER
151B 23 INX H
151C 46 MOV B,M ;HIGH ORDER OLD HEADER
151D 72 MOV M,D ;HIGH ORDER NEW HEADER TO HASH TABLE
151E 28 DCX H
151F 73 MOV M,E ;LOW ORDER NEW HEADER TO HASH TABLE
1520 EB XCHG ;H,L HOLDS SYMBOL TABLE ADDRESS
1521 71 MOV M,C ;LOW ORDER OLD HEADER TO COLLISION FIELD
1522 23 INX H
1523 70 MOV M,B ;HIGH ORDER OLD HEADER TO COLLISION FIELD

; HASH CHAIN NOW REPAIRED FOR THIS ENTRY, COPY THE PRINTNAME
1524 118801 LXI D,ACCLN
1527 1A LDAX D ;GET SYMBOL LENGTH
1528 FE11 CPI 17 ;LARGER THAN 16 SYMBOLS?
152A CA2F15 JC ENTL
152D 3E10 MVI A,16 ;TRUNCATE TO 16 CHARACTERS
; COPY LENGTH FIELD, FOLLOWED BY PRINTNAME CHARACTERS
152F 47 ENT1: MOV B,A ;COPY LENGTH TO B
1530 3D DCR A ;1-16 CHANGED TO 0-15

```

```

1531 23 INX H ;FOLLOWING COLLISION FIELD
1532 77 MOV M,A ;STORE LENGTH WITH UNDEFINED TYPE (0000)
1533 23 ENT2: INX H
1534 13 INX D
1535 1A LDAX D
1536 77 MOV M,A ;STORE NEXT CHARACTER OF PRINTNAME
1537 05 DCR B ;LENGTH=LENGTH-1
1538 C23315 JNZ ENT2 ;FOR ANOTHER CHARACTER

; PRINTNAME COPIED, ZERO THE VALUE FIELD
153B AF XRA A ;ZERO A
153C 23 INX H ;LOW ORDER VALUE
153D 77 MOV M,A
153E 23 INX H
153F 77 MOV M,A ;HIGH ORDER VALUE
1540 C9 RET

; OVERER: ;OVERFLOW IN SYMBOL TABLE
1541 214A15 LXI H,ERRO
1544 CD1202 CALL PCON
1547 C31E02 JMP EOR ;END OF EXECUTION
154A 53594D424FERRO: DB "SYMBOL TABLE OVERFLOW",CR

; SETTY: ;SET CURRENT SYMBOL TYPE TO VALUE IN REG-A
1560 17 RAL
1561 17 RAL
1562 17 RAL
1563 17 RAL
1564 E6F0 ANI 0F0H ;TYPE MOVED TO HIGH ORDER 4-BITS
1566 47 MOV B,A ;SAVE IT IN B
1567 2AD601 LHLD SYADR ;BASE OF SYMBOL TO ACCESS
156A 23 INX H
156B 23 INX H ;ADDRESS OF TYPE/LENGTH FIELD
156C 7E MOV A,M ;GET IT AND MASK
156D E60F ANI 0FH ;LEAVE LENGTH
156F B0 ORA B ;MASK IN TYPE
1570 77 MOV M,A ;STORE IT
1571 C9 RET

; GETTY: ;RETURN THE TYPE OF THE VALUE IN CURRENT SYMBOL
1572 2AD601 LHLD SYADR
1575 23 INX H
1576 23 INX H
1577 7E MOV A,M
1578 1F RAR
1579 1F RAR
157A 1F RAR
157B 1F RAR
157C E60F ANI 0FH ;TYPE MOVED TO LOW 4-BITS OF REG-A
157E C9 RET

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

QP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1975  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

; VALADR:
157F CD8E14 CALL GETLN ;GET VALUE FIELD ADDRESS FOR CURRENT SYMBOL
1582 2AD601 LHLD SYADR ;PRINTNAME LENGTH TO ACCUM
1585 5P MOV E,A ;BASE ADDRESS
1586 1600 MVI D,0
1588 19 DAD D ;BASE(LEN)
1589 23 INX H
158A 23 INX H ;FOR COLLISION FIELD
158B 23 INX H ;FOR TYPE/LEN FIELD
158C C9 RET ;WITH H,L ADDRESSING VALUE FIELD

; SETVAL:
158D E5 VALUE IS SENT IN H,L
158E CD7F15 PUSH H ;SAVE VALUE TO SET
1591 D1 CALL VALADR
1592 73 POP D ;POP VALUE TO SET, HL HAS ADDRESS TO FILL
1593 23 MOV M,E
1594 72 INX H
1595 C9 MOV M,D ;FIELD SET
RET

; GETVAL:
1596 CD7F15 CALL VALADR ;GET THE VALUE FIELD OF THE CURRENT SYMBOL TO H,L
1599 5E MOV E,M ;ADDRESS OF VALUE FIELD TO H,L
159A 23 INX H
159B 56 MOV D,M
159C EB XCHG
159D C9 RET

15A0 = ENDMOD EQU ($ AND 0FF00H) + 20H
159E END

```

33

```

15A0 ORG 15A0H
15A0 C36018 JMP ENDMOD ;TO NEXT MODULE
15A3 C38317 JMP BSEAR
15A6 C31018 JMP BGET

; COMMON EQUATES
PBMAX EQU 120 ;MAX PRINT SIZE
PBUFF EQU 10CH ;PRINT BUFFER
PBP EQU PBUFF+PBMAX ;PRINT BUFFER POINTER

; TOKEN EQU PBP+1 ;CURRENT TOKEN UDER SCAN
VALUE EQU TOKEN+1 ;VALUE OF NUMBER IN BINARY
ACCLEN EQU VALUE+2 ;ACCUMULATOR LENGTH
ACCMAX EQU 64 ;MAX ACCUMULATOR LENGTH
ACCM EQU ACCLEN+1

; EVALUE EQU ACCUM+ACMAX ;VALUE FROM EXPRESSION ANALYSIS

; SYTOP EQU EVALUE+2 ;CURRENT SYMBOL TOP
SYMAY EQU SYTOP+2 ;MAX ADDRESS+1

; PASS EQU SYMAX+2 ;CURRENT PASS NUMBER
FPC EQU PASS+1 ;FILL ADDRESS FOR NEXT HEX BYTE
ASPC EQU FPC+2 ;ASSEMBLER'S PSEUDO PC

; GLOBAL EQUATES
IDEN EQU 1 ;IDENTIFIER
NUMB EQU 2 ;NUMBER
STRNG EQU 3 ;STRING
SPFCL EQU 4 ;SPECIAL CHARACTER

; PLABT EQU 0001B ;PROGRAM LABEL
DLABT EQU 0010B ;DATA LABEL
EQUATE EQU 0100B ;EQUATE
SETT EQU 0101B ;SET
MACT EQU 0110B ;MACRO

; EXTT EQU 1000B ;EXTERNAL
REFT EQU 1011B ;REFER
GLBT EQU 1100B ;GLOBAL

; CR EQU 0DH ;CARRIAGE RETURN

; TABLE DEFINITIONS
; TYPES

```

```

0000 = XBASE EQU 0 ;START OF OPERATORS
;
; O1 THROUGH O15 DENOTE OPERATIONS
0010 = RT EQU 16
0011 = PT EQU RT+1 ;RT IS REGISTER TYPE, PT IS PSEUDO OPERATION
0012 = OBASE EQU PT+1
0013 = O1 EQU OBASE+1 ;SIMPLE
0014 = O2 EQU OBASE+2 ;LXI
0015 = O3 EQU OBASE+3 ;DAD
0016 = O4 EQU OBASE+4 ;PUSH/POP
0017 = O5 EQU OBASE+5 ;JMP/CALL
0018 = O6 EQU OBASE+6 ;MOV
0019 = O7 EQU OBASE+7 ;MVI
001A = O8 EQU OBASE+8 ;ACC IMMEDIATE
001B = O9 EQU OBASE+9 ;LDAX/STAX
001C = O10 EQU OBASE+10 ;LHLD/SHLD/LDA/STA
001D = O11 EQU OBASE+11 ;ACCUM REGISTER
001E = O12 EQU OBASE+12 ;INC/DEC
001F = O13 EQU OBASE+13 ;INX/DCX
0020 = O14 EQU OBASE+14 ;RST
0021 = O15 EQU OBASE+15 ;IN/OUT
;
;

```

```

; X1 THROUGH X15 DENOTE OPERATORS
0000 = X1 EQU XBASE ;*
0001 = X2 EQU XBASE+1 ;/
0002 = X3 EQU XBASE+2 ;MOD
0003 = X4 EQU XBASE+3 ;SHL
0004 = X5 EQU XBASE+4 ;SHR
0005 = X6 EQU XBASE+5 ;+
0006 = X7 EQU XBASE+6 ;-
0007 = X8 EQU XBASE+7 ;UNARY -
0008 = X9 EQU XBASE+8 ;NOT
0009 = X10 EQU XBASE+9 ;AND
000A = X11 EQU XBASE+10 ;OR
000B = X12 EQU XBASE+11 ;XOR
000C = X13 EQU XBASE+12 ;(
000D = X14 EQU XBASE+13 ;)
000E = X15 EQU XBASE+14 ;,
000F = X16 EQU XBASE+15 ;CR
;
;

```

RESERVED WORD TABLES

BASE ADDRESS VECTOR FOR CHARACTERS

```

15A9 C415 CINX: DW CHAR1 ;LENGTH 1 BASE
15AB D415 DW CHAR2 ;LENGTH 2 BASE
15AD E615 DW CHAR3 ;LENGTH 3 BASE
15AF 8216 DW CHAR4 ;LENGTH 4 BASE
15B1 AE16 DW CHAR5 ;LENGTH 5 BASE

```

```

15B3 BD16 DW CHAR6 ;LENGTH 6 BASE
;
0005 = CMAX EQU ($-CINX)/2-1 ;LARGEST STRING TO MATCH
;
; CLEN: ;LENGTH VECTOR GIVES THE NUMBER OF ITEMS IN EACH TABLE
15B5 10 DB CHAR2-CHAR1
15B6 09 DB (CHAR3-CHAR2)/2
15B7 34 DB (CHAR4-CHAR3)/3
15B8 0B DB (CHAR5-CHAR4)/4
15B9 03 DB (CHAR6-CHAR5)/5
;
; TVINX: ;TABLE OF TYPE,VALUE PAIRS FOR EACH RESERVED SYMBOL
15BA BD16 DW TV1
15BC DD16 DW TV2
15BE EF16 DW TV3
15C0 5717 DW TV4
15C2 6D17 DW TV5
;
; CHARACTER VECTORS FOR 1,2,3,4, AND 5 CHARACTER NAMES
15C4 0D28292A CHAR1: DB CR, '()*'
15C8 2B DB '+', '.,-/A'
15C9 2C2D2F41 DB 'BCDE'
15CD 42434445 DB 'HLM'
15D1 484C4D DB
;
15D4 4442444944 CHAR2: DB 'DBDIDSDW'
15DC 4549494649 DB 'EIIFINOR'
15E4 5350 DB 'SP'
;
15E6 4143494144 CHAR3: DB 'ACIADCADDADI'
15F2 414E41414E DB 'ANAANDANICMA'
15FE 434D43434D DB 'CMCCMPCPIDAA'
160A 4441444443 DB 'DADDCRDCRND'
1616 455155484C DB 'EQUHLTINRINX'
1622 4A4D504C44 DB 'JMFLDALXIMOD'
162E 4D4F564D56 DB 'MOVXVINOPNOT'
163A 4F52414F52 DB 'ORACRGORIOUT'
1646 504F505053 DB 'POPPSWALNAR'
1652 524554524C DB 'RETRLCRCRST'
165E 5342425342 DB 'SBBSEISETSHL'
166A 5348525354 DB 'SHRSTASTCSUB'
1676 535549584F DB 'SUIXORXRAXRI'
;
1682 43414C4C45 CHAR4: DB 'CALLENMLDAXLHLDPCHL'
1696 5055534853 DB 'PUSHSHLDSPHLSTAX'
16A6 5843484758 DB 'XCHGXTHL'
;
16AE 454E444946 CHAR5: DB 'ENDIFMACROTITLE'
;
; CHAR6: ;END OF CHARACTER VECTOR

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950

```

;
;TV1: ;TYPE,VALUE PAIRS FOR CHAR1 VECTOR
16BD 0F0A0C14 DB X16,10, X13,20
16C1 0D1E0050 DB X14,30, X1,80
16C5 0546 DB X6,70
16C7 0E0A0646 DB X15,10, X7,70
16CB 01521007 DB X2,80, RT,7
16CF 10001001 DB RT,0, RT,1
16D3 10021003 DB RT,2, RT,3
16D7 10041005 DB RT,4, RT,5
16DB 1006 DB RT,6

```

```

;
;TV2: ;TYPE,VALUE PAIRS FOR CHAR2 VECTOR
16DD 110113F3 DB PT,1, O1,0F3H ;DB DI
16E1 11021103 DB PT,2, PT,3 ;DS DW
16E5 13FB1108 DB O1,0FBH, PT,8 ;EI IF
16E9 21CB0A28 DB O15,0DBH, X11,40 ;IN OR
16ED 1006 DB RT,6 ;SP

```

```

;
;TV3: ;TYPE,VALUE PAIRS FOR CHAR3 VECTOR
16EF 1ACE1D88 DB O8,0CEH, O11,88H ;ACI ADC
16F3 1D901AC6 DB O11,80H, O8,0C6H ;ADD ADI
16F7 1DA00932 DB O11,0A0H, X10,50 ;ANA AND
16FB 1AE6132F DB O8,0E6H, O1,2FH ;ANI CMA
16FF 133F1D88 DB O1,3FH, O11,0B8H ;CMC CMP
1703 1AFE1327 DB O8,0FEH, O1,27H ;CPI DAA
1707 15991E05 DB O3,09H, O12,05H ;DAD DCR
170B 1F0B1104 DB O13,0BH, PT,4 ;DCX END
170F 11071376 DB PT,7, O1,76H ;EQU HLT
1713 1E041E03 DB O12,04H, O13,03H ;INR INX
1717 17C31C3A DB O5,0C3H, O10,3AH ;JMP LDA
171B 14010250 DB O2,01H, X3,80 ;LXI MOD
171F 18401906 DB O6,40H, O7,06H ;MOV MVI
1723 1320003C DB O1,00H, X9,60 ;NOP NOT
1727 1DB0110A DB O11,0B0H, PT,10 ;ORA ORG
172B 1AF621D3 DB O8,0F6H, O15,0D3H ;ORI OUT
172F 16C11006 DB O4,0C1H, RT,6 ;POP PSW
1733 1317131F DB O1,17H, O1,1FH ;RAL RAR
1737 13C91307 DB O1,0C9H, O1,07H ;RET RLC
173B 130F20C7 DB O1,0FH, O14,0C7H ;RRC RST
173F 1D981ADE DB O11,098H, O3,0DEH ;SBB SBI
1743 110B0350 DB PT,11, X4,80 ;SET SHL
1747 04501C32 DB X5,80, O10,32H ;STA STC
174B 13371D90 DB O1,37H, O11,90H ;STC SUB
174F 1AD60B28 DB O8,0D6H, X12,40 ;SUI XOR
1753 1DA81AE8 DB O11,0A8H, O8,0EEH ;XRA XRI

```

```

;
;TV4: ;TYPE,VALUE PAIRS FOR CHAR4 VECTOR

```

SER, CR (#)

```

1757 17CD DB O5,0CDH ;CALL
1759 11061B0A DB PT,6, O9,0AH ;ENEM LDAX
175D 1C2A13E9 DB O10,02AH, O1,0E9H ;LHLD PCHL
1761 16C51C22 DB O4,0C5H, O10,22H ;PUSH SHLD
1765 13F91B02 DB O1,0F9H, O9,02H ;SPLH STAX
1769 13EB13E3 DB O1,0EBH, O1,0E3H ;XCHG XTAL

```

```

;
;TV5: ;TYPE,VALUE PAIRS FOR CHAR5 VECTOR
176D 11051109 DB PT,5, PT,9 ;ENDIF MACRO
1771 110C DB PT,12 ;TITLE

```

```

;
;SUPTAB: ;TABLE OF SUFFIXES FOR J C AND R OPERATIONS
1773 4E5A5A204E DB 'NZZ NCC POPEP M'

```

```

;
;BSEAR: ;BINARY SEARCH MNEMONIC TABLE
; INPUT: UR = UPPER BOUND OF TABLE (I.E., TABLE LENGTH-1)
; SR = SIZE OF EACH TABLE ELEMENT
; H,L ADDRESS BASE OF TABLE TO SEARCH
; OUTPUT: ZERO FLAG INDICATES MATCH WAS FOUND, IN WHICH CASE
; THE ACCUMULATOR CONTAINS AN INDEX TO THE ELEMENT
; NOT ZERO FLAG INDICATES NO MATCH FOUND IN TABLE

```

```

;
; UR EQU B ;UPPER BOUND REGISTER
; LR EQU C ;LOWER BOUND REGISTER
; SR EQU D ;SIZE REGISTER
; MR EQU E ;MIDDLE POINTER REGISTER
; SPL EQU B ;SIZE PRIME, USED IN COMPUTING MIDDLE POSITON
; SPLP EQU C ;ANOTHER COPY OF SIZE PRIME
; KR EQU H ;K

```

```

1783 1EFF MVI MR,255 ;MARK M <> OLD M
1785 04 INR UR ;U=U+1
1786 0E00 MVI LR,0 ;L = 0

```

```

;
; COMPUTE M' = (U+L)/2
; NEXT: XRA A
; MOV A,UR ;CY=0, A=U
; ADD LR ;(U+L)
; RAR ;(U+L)/2
; CMP MR ;SAME AS LAST TIME THROUGH?
; JZ NMATCH ;JUMP IF = TO NO MATCH

```

```

;
; MORE ELEMENTS TO SCAN
; MOV MR,A ;NEW MIDDLE VALUE
; PUSH H ;SAVE A COPY OF THE BASE ADDRESS
; PUSH D ;SAVE S,M
; PUSH B ;SAVE U,L
; PUSH H ;SAVE ANOTHER COPY OF THE BASE ADDRESS
; MOV SPL,SR ;S' = S
; MOV SPL,SPL ;S'' = S'

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950

```

1797 1600      MVI    SR,0      ;FOR DOUBLE ADD OPERATION BELOW (DOUBLE M)
;
1799 210000    LXI    KR,0      ;K=0
SUMK: DAD     D        ;K = K + M
179C 19        DCR     SP1      ;S' = S' - 1
179D 05        JNZ     SUMK      ;DECREMENT IF SP1 <> 0
179E C29C17

```

```

;
; K IS NOW RELATIVE BYTE POSITION
17A1 D1        POP     D        ;TABLE BASE ADDRESS
17A2 19        DAD     D        ;H,L CONTAINS ABSOLUTE ADDRESS OF BYTE TO COMPARE
17A3 118901    LXI    D,ACCUM ;D,E ADDRESS CHARACTERS TO COMPARE
;

```

```

; COMK: ;COMPARE NEXT CHARACTER
17A6 1A        LDAX   D        ;ACCUM CHARACTER TO REG A
17A7 BE        CMP     M        ;SAME AS TABLE ENTRY?
17A8 13        INX    D
17A9 23        INX    H        ;TO NEXT POSITIONS
17AA C2B617    JNZ     NCOM     ;JUMP IF NOT THE SAME
17AD 0D        DCR     SP1P     ;MORE CHARACTERS?
17AE C2A617    JNZ     COMK

```

```

;
; COMPLETE MATCH AT M
17B1 C1        POP     B
17B2 D1        POP     D        ;M RESTORED
17B3 E1        POP     H
17B4 7B        MOV     A,MR     ;VALUE OF M COPIED IN A
17B5 C9        RET

```

```

; NCOM: ;NO MATCH, DETERMINE IF LESS OR GREATER
17B6 C1        POP     B        ;U,L
17B7 D1        POP     D        ;S,M
17B8 E1        POP     H        ;TABLE ADDRESS
17B9 DAC017    JC      NCOML
; ACCUM IS HIGHER
17BC 4B        MOV     LR,MR     ;L = M
17BD C38817    JMP     NEXT

```

```

; NCOML: ;ACCUMULATOR IS LOW
17C0 43        MOV     UR,MR     ;U = M
17C1 C38817    JMP     NEXT

```

```

; NMATCH: ;NO MATCH
17C4 AF        XRA     A
17C5 3C        INR     A        ;SETS NOT ZERO FLAG
17C6 C9        RET

```

```

; PREFIX: ;J C OR R PREFIX?
17C7 3A8931    LDA     ACCUM
17CA 0117C2    LXI    B,(0C2H SHL 8) OR 05 ;JNZ OPCODE TO B, TYPE TO C
17CD FE4A      CPI     'J'

```

```

17CF C8
17D0 06C4
17D2 FE43
17D4 C8
17D5 0113C0
17D8 FE52
17DA C9

```

```

17DB 3A8801
17DE FE04
17E0 D20D18
17E3 FE03
17E5 CAF217
17E8 FE02
17EA C20D18
17ED 218B01
17F0 3620

```

```

17F2 010800
17F5 117317

```

```

17F8 218A01
17FB 1A
17FC BE
17FD 13
17FE C20518
1801 1A
1802 23
1803 BE
1804 C8
1805 13
1806 04
1807 0D
1808 C2F817

```

```

180B 0C
180C C9

```

```

180D AF
180E 3C
180F C9

```

```

1810 3A8801
1813 4F

```

```

RZ ;RETURN WITH ZERO FLAG SET IF J
MVI B,0C4H ;CNZ OPCODE TO B, TYPE IS IN C
CPI 'C'
RZ
LXI B,(0C0H SHL 8) OR 01 ;RNZ OPCODE
CPI 'R'
RET

```

```

; SUFFIX: ;J R OR C RECOGNIZED, LOOK FOR SUFFIX
ACLEN
CPI 4 ;CHECK LENGTH
JNC NSUFF ;CARRY IF 0,1,2,3 IN LENGTH
CPI 3
JZ SUF0 ;ASSUME 1 OR 2 IF NO BRANCH
CPI 2
JNZ NSUFF ;RETURNS IF 0 OR 1
LXI H,ACCUM+2
MVI M, ;BLANK-OUT FOR MATCH ATTEMPT
SUF0: ;SEARCH 'TIL END OF TABLE
LXI B,8 ;B=0, C=8 COUNTS TABLE DOWN TO ZERO OR MATCH
LXI D,SUFTAB

```

```

NEXTS: ;LOOK AT NEXT SUFFIX
LXI H,ACCUM+1 ;SUFFIX POSITION
LDAX D ;CHARACTER TO ACCUM
CMP M
INX D ;READY FOR NEXT CHARACTER
JNZ NEXT0 ;JMP IF NO MATCH
LDAX D ;GET NEXT CHARACTER
INX H ;READY FOR COMPARE WITH ACCUM
CMP M ;SAME?
RZ ;RETURN WITH ZERO FLAG SET, B IS SUFFI
NEXT0: INX D ;MOVE TO NEXT CHARACTER
INR B ;COUNT SUFFIX UP
DCR C ;COUNT TABLE LENGTH DOWN
JNZ NEXTS
; END OF TABLE, MARK WITH NON ZERO FLAG
INR C
RET

```

```

; NSUFF: ;NOT PROPER SUFFIX - SET NON ZERO FLAG
XRA A
INR A
RET

```

```

; BGET: ;PERFORM BINARY SEARCH, AND EXTRACT TYPE AND VAL FIELDS FOR
; THE ITEM. ZERO FLAG INDICATES MATCH WAS FOUND, WITH TYPE
; IN THE ACCUMULATOR, AND VAL IN REGISTER B. THE SEARCH IS BAS
; UPON THE LENGTH OF THE ACCUMULATOR
LDA ACCLEN ;ITEM LENGTH
MOV C,A ;SAVE A COPY

```

CP/M VERSION \_\_\_\_\_  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # \_\_\_\_\_

```

1814 3D      DCR      A      ;ACCLEN-1
1815 5F      MOV      E,A
1816 1600    MVI      D,0      ;DOUBLE ACCLEN-1 TO D,E SER. # _____
1818 D5      PUSH     D      ;SAVE A COPY FOR LATER
1819 FE05    CPI      CMAX   ;TOO LONG?
181B D25A18  JNC      NGET     ;NOT IN RANGE IF CARRY
181E 21B515  LXI      H,CLEN   ;LENGTH VECTOR
1821 19      DAD      D
1822 46      MOV      UR,M     ;FILL UPPER BOUND FROM MEMORY
1823 21A915  LXI      H,CINX
1826 19      DAD      D
1827 19      DAD      D      ;BASE ADDRESS TO H,L
1828 56      MOV      D,M
1829 23      INX      H
182A 66      MOV      H,M
182B 6A      MOV      L,D      ;NOW IN H,L
182C 51      MOV      SR,C     ;FILL THE SIZE REGISTER
182D CD9317  CALL    BSEAR    ;PERFORM THE BINARY SEARCH
1830 C24518  JNZ     SCASE    ;ZERO IF FOUND
1833 D1      POP      D      ;RESTORE INDEX
1834 21BA15  LXI      H,TVINX
1837 19      DAD      D
1838 19      DAD      D      ;ADDRESSING PROPER TV ELEMENT
1839 5E      MOV      E,M
183A 23      INX      H
183B 56      MOV      D,M
; D,E IS BASE ADDRESS OF TYPE/VALUE VECTOR, ADD DISPLACEMENT
183C 6F      MOV      L,A
183D 2600    MVI      H,0
183F 29      DAD      H      ;DOUBLED
1840 19      DAD      D      ;INDEXED
1841 7E      MOV      A,M     ;TYPE TO ACC
1842 23      INX      H
1843 46      MOV      B,M     ;VALUE TO B
1844 C9      RET
;
1845 D1      POP      D      ;RESTORE INDEX
1846 CDC717  CALL    PREFIX
1849 C0      RNZ
184A C5      PUSH     B      ;NOT FOUND AS PREFIX J C OR R IF NOT ZERO FLAG
1843 CDB17  CALL    SUFFIX   ;SAVE VALUE AND TYPE
184E 78      MOV      A,B     ;ZERO IF SUFFIX MATCHED
184F C1      POP      B      ;READY FOR MASK IF ZERO FLAG
1850 C0      RNZ
; MASK IN THE PROPER BITS AND RETURN
1851 B7      ORA      A      ;RECALL VALUE AND TYPE
1852 17      RAL
1853 17      RAL
1854 17      RAL
;RETURN IF NOT ZERO FLAG SET

```

CP/M VERSION \_\_\_\_\_

COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950

```

1855 B0
1856 47
1857 79
1858 BF
1859 C9

```

```

ORA      B      ;VALUE SET TO JNZ ...
MOV      B,A    ;REPLACE
MOV      A,C    ;RETURN WITH TYPE IN REGISTER A
CMP      A      ;CLEAR THE ZERO FLAG
RET

```

```

;
; NGET: ;CAN'T FIND THE ENTRY, RETURN WITH ZERO FLAG RESET
;       POP      D      ;GET THE ELEMENT BACK
;       XRA      A      ;CLEAR
;       INR      A      ;ZERO FLAG RESET
;       RET

```

```

185A D1
185B AF
185C 3C
185D C9

```

```

;
; ENDMOD EQU ($ AND 0FFEH) + 20H ;NEXT MODULE ADDRESS
END

```

CP/M VERSION \_\_\_\_\_

COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

```

1860      ; OPERAND SCAN MODULE
          ORG      1860H

          ;
          ; EXTERNALS
0200 =   IOMOD EQU    200H    ;I/O MODULE
1100 =   SCMOD EQU    1100H   ;SCANNER MODULE
1340 =   SYMOD EQU    1340H   ;SYMBOL TABLE MODULE
15A0 =   BMOD EQU     15A0H   ;BINARY SEARCH MODULE
          ;
0218 =   PERR EQU     IOMOD+18H
1106 =   SCAN EQU     SCMOD+6H    ;SCANNER ENTRY POINT
022D =   CR EQU       0DH      ;CARRIAGE RETURN
          ;
1346 =   LOOKUP EQU   SYMOD+6H   ;LOOKUP
1349 =   FOUND EQU   LOOKUP+3    ;FOUND SYMBOL IF ZERO FLAG NOT SET
134C =   ENTER EQU   FOUND+3     ;ENTER SYMBOL
134F =   SETY EQU    ENTER+3     ;SET TYPE FIELD
1352 =   GETY EQU    SETY+3     ;SET TYPE FIELD
1355 =   SETVAL EQU  GETY+3     ;SET VALUE FIELD
1358 =   GETVAL EQU  SETVAL+3    ;GET VALUE FIELD
          ;
15A3 =   BSEAR EQU   BMOD+3     ;BINARY SEARCH ROUTINE
15A6 =   BGET EQU    BSEAR+3    ;GET VALUES WITH SEARCH
          ;
          ; COMMON EQUATES
0078 =   PBMAX EQU   128      ;MAX PRINT SIZE
010C =   PBUFF EQU   10CH     ;PRINT BUFFER
0184 =   PBP EQU     PBUFF+PBMAX ;PRINT BUFFER POINTER
          ;
0185 =   TOKEN EQU   PBP+1     ;CURRENT TOKEN UDER SCAN
0186 =   VALUE EQU   TOKEN+1   ;VALUE OF NUMBER IN BINARY
0188 =   ACCLEN EQU  VALUE+2   ;ACCUMULATOR LENGTH
0240 =   ACMAX EQU   64       ;MAX ACCUMULATOR LENGTH
0189 =   ACCUM EQU   ACCLEN+1
          ;
01C9 =   EVALUE EQU  ACCUM+ACMAX ;VALUE FROM EXPRESSION ANALYSIS
          ;
01C8 =   SYTOP EQU   EVALUE+2   ;CURRENT SYMBOL TOP
01C0 =   SYMAX EQU   SYTOP+2    ;MAX ADDRESS+1
          ;
01CF =   PASS EQU    SYMAX+2    ;CURRENT PASS NUMBER
01D0 =   PPC EQU     PASS+1     ;FILL ADDRESS FOR NEXT HEX BYTE
01D2 =   ASPC EQU    PPC+2     ;ASSEMBLER'S PSEUDO PC
          ;
          ; GLOBAL EQUATES
0201 =   IDEN EQU    1         ;IDENTIFIER
0202 =   NUMB EQU    2         ;NUMBER
  
```

```

0003 =   STRNG EQU    3         ;STRING
0004 =   SPFCCL EQU   4         ;SPECIAL CHARACTER
          ;
0001 =   PLABT EQU    0001B    ;PROGRAM LABEL
0002 =   DLABT EQU    0010B    ;DATA LABEL
0004 =   EQUT EQU     0100B    ;EQUATE
0005 =   SETT EQU     0101B    ;SET
0006 =   MACT EQU     0110B    ;MACRO
          ;
0008 =   EXTT EQU     1000B    ;EXTERNAL
000B =   REFT EQU     1011B    ;REFER
000C =   GLBT EQU     1100B    ;GLOBAL
          ;
          ;
          ; TABLE DEFINITIONS
0000 =   XBASE EQU    0         ;START OF OPERATORS
000F =   OPER EQU     15        ;LAST OPERATOR
0010 =   RT EQU       16
0011 =   PT EQU       RT+1     ;RT IS REGISTER TYPE, PT IS PSEUDO OPERATION
0012 =   OBASE EQU    PT+1
          ;
0005 =   PLUS EQU     5
0006 =   MINUS EQU    6
0008 =   NOTF EQU     8        ;NOT
000C =   LPAR EQU     12
000D =   RPAR EQU     13
000A =   OSMAX EQU    10
0010 =   VSMAX EQU    8*2
          ;
          ;
          ; BEGINNING OF MODULE
1860 C3A01B JMP     ENDMOD ;PAST THIS MODULE
1863 C3191A JMP     OPAND  ;SCAN OPERAND FIELD
1866 C36E19 JMP     MULF   ;MULTIPLY FUNCTION
1869 C33819 JMP     DIVE   ;DIVIDE FUNCTION
186C          UNARY: DS      1 ;TRUE IF NEXT OPERATOR IS UNARY
186D          OPERV: DS     OSMAX ;OPERATOR STACK
1877          HIERV: DS     OSMAX ;OPERATOR PRIORITY
1881          VSTACK: DS     VSMAX ;VALUE STACK
1891          OSP: DS      1 ;OPERATOR STACK POINTER
1892          VSP: DS      1 ;VALUE STACK POINTER
          ;
          ;
          ;
1893 EB          STKV: ;PLACE CURRENT H,L VALUE AT TOP OF VSTACK
1894 219218 XCHG    ;HOLD VALUE IN D,E
1897 7E          LXI    H,VSP
1898 FE10          MOV    A,M
189A DAA218          CPI    VSMAX
          JC     STKV0
  
```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

```

189D CD851B CALL ERREX ;OVERFLOW IN EXPRESSION
18A0 3600 MVI M,0 ;VSP=0
STKV8: MOV A,M ;GET VSP
18A2 72 INR M ;VSP=VSP+1
18A3 34 INR M ;VSP=VSP+2
18A4 34 MOV C,A ;SAVE VSP
18A5 4F MVI B,0 ;DOUBLE VSP
18A6 0600 LXI H,VSTACK
18AB 09 DAD B
18AC 73 MOV M,E ;LOW BYTE
18AD 23 INX H
18AE 72 MOV M,D ;HIGH BYTE
18AF C9 RET

```

```

;
; STKO: ;STACK OPERATOR (REG-A) AND PRIORITY (REG-B)
18B8 F5 PUSH PSW ;SAVE IT
18B1 219118 LXI H,OSP
18B4 7E MOV A,M
18B5 FEA8 CPI OSMAX
18B7 DABF18 JC STK01
18BA 3600 MVI M,0
18BC CD851B CALL ERREX ;OPERATOR STACK OVERFLOW
18BF 5E STK01: MOV E,M ;GET OSP
18C0 1600 MVI D,0
18C2 34 INR M ;OSP=OSP+1
18C3 F1 POP PSW ;RECALL OPERATOR
18C4 216D18 LXI H,OPERV
18C7 19 DAD D ;OPERV(OSP)
18C8 77 MOV M,A ;OPERV(OSP)=OPERATOR
18C9 217718 LXI H,HIERV
18CC 19 DAD D
18CD 70 MOV M,B ;HIERV(OSP)=PRIORITY
18CE C9 RET

```

```

;
; LODV1: ;LOAD TOP ELEMENT FROM VSTACK TO H,L
18CF 219218 LXI H,VSP
18D2 7E MOV A,M
18D3 B7 ORA A
18D4 C2DE18 JNZ LODOK
18D7 CD851B CALL ERREX ;UNDERFLOW
18DA 210000 LXI H,0
18DD C9 RET

```

```

;
; LODOK: DCR M
18DE 35 DCR M ;VSP=VSP-2
18DF 35 DCR M ;LOW BYTE
18E0 4E MOV C,M
18E1 0600 MVI B,0
18E3 218118 LXI H,VSTACK
18E6 09 DAD B ;VSTACK(VSP)
18E7 4E MOV C,M ;GET LOW BYTE

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

18E8 23 INX H
18E9 66 MOV H,M
18EA 69 MOV L,C
18EB C9 RET
;
; LODV2: ;LOAD TOP TWO ELEMENTS DE HOLDS TOP, HL HOLDS TOP-1
18EC CDCF18 CALL LODV1
18EF EB XCHG
18F0 CDCF18 CALL LODV1
18F3 C9 RET

```

```

;
; APPLY: ;APPLY OPERATOR IN REG-A TO TOP OF STACK
18F4 6F MOV L,A
18F5 2600 MVI H,B
18F7 29 DAD H ;OPERATOR NUMBER*2
18F8 110119 LXI D,OPTAB
18FB 19 DAD D ;INDEXED OPTAB
18FC 5E MOV E,M ;LOW ADDRESS
18FD 23 INX H
18FE 66 MOV H,M ;HIGH ADDRESS
18FF 6B MOV L,E
1900 E9 PCHL ;SET PC AND GO TO SUBROUTINE

```

```

;
; OPTAB: DW MULOP
; DIVOP
; MODOP
; SHLOP
; SHROP
; ADDOP
; SUBOP
; NEGOP
; NOTOP
; ANDOP
; OROP
; XOROP
; ERREX ; (

```

```

;
; SHFT: ;SET UP OPERANDS FOR SHIFT L AND R
191B CDEC18 CALL LODV2
191E 7A MOV A,D ;ENSURE 0-15
191F B7 ORA A
1920 C22719 JNZ SHERR
1923 78 MOV A,E
1924 FE11 CPI 17
1926 D8 RC ;RETURN IF 0-16 SHIFT
SHERR: CALL ERREX
192A 3E10 MVI A,16
192C C9 RET
;

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```



CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

192D AF      XRA      A
192E 95      SUB      L
192F 6F      MOV      L,A
1930 3E00    MVI      A,0
1932 9C      SBB      H
1933 67      MOV      H,A
1934 C9      RET

;
1935 CDEC18  DIVF:   CALL    LODV2
;           ;(EXTERNAL ENTRY FROM MAIN PROGRAM)
1938 EB      DIVE:   XCHG    ;SWAP D,E WITH H,L FOR DIVIDE FUNCTION
;           COMPUTE X/Y WHERE X IS IN D,E AND Y IS IN H,L
;           ; THE VALUE OF X/Y APPEARS IN D,E AND X MOD Y IS IN H,L
;
1939 226B19  SHLD    DTEMP ;SAVE X IN TEMPORARY
193C 216D19  LXI     H,BNUM ;STORE BIT COUNT
193F 3611    MVI     M,11H
1941 010300  LXI     B,0 ;INTIALIZE RESULT
1944 C5      PUSH    B
1945 AF      XRA     A ;CLEAR FLAGS

DLOOP:
1946 7B      MOV     A,E ;GET LOW Y BYTE
1947 17      RAL
1948 5F      MOV     E,A
1949 7A      MOV     A,D
194A 17      RAL
194B 57      MOV     D,A
194C 35      DCR     M ;DECREMENT BIT COUNT
194D E1      POP     H ;RESTORE TEMP RESULT
194E C8      RZ     ;ZERO BIT COUNT MEANS ALL DONE
194F 3E00    MVI     A,0 ;ADD IN CARRY
1951 CE00    ACI     0 ;CARRY
1953 29      DAD     H ;SHIFT TEMP RESULT LEFT ONE BIT
1954 44      MOV     B,H ;COPY HA AND L TO A AND C
1955 85      ADD     L
1956 2A6B19  LHLD    DTEMP ;GET ADDRESS OF X
1959 95      SUB     L ;SUBTRACT FROM TEMPORARY RESULT
195A 4F      MOV     C,A
195B 78      MOV     A,B
195C 9C      SBB     H
195D 47      MOV     B,A
195E C5      PUSH    B ;SAVE TEMP RESULT IN STACK
195F D26419  JNC     DSKIP ;NO BORROW FROM SUBTRACT
1962 09      DAD     B ;ADD X BACK IN
1963 E3      XTHL   ;REPLACE TEMP RESULT ON STACK
1964 216D19  DSKIP: LXI     H,BNUM ;RESTORE H,L
1967 3F      CMC
1968 C34619  JMP     DLOOP ;REPEAT LOOP STEPS

```

```

196B      DTEMP: DS      2
196D      BNUM:  DS      1
;
MULP: ;MULTIPLY D,E BY H,L AND REPLACE H,L WITH RESULT
      MOV     B,H
      MOV     C,L ;COPY OF 1ST VALUE TO B,C FOR SHIFT AND ADD
      LXI     H,0 ;H,L IS THE ACCUMULATOR
MUL0: XRA     A
      MOV     A,B ;CARRY IS CLEARED
      RAR
      MOV     B,A
      MOV     A,C
      RAR
      MOV     C,A
      JC      MUL1 ;SKIP THIS ADD IF LSB IS ZERO
      ORA     B
      RZ     ;RETURN WITH H,L
      JMP     MUL2 ;SKIP ADD
MUL1: DAD     D ;ADD CURRENT VALUE OF D
MUL2: XCHG   ;READY FOR *2
      DAD     H
      XCHG
      JMP     MUL0
;
MULOP: ;MULTIPLY D,E BY H,L
      CALL    LODV2
      CALL    MULF
      JMP     ENDOP
;
DIVOP: ;DIVIDE H,L BY D,E
      CALL    DIVF
      XCHG
      JMP     ENDOP ;RESULT TO H,L
;
MODOP: CALL    DIVF
      JMP     ENDOP
;
SHLOP: CALL    SHFT ;CHECK VALUES
      SHL0: ORA     A ;DONE?
      JZ      ENDOP
      DAD     H ;HL=HL*2
      DCR     A
      JMP     SHL0
;
SHROP: CALL    SHFT
      SHR0: ORA     A ;DONE?
      JZ      ENDOP
      PUSH   PSW ;SAVE CURRENT COUNT
      XRA     A
      MOV     A,H

```

```

1985 1F      RAR
1986 67      MOV
1987 7D      MOV
1988 1F      RAR
1989 6F      MOV
198A F1      POP
198B 3D      DCR
198C C3AE19  JMP

;
198F CDEC18  ADDOP: CALL LODV2
19C2 19      ADD0: DAD   D
19C3 C3011A  JMP      ENDOP

;
19C6 CDEC18  SUBOP: CALL LODV2
19C9 EB      XCHG
19CA CD2D19  CALL
19CD C3C219  JMP      ADD0

;
19D0 C0CF18  NEGOP: CALL LODV1
19D3 CD2D19  NEG0: CALL  NEG
19D6 C3011A  JMP      ENDOP

;
19D9 C0CF18  NOTOP: CALL LODV1
19DC 23      INX
19DD C3D319  JMP      NEG0

;
19E0 CDEC18  ANDOP: CALL LODV2
19E3 7A      MOV      A,D
19E4 A4      ANA      H
19E5 67      MOV      H,A
19E6 7B      MOV      A,E
19E7 A5      ANA      L
19E8 6F      MOV      L,A
19E9 C3011A  JMP      ENDOP

;
19EC CDEC18  OROP: CALL LODV2
19EF 7A      MOV      A,D
19F0 B4      ORA      H
19F1 67      MOV      H,A
19F2 7B      MOV      A,E
19F3 B5      ORA      L
19F4 6F      MOV      L,A
19F5 C3011A  JMP      ENDOP

;
19F8 CDEC18  XOROP: CALL LODV2
19FB 7A      MOV      A,D
19FC AC      XRA      H
19FD 67      MOV      H,A
19FE 7B      MOV      A,E
19FF AD      XRA      L

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

;TREAT AS HL+(-DE)
;0-HL
;COMPUTE 0-HL
;65536-HL = 65535-(HL+1)

```

```

1A00 6F      MOV      L,A
1A01 C39318  ENDOP: JMP  STKV
;
;
;
ENDEXP: ;RETURNS ZERO FLAG IF SYMBOL IS CR, ;, OR .
LDA      TOKEN
CPI      SPECL
RNZ      ;NOT END IF NOT SPECIAL

;
LDA      ACCUM
CPI      CR
RZ      ;
CPI      ;
RZ      ;
CPI      ;
RZ      ;
CPI      ;
RET

;
OPAND: ;SCAN THE OPERAND FIELD OF AN INSTRUCTION
; (NOT A DB WITH FIRST TOKEN STRING > 2 OR 0)
XRA      A
STA      OSP ;ZERO OPERATOR STACK POINTER
STA      VSP
DCR      A ;255
STA      UNARY
LXI      H,0
SHLD     EVALUE

;
OP0: ;ARRIVE HERE WITH NEXT ITEM ALREADY SCANNED
CALL     ENDEXP ;DONE?
JNZ      OPI
;
EMPTY THE OPERATOR STACK
EMPOP: LXI      H,OSP
MOV      A,M ;GET THE OSP AND CHECK FOR EMPTY
ORA      A
JZ       CHKVAL ;JUMP IF EMPTY
DCR      M ;POP ELEMENT
MOV      E,A ;COPY FOR DOUBLE ADD
DCR      E
MVI      D,0
LXI      H,OPERV
DAD      D ;INDEXED - OPERV(OSP)
MOV      A,M ;GET OPERATOR
CALL     APPLY ;APPLY OPERATOR
JMP      EMPOP

;
CHKVAL:

```

```

1A04 3A8501  LDA
1A07 FE04    CPI
1A09 C0      RNZ

1A0A 3A8901  LDA
1A0D FE0D    CPI
1A0F C8      RZ
1A10 FE3B    CPI
1A12 C8      RZ
1A13 FE2C    CPI
1A15 C8      RZ
1A16 FE21    CPI
1A18 C9      RET

1A19 AF      XRA
1A1A 329118  STA
1A1D 329218  STA
1A20 3D      DCR
1A21 326C18  STA
1A24 210000  LXI
1A27 22C901  SHLD

1A2A CD041A  LDA
1A2D C25D1A  LDA

1A30 219118  LDA
1A33 7E      MOV
1A34 B7      ORA
1A35 CA481A  JZ
1A38 35      DCR
1A39 5F      MOV
1A3A 1D      DCR
1A3B 1600    MVI
1A3D 216D18  LXI
1A40 19      DAD
1A41 7E      MOV
1A42 CDF418  CALL
1A45 C3301A  JMP

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

1A48 3A9218 LDA VSP ;MUST HAVE ONE ELEMENT IT THE STACK
1A48 FE02 CPI 2
1A4D C4851B CNZ ERREX
1A50 3A3C01 LDA PBUFF
1A53 FE20 CPI
1A55 C0 RNZ ;EVALUE REMAINS AT ZERO
1A56 2A8118 LHLD VSTACK ;GET DOUBLE BYTE IN
1A59 22C981 SHLD EVALUE
1A5C C9 RET

```

```

;
OP1: ;MORE TO SCAN
LDA PBUFF
CPI
JNZ GETOP
LDA TOKEN
CPI STRNG ;IS THIS A STRING?
JNZ OP3

```

```

;
STRING - CONVERT TO DOUBLE PRECISION
LDA ACCLEN
ORA A
CZ ERREX ;ERROR IF LENGTH=0
CPI 3
CNC ERREX ;ERROR IF LENGTH>2
MVI D,0
LXI H,ACCUM
MOV E,M ;LSBYTE
INX H
DCR A ;A HAS THE LENGTH
JZ OP2 ;ONE OR TWO BYTES
MOV D,M ;FILL HIGH ORDER
OP2: XCHG ;VALUE TO H,L
JMP STNUM ;STORE TO STACK

```

```

;
OP3: ;NOT A STRING, CHECK FOR NUMBER
CPI NUMB
JNZ OP4
LHLD VALUE ;NUMERIC VALUE
JMP STNUM

```

```

;
OP4: ;NOT STRING OR NUMBER, MUST BE ID OR SPECL
CALL BGET ;BINARY SEARCH, GET ATTRIBUTES
JNZ OP6 ;MATCH?
YES, MAY BE OPERATOR
CPI OPER+1
JNC OP5

```

```

;
; OPERATOR ENCOUNTERED MS NIBBLE OF B IS PRIORITY NUMBER LS NIBBLE
; IS THE OPERATOR
; ACC HAS THE OPERATOR NUMBER, B HAS PRIORITY
CPI LPAR ;{?

```

```
1A9F FE0C
```

```

1AA1 4F MOV C,A ;SAVE COPY OF OPERATOR NUMBER
1AA2 3A6C18 LDA UNARY
1AA5 C2B51A JNZ OPER1 ;JUMP IF NOT A (
; ( ENCOUNTERED, UNARY MUST BE TRUE
ORA A
CZ ERREX
MVI A,OPFH
STA UNARY ;UNARY IS SET TRUE
MOV A,C ;RECOVER OPERATOR
JMP OPER4 ;CALLS STKO AND SETS UNARY TO TRUE
;
;

```

```

OP1: ;NOT A LEFT PAREN
ORA A
JNZ OPER6 ;MUST BE + OR - SINCE UNARY IS SET
;
;
UNARY NOT SET, MUST BE BINARY OPERATOR
OP2: ;COMPARE HIERARCHY OF TOS
PUSH B ;SAVE PRIORITY AND OPERATOR NUMBER
LDA OSP
ORA A
JZ OPER3 ;NO MORE OPERATORS IN STACK
MOV E,A ;OSP TO E
DCR E ;OSP-1
MVI D,0
LXI H,HIERV
DAD D ;HL ADDRESSES TOP OF OPERATOR STACK
MOV A,M ;PRIORITY OF TOP OPERATOR
CMP B ;CURRENT GREATER?
JC OPER3 ;JUMP IF SO
; APPLY TOP OPERATOR TO VALUE STACK
LXI H,OSP
MOV M,E ;OSP=OSP-1
LXI H,OPERV
DAD D
MOV A,M ;OPERATOR NUMBER TO ACC
CALL APPLY
POP B ;RESTORE OPERATOR NUMBER AND PRIORITY
JMP OPER2 ;FOR ANOTHER TEST
;
;
OP3: ;ARRIVE HERE WHEN OPERATOR IS STACKED
CHECK FOR RIGHT PAREN BALANCE
POP B ;OPERATOR NUMBER IN C, PRIORITY IN B
MOV A,C
CPI RPAR
JNZ OPER4 ;JUMP IF NOT A RIGHT PAREN
;
;
RIGHT PAREN FOUND, STACK MUST CONTAIN LEFT PAREN TO DELETE
LXI H,OSP
MOV A,M

```

```
1AB5 B7
1AB6 C20E1B
```

```

1AB9 C5
1ABA 3A9118
1ABD B7
1ABE CADE1A
1AC1 5F
1AC2 1D
1AC3 1600
1AC5 217718
1AC8 19
1AC9 7E
1ACA 88
1ACB DADE1A
1ACE 219118
1AD1 73
1AD2 216D18
1AD5 19
1AD6 7E
1AD7 CDF418
1ADA C1
1ADB C3B91A

```

```

1ADE C1
1ADF 79
1AE3 FE0D
1AE2 C2031B

```

```

1AE5 219118
1AE8 7E

```

COPY:AC VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

1AE9 B7      ORA      A      ;ZERO?
1AEA CAPCIA  JZ       LPERR   ;PAREN ERROR IF SO
1AED 3D      DCR      A      ;OSP-1
1AEE 77      MOV      M,A     ;STORED TO MEMORY
1AEF 5F      MOV      E,A
1AF0 1600    MVI      D,0
1AF2 216018 LXI      H,OPERV
1AF5 19      DAD      D
1AF6 7E      MOV      A,M     ;TOP OPERATOR IN REG-A
1AF7 FE0C    CPI      LPAR
1AF9 CAFF1A JZ       NLERR   ;JMP IF NO ERROR - PARENS BALANCE
1AFC CD851B LPERR: CALL ERREX
NLERR: ;ERROR REPORTING COMPLETE
1AFF AF      XRA      A
1B00 C3081B JMP      OPERS   ;TO CLEAR UNARY FLAG
;
1B03 CD0018 ;OPER4: ;ORDINARY OPERATOR
1B06 3EFF    CALL     STKO
1B08 326C18 MVI     A,0FFH ;TO SET UNARY FLAG
1B0B C37F1B OPERS: STA UNARY
JMP     GETOP ;FOR ANOTHER ELEMENT
;
1B0E 79      ;OPER6: ;UNARY SET, MUST BE + OR -
1B0F FE05    MOV     A,C     ;RECALL OPERATOR
1B11 CA7F1B CPI     PLUS
1B14 FE06    JZ      GETOP  ;IGNORE UNARY PLUS
1B16 C21E1B CPI     MINUS
1B19 3C      JNZ     CHKNOT
1B1A 4F      INR     A      ;CHANGE TO UNARY MINUS
1B1B C3B91A MOV     C,A
JMP     OPER2
CHKNOT: ;UNARY NOT SYMBOL?
1B1E FE08    CPI     NOTP
1B20 C4851B CNZ     ERREX
1B23 C3B91A JMP     OPER2
;
1B26 FE11    ;OPER5: ;ELEMENT FOUND IN TABLE, NOT AN OPERATOR
1B28 CC851B CPI     PT      ;PSEUDO OPERATOR?
1B2B 68      CZ      ERREX  ;ERROR IF SO
1B2C 2600    MOV     L,B    ;GET LOW VALUE TO L
1B2E C3711B MVI     H,0    ;ZERO HIGH ORDER BYTE
JMP     STNUM  ;STORE IT
;
1B31 3A8501 ;OPER6: ;NOT FOUND IN TABLE SCAN, $?
1B34 FE04    LDA     TOKEN
1B36 C2501B CPI     SPECL
1B39 3A8901 JNZ     OP7
1B3C FE24    LDA     ACCUM
CPI     '$'

```

CP/M VERSION \_\_\_\_\_

COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

```

1B3E CA4A1B
1B41 CD851B
1B44 210000
1B47 C3711B
1B4A 2AD201
1B4D C3711B

```

```

1B50 CD4613
1B53 CD4913
1B56 C2641B

```

```

1B59 3E50
1B5B CD1802
1B5E CD4C13
1B61 C36E1B
1B64 CD5213
1B67 E607
1B69 3E55
1B6B CC1802

```

```

1B6E CD5813

```

```

1B71 3A6C18
1B74 B7
1B78 CD851B
1B79 376C18
1B7C CD9318

```

```

1B7E C3261A

```

```

1B85 E5
1B86 3E45
1B88 CD1802
1B8B E1
1B8C C9

```

```

1B80 =
1B8D

```

```

JZ      CURPC ;USE CURRENT PC
CALL    ERREX
LXI     H,0
JMP     STNUM
CURPC:  LHLD   ASPC ;GET CURRENT PC
JMP     STNUM

```

```

;OP7: ;NOT $, LOOK IT UP

```

```

CALL    LOOKUP
CALL    FOUND
JNZ     FIDENT

```

```

; NOT FOUND IN SYMBOL TABLE, ENTER IF PASS 1

```

```

MVI     A,'P'
CALL    PERR
CALL    ENTER ;ENTER SYMBOL WITH ZERO TYPE FIELD
JMP     FIDES
FIDENT: CALL    GETTY ;TYPE TO H,L
ANI     111B
MVI     A,'U'
CZ      PERR

```

```

; FIDES: CALL    GETVAL ;VALUE TO H,L

```

```

; STNUM: ;STORE H,L TO VALUE STACK
LDA     UNARY
ORA     A
ERREX   ;UNARY OPERATION SET
STA     UNARY ;UNARY ENCOUNTERED WITH UNARY OFF
;SET TO OFF
CALL    STKV ;STACK THE VALUE

```

```

GETOP: CALL    SCAN

```

```

; ERREX: ;PUT 'E' ERROR IN OUTPUT BUFFER
PUSH   H
MVI     A,'E'
CALL    PERR
POP     H
RET

```

```

; ENDMOD EQU ($ AND 0FF0H) + 20H ;NEXT HALF PAGE
END

```

CP/M VERSION \_\_\_\_\_

COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

```

; CP/M RESIDENT ASSEMBLER MAIN PROGRAM
;
; COPYRIGHT (C) 1976
; GARY A. KILDALL
;

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

1BA0 = ORG 1BA0H
; MODULE ENTRY POINTS
0200 = IOMOD EQU 200H ;IO MODULE
1100 = SCMOD EQU 1100H ;SCANNER MODULE
1340 = SYMOD EQU 1340H ;SYMBOL TABLE MODULE
15A0 = BMOD EQU 15A0H ;BINARY SEARCH MODULE
1860 = OPMOD EQU 1860H ;OPERAND SCAN MODULE
;
0203 = SETUP EQU IOMOD+3H ;FILE SETUP FOR EACH PASS
0212 = PCON EQU IOMOD+12H ;WRITE CONSOLE BUFFER TO CR
0215 = WOBUFF EQU IOMOD+15H ;WRITE PRINT BUFFER AND REINITIALIZE
0218 = PERR EQU IOMOD+18H ;WRITE ERROR CHARACTER TO PRINT BUFFER
021B = DHEX EQU IOMOD+1BH ;SEND HEX CHARACTER TO MACHINE CODE FILE
021E = ECR EQU IOMOD+1EH ;END OF PROCESSING, CLOSE FILES AND TERMINATE
;
1103 = INITS EQU SCMOD+3H ;INITIALIZE SCANNER MODULE
1106 = SCAN EQU SCMOD+6H ;SCAN NEXT TOKEN
;
1343 = INISY EQU SYMOD+3H ;INITIALIZE SYMBOL TABLE
1346 = LOOKUP EQU SYMOD+6H ;LOOKUP SYMBOL IN ACCUMULATOR
1349 = FOUND EQU SYMOD+9H ;FOUND IF NZ FLAG
134C = ENTER EQU SYMOD+0CH ;ENTER SYMBOL IN ACCUMULATOR
134F = SETTY EQU SYMOD+0FH ;SET TYPE FIELD
1352 = GETTY EQU SYMOD+12H ;GET TYPE FIELD
1355 = SETVAL EQU SYMOD+15H ;SET VALUE FIELD
1358 = GETVAL EQU SYMOD+18H ;GET VALUE FIELD
;
15A6 = BGET EQU BMOD+6H ;BINARY SEARCH AND GET TYPE/VALUE PAIR
;
1863 = OPAND EQU OPMOD+3H ;GET OPERAND VALUE TO "EVALUE"
1866 = MULP EQU OPMOD+6H ;MULT D,E BY H,L TO H,L
1869 = DIVP EQU OPMOD+9H ;DIVIDE HL BY DE, RESULT TO DE
;
; COMMON EQUATES
0078 = PBMAX EQU 120 ;MAX PRINT SIZE
010C = PBUFP EQU 18CH ;PRINT BUFFER
0184 = PSP EQU PBUFP+PBMAX ;PRINT BUFFER POINTER
;
0185 = TOKEN EQU PBP+1 ;CURRENT TOKEN UDER SCAN
0186 = VALUE EQU TOKEN+1 ;VALUE OF NUMBER IN BINARY
0188 = ACCLEN EQU VALUE+2 ;ACCUMULATOR LENGTH
0040 = ACMAX EQU 64 ;MAX ACCUMULATOR LENGTH

```

```

0189 = ACCUM EQU ACCLEN+1
;
01C9 = EVALUE EQU ACCUM+ACMAX ;VALUE FROM EXPRESSION ANALYSIS
;
01CB = SYTOP EQU EVALUE+2 ;CURRENT SYMBOL TOP
01CD = SYMAX EQU SYTOP+2 ;MAX ADDRESS+1
;
01CF = PASS EQU SYMAX+2 ;CURRENT PASS NUMBER
01D0 = FPC EQU PASS+1 ;FILL ADDRESS FOR NEXT HEX BYTE
01D2 = ASPC EQU FPC+2 ;ASSEMBLER'S PSEUDO PC
01D4 = SYBAS EQU ASPC+2 ;BASE OF SYMBOL TABLE
01D6 = SYADR EQU SYBAS+2 ;CURRENT SYMBOL ADDRESS
;
; GLOBAL EQUATES
0001 = IDFN EQU 1 ;IDENTIFIER
0002 = NUMB EQU 2 ;NUMBER
0003 = STRNG EQU 3 ;STRING
0004 = SPFCL EQU 4 ;SPECIAL CHARACTER
;
0001 = PLABT EQU 0001B ;PROGRAM LABEL
0002 = DLABT EQU 0010B ;DATA LABEL
0004 = ECUT EQU 0100B ;EQUATE
0005 = SETT EQU 0101B ;SET
0006 = MACT EQU 0110B ;MACRO
;
0008 = EXPT EQU 1000B ;EXTERNAL
000B = REPT EQU 1011B ;REFER
000C = GLBT EQU 1100B ;GLOBAL
;
000D = CR EQU 0DH ;CARRIAGE RETURN
000A = LF EQU 0AH ;LINE FEED
001A = EOF EQU 1AH ;END OF FILE
0010 = NBMAX EQU 16 ;STARTING POSITION OF PRINT LINE
;
;
0010 = RT EQU 16 ;REGISTER TYPE
0011 = PT EQU RT+1 ;PSEUDO OPERATION
0005 = PENDIF EQU 5 ;PSEUDO OPERATOR "ENDIF"
0012 = OBASE EQU PT+1
0013 = O1 EQU OBASE+1 ;FIRST OPERATOR
0021 = O15 EQU OBASE+15;LAST OPERATOR
;
; MAIN STATEMENT PROCESSING LOOP
1BA0 AF XRA A
1BA1 32CF01 STA PASS ;SET TO PASS 0 INITIALLY
1BA4 CD4313 CALL INISY ;INITIALIZE THE SYMBOL TABLE
RESTART: CALL ;PASS LOOP GOES FROM 0 TO 1
CALL INITS ;INITIALIZE THE SCANNER
CALL SETUP ;SET UP THE INPUT FILE
LXI R,0

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

1B33 22E220      SHLD  SYLAB  ;ASSUME NO STARTING LABEL
1B33 22D001      SHLD  FPC
1B26 22D201      SHLD  ASPC
1B39 22ED20      SHLD  EPC      ;END PC
;
;SCNEXT:        ;SCAN THE NEXT INPUT ITEM
1B8C CD0611      CALL  SCAN
1B8F 3A8501      LDA   TOKEN
1B82 FE02        CPI   NUMB  ;SKIP LEADING NUMBERS FROM LINE EDITORS
1B84 CAB01B      JZ   SCNEXT
1B87 FE04        CPI   SPECL ;MAY BE PROCESSOR TECH'S COMMENT
1B89 C2DD1B      JNZ  SCN1
;
;SPECIAL CHARACTER, CHECK FOR *
1BCC 3A8901      LDA   ACCUM
1B8F FE2A        CPI   "*"
1BD1 C2311F      JNZ  CHEND  ;END OF LINE IF NOT *
;
;* FOUND, NO PRECEDING LABEL ALLOWED
1BD4 CD0020      CALL  SETLA
1BD7 C27C1F      JNZ  STERR  ;ERROR IF LABEL
1BDA C3521F      JMP  CHEN1  ;SCAN THE COMMENT OTHERWISE
;
;SCN1:         ;NOT NUMBER OR SPECIAL CHARACTER, CHECK FOR IDENTIFIER
1BD8 FE01        CPI   IDEN
1BDF C27C1F      JNZ  STERR  ;ERROR IF NOT
;
;IDENTIFIER FOUND, MAY BE LABEL, OPCODE, OR MACRO
1BE2 CDA615      CALL  BG0ET  ;BINARY SEARCH FIXED DATA
1BE5 CA301C      JZ   CHKPT  ;CHECK FOR PSEUDO OR REAL OPERATOR
;
;BINARY SEARCH WAS UNSUCCESSFUL, CHECK FOR MACRO
1BE8 CD4613      CALL  LOOKUP
1BE8 CD4913      CALL  FOUND
1BEE C2FE1B      JNZ  LFOUN  ;NZ FLAG SET IF FOUND
;
;NOT FOUND, ENTER IT
1BF1 CD4C13      CALL  ENTER ;THIS MUST BE PASS
1BF4 3ACF01      LDA   PASS
1BF7 B7          ORA   A
1BF8 C4D720      CNZ  ERRL  ;PHASE ERROR IF NOT
1BF8 C30C1C      JMP  SETSY  ;SET SYLAB
;
;ITEM WAS FOUND, CHECK FOR MACRO
1BFE CD5213      LFOUN: CALL  GETTY
1C31 FE06        CPI   MACT
1C03 C20C1C      JNZ  SETSY
;
;MACRO DEFINITION FOUND, EXPAND MACRO
1C86 CDE320      CALL  ERRL  ;NOT CURRENTLY IMPLEMENTED
1C09 C3521F      JMP  CHEN1  ;SCANS TO END OF CURRENT LINE
;

```

```

1C0C 2AEB20      LHL  SYLAB
1C0F 7D          MOV  A,L
1C10 B4          ORA  H
1C11 C40D20      CNZ  ERRL  ;LABEL ERROR IF NOT
1C14 2AD601      LHL  SYADR ;ADDRESS OF SYMBOL
1C17 22EB20      SHLD SYLAB ;MARK AS LABEL FOUND
;
1C1A CD0611      CALL  SCAN
1C1D 3A8501      LDA   TOKEN
1C20 FE04        CPI   SPECL
1C22 C20F1B      JNZ  SCN0  ;SKIP NEXT SCAN IF NOT SPECIAL
1C25 3A8901      LDA   ACCUM
1C28 FE3A        CPI   "*"
1C2A C20F1B      JNZ  SCN0
1C2D C30C1B      JMP  SCNEXT ;TO IGNORE "*"
;
1C30 FE11        CPI   PT
1C32 C2D71D      JNZ  CHKOT
;
1C35 58          MOV  E,B ;B HAS PARTICULAR OPERATOR NUMBER
1C36 1600        MVI  D,0 ;DOUBLE PRECISION VALUE TO D,E
1C38 1B          DCX  D ;BIASED BY +1
1C39 21431C      LXI  H,PTTAB ;BASE OF JUMP TABLE
1C3C 19          DAD  D
1C3D 19          DAD  D
1C3E 5E          MOV  E,M
1C3F 23          INX  H
1C40 66          MOV  H,M
1C41 6B          MOV  L,E
1C42 E9          PCHL ;JUMP INTO TABLE
;
1C43 5B1C        DW   SDB ;DB
1C45 A91C        DW   SDS ;DS
1C47 C01C        DW   SDW ;DW
1C49 DE1C        DW   SEND ;END
1C4B 151D        DW   SENDIF ;ENDIF
1C4D 181D        DW   SENDM ;ENDM
1C4F 1E1D        DW   SECU ;EQU
1C51 401D        DW   SIF ;IF
1C53 871D        DW   SMACRO ;MACRO
1C55 8D1D        DW   SORG ;ORG
1C57 A71D        DW   SSET ;SET
1C59 CE1D        DW   STITLE ;TITLE
;
SDB:

```

```

SETS: ;LABEL FOUND - IS IT THE ONLY ONE?
;
; LABEL FOUND, SCAN OPTIONAL "*"
CALL SCAN
LDA TOKEN
CPI SPECL
JNZ SCN0 ;SKIP NEXT SCAN IF NOT SPECIAL
LDA ACCUM
CPI "*"
JNZ SCN0
JMP SCNEXT ;TO IGNORE "*"
;
; BINARY SEARCH FOUND SYMBOL, CHECK FOR PSEUDO OR REAL OP
CHKPT: CPI PT ;PSEUDO OPCODE?
JNZ CHKOT
;
; PSEUDO OPCODE FOUND, BRANCH TO CASES
MOV E,B ;B HAS PARTICULAR OPERATOR NUMBER
MVI D,0 ;DOUBLE PRECISION VALUE TO D,E
DCX D ;BIASED BY +1
LXI H,PTTAB ;BASE OF JUMP TABLE
DAD D
DAD D
MOV E,M
INX H
MOV H,M
MOV L,E
PCHL ;JUMP INTO TABLE
;
PTTAB: ;PSEUDO OPCODE JUMP TABLE
DW SDB ;DB
DW SDS ;DS
DW SDW ;DW
DW SEND ;END
DW SENDIF ;ENDIF
DW SENDM ;ENDM
DW SECU ;EQU
DW SIF ;IF
DW SMACRO ;MACRO
DW SORG ;ORG
DW SSET ;SET
DW STITLE ;TITLE
;
SDB:

```

CP/M VERSION \_\_\_\_\_  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # \_\_\_\_\_

```

1C5B CD0A20      CALL    FILAB    ;SET LABEL FOR THIS LINE TO ASPC
;
SDB0:           CALL    SCAN    ;PAST DB TO NEXT ITEM
                LDA     TOKEN    ;LOOK FOR LONG STRING
                CPI     STRNG
                JNZ     SDBC     ;SKIP IF NOT STRING
                LDA     ACCLEN
                DCR     A        ;LENGTH 1 STRING?
                JZ      SDBC
;
                LENGTH 0,2,... STRING
                MOV     B,A
                INR     B
                INR     B        ;BECOMES 1,3,... FOR 0,2,... LENGTHS
                LXI     H,ACCUM   ;ADDRESS CHARACTERS IN STRING
SDB1:           DCR     B        ;COUNT DOWN TO ZERO
                JZ      SDB2     ;SCAN DELIMITER AT END OF STRING
                PUSH    B        ;SAVE COUNT
                MOV     B,M      ;GET CHARACTER
                INX     H
                PUSH    H        ;SAVE ACCUM POINTER
                CALL    FILHB    ;SEND TO HEX FILE
                POP     H
                POP     B
                JMP     SDB1
SDB2:           CALL    SCAN    ;TO THE DELIMITER
                JMP     SDB3
;
;
                NOT A LONG STRING
SDBC:           CALL    OPAND    ;COMPUTE OPERAND
                LHL    EVALUE    ;VALUE TO H,L
                MOV     A,H
                ORA     A        ;HIGH ORDER MUST BE ZERO
                MOV     B,L      ;GET LOW BYTE
                CALL    FILHB
SDB3:           ;END OF ITEM - UPDATE ASPC
                CALL    SETAS    ;SET ASPC TO FPC
                CALL    DELIM
                CPI     ' '
                JZ      SDB0     ;FOR ANOTHER ITEM
                JMP     CHEND    ;CHECK END OF LINE SYNTAX
;
SDS:           CALL    FILAB    ;HANDLE LABEL IF IT OCCURRED
                CALL    PADD     ;PRINT ADDRESS
                CALL    EXP16    ;SCAN AND GET 16BIT OPERAND
                XCHG
                LHL    ASPC     ;CURRENT PSEUDO PC
                DAD     D        ;+EXPRESSION
                SHLD   ASPC

```

```

1C8A 22D001      SHLD   FPC    ;NEXT TO FILL
1CBD C3311F      JMP     CHEND

```

```

;
SDW:           CALL    FILAB    ;HANDLE OPTIONAL LABEL
SDW0:          CALL    EXP16    ;GET 16BIT OPERAND
                PUSH    H        ;SAVE A COPY
                MOV     B,L      ;LOW BYTE FIRST
                CALL    FILHB    ;SEND LOW BYTE
                POP     H        ;RECLAIM A COPY
                MOV     B,H      ;HIGH BYTE NEXT
                CALL    FILHB    ;SEND HIGH BYTE
                CALL    SETAS    ;SET ASPC=FPC
                CALL    DELIM    ;CHECK DELIMITER SYNTAX
                CPI     ' '
                JZ      SDW0     ;GET MORE DATA
                JMP     CHEND
;
SEND:          CALL    FILAB
                CALL    PADD     ;WRITE LAST LOC
                LDA     PBUFF
                CPI     ' '
                JNZ     CHEND
                CALL    EXP16    ;GET EXPRESSION IF IT'S THERE
                LDA     PBUFF
                CPI     ' '
                JNZ     SEND0
                SHLD   EPC      ;EXPRESSION FOUND, STORE IT FOR LATER
                MVI     A,' '
                STA     PBUFF    ;CLEAR ERROR, IF IT OCCURRED
                CALL    SCAN    ;CLEAR CR
                LDA     TOKEN
                CPI     SPECL
                JNZ     STERR
                LDA     ACCUM
                CPI     LF
                JNZ     STERR
                JMP     ENDAS    ;END OF ASSEMBLER
;
SENDIF:        JMP     POEND
;
SENDM:         CALL    ERN     ;
                JMP     POEND
;
SEQU:         CALL    SETLA

```

```

1CC0 CD0A20
1CC3 CDD11E
1CC6 E5
1CC7 45
1CC8 CD4820
1CCB E1
1CCC 44
1CCD CD4820
1CD0 CDF91F
1CD3 CDBA1E
1CD6 FE2C
1CD8 CAC31C
1CDB C3311F

```

```

1CDE CD0A20
1CE1 CDA620
1CE4 3A0C01
1CE7 FE20
1CE9 C2311F
1CEC CDD11E
1CEF 3A0C01
1CF2 FE20
1CF4 C2FA1C
1CF7 22ED20
1CFA 3E20
1CFC 32CC01
1CFF CD0611
1D02 3A8501
1D05 FE04
1D07 C27C1F
1D0A 3A8901
1D0D FE0A
1D0F C27C1F
1D12 C36B1F

```

```

1D15 C3D11D
1D18 CDE320
1D1B C3D11D
1D1E CD0020

```

CP/M VERSION \_\_\_\_\_  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # \_\_\_\_\_

```

1021 CA7C1F      JZ      STERR  ;MUST BE A LABEL
1024 2AD201     LHLD   ASPC   ;HOLD TEMP ASPC
1027 E5         PUSH  H      ;IN STACK
1028 C0D11E     CALL   EXP16  ;GET 16BIT OPERAND
102B 22D201     SHLD  ASPC   ;VALUE OF EXPRESSION
102E C09A20     CALL   FILAB
1031 CDA920     CALL   PADDR  ;COMPUTED VALUE
1034 211201     LXI   H,PBUFF+6 ;SPACE AFTER VALUE
1037 363D      MVI   M," "
1039 E1         POP   H      ;REAL ASPC
103A 22D201     SHLD  ASPC   ;CHANGE BACK
103D C3311F     JMP   CHEND

```

```

;
;SIF:

```

```

1040 CD2A20     CALL   FILAB  ;IN CASE OF LABEL
1043 C0D11E     CALL   EXP16  ;GET IF EXPRESSION
1046 3A8C01     LDA   PBUFF
1049 FE20      CPI
104B C2311F     JNZ   CHEND  ;SKIP IF ERROR
104E 7D        MOV   A,L    ;GET LSB
104F 1F        RAR
1050 DA311F     JC    CHEND  ;TRUE IF CARRY BIT SET

```

```

;
;SIF0:

```

```

1053 C0B611     CALL   SCAN
1056 3A8501     LDA   TOKEN
1059 FE34      CPI   SPECL
105B C26E1D     JNZ   SIF1
105E 3A9901     LDA   ACCUM
1061 FE1A      CPI   EOP
1063 3E42      MVI   A,"B"  ;BALANCE ERROR
1065 CC1602     CZ    PERR
1068 CA8B1F     JZ    ENDAS
106B C3531D     JMP   SIF0   ;FOR ANOTHER

```

```

;SIF1: ;NOT A SPECIAL CHARACTER

```

```

106E FE01      CPI   IDEN
1070 C2531D     JNZ   SIF0   ;NOT AN IDENTIFIER
1073 CDA615     CALL  BGET   ;LOOK FOR ENDIF
1076 C2531D     JNZ   SIF0   ;NOT FOUND
1077 FE11      CPI   PT     ;PSEUDO OP?
107B C2531D     JNZ   SIF0
107E 78        MOV   A,B    ;GET OPERATOR NUMBER
107F FE05      CPI   PENDIF ;ENDIF?
1081 C2531D     JNZ   SIF0   ;GET ANOTHER TOKEN
1084 C3D11D     JMP   POEND  ;OK, CHECK END OF LINE

```

```

;SMACRO:

```

```

1087 CDE320     CALL  ERRN
108A C3311F     JMP   CHEND

```

```

108D CDD11E     SORG: CALL EXP16
1090 3A0C01     LDA   PBUFF
1093 FE20      CPI
1095 C2311F     JNZ   CHEND ;SKIP ORG IF ERROR
1098 22D201     SHLD  ASPC  ;CHANGE PC
109B 22D001     SHLD  FPC  ;CHANGE NEXT TO FIC/M VERSION
109E C09A20     CALL  FILAB ;IN CASE OF LABEL
1DA1 CDA620     CALL  PADD
1DA4 C3311F     JMP   CHEND

```

```

;SSET:

```

```

1DA7 CD0020     CALL  SETLA
1DAA CA7C1F     JZ    STERR ;MUST BE LABELLED

```

```

;

```

```

1DAD CD5213     CALL  GETTY
1DB0 FE05      CPI   SETT
1DB2 C4DD20     CNZ  ERRL  ;LABEL ERROR
1DB5 3E05      MVI   A,SETT
1DB7 C04F13     CALL  SETTY ;REPLACE TYPE WITH "SET"
1DBA CDD11E     CALL  EXP16 ;GET THE EXPRESSION
1DBD E5        PUSH  H
1DBE CD0020     CALL  SETLA ;SAVE IT
1DC1 E1        POP   H     ;RECLAIM IT
1DC2 CD5513     CALL  SETVAL
1DC5 210000     LXI   H,0
1DC8 22EB20     SHLD  SYLAB ;PREVENT LABEL PROCESSING
1DCB C3311F     JMP   CHEND

```

```

;

```

```

;STITLE:

```

```

1DCE CDE320     CALL  ERRN  ;NOT IMPLEMENTED

```

```

;POEND: ;PSEUDO OPERATOR END - SCAN TO NEXT TOKEN

```

```

1DD1 CD0611     CALL  SCAN
1DD4 C3311F     JMP   CHEND

```

```

;CHKOT: ;NOT A PSEUDO OPCODE, CHECK FOR REAL OPCODE

```

```

1DD7 D613      SUI   O1    ;BASE OF OPCODES
1DD9 FE21      CPI   O15   ;PAST LAST OPCODE?
1DDB D27C1F     JNC  STERR ;STATEMENT ERROR IF SO

```

```

;

```

```

;FOUND OPCODE, COMPUTE INDEX INTO TABLE AND JUMP TO CASE

```

```

1DDE 5F        MOV   E,A
1DDF 1600     MVI   D,0
1DE1 21EB1D     LXI  H,OPTAB
1DE4 19        DAD  D
1DE5 19        DAD  D
1DE6 5E        MOV  E,M
1DE7 23        INX  H

```

CP/M VERSION \_\_\_\_\_  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # \_\_\_\_\_



```

1DE8 66      MOV      H,M
1DE9 6B      MOV      L,E
1DEA E9      PCHL           ;JUMP TO CASE

```

```

;
;OPTAB: ;OPCODE CATEGORIES
1DEB 891E    DW      SSIMP ;SIMPLE
1DED 121E    DW      SLXI  ;LXI
1DEF 1E1E    DW      SDAD  ;DAD
1DF1 241E    DW      SPUSH ;PUSH/POP
1DF3 381E    DW      SJMP  ;JMP/CALL
1DF5 411E    DW      SMOV  ;MOV
1DF7 501E    DW      SMVI  ;MVI
1DF9 601E    DW      SACCI ;ACCUM IMMEDIATE
1DFB 691E    DW      SLDAX ;LDAX/STAX
1DFD 781E    DW      SLHLD ;LHLD/SHLD/LDA/STA
1DFE 811E    DW      SACCR ;ACCUM-REGISTER
1E21 881E    DW      SINC  ;INC/DCR
1E23 8F1E    DW      SINX  ;INX/DCX
1E25 9E1E    DW      SKST  ;RESTART
1E27 A51E    DW      SIN   ;IN/OUT

```

```

;
;SSIMP: ;SIMPLE OPERATION CODES
1E09 CD4820  CALL   FILHB ;SEND HEX VALUE TO MACHINE CODE FILE
1E0C CD0611  CALL   SCAN  ;TO NEXT TOKEN
1E0F C3B11E  JMP    INCP

```

```

;
;SLXI: ;LXI H,16B
1E12 CDFC1E  CALL   SHDREG ;SCAN DOUBLE PRECISION REGISTER
1E15 CD171F  CALL   CHCOM  ;CHECK FOR COMMA FOLLOWING REGISTER
1E18 CD111F  CALL   SETADR ;SCAN AND EMIT DOUBLE PRECISION OPERAND
1E1B C3B11E  JMP    INCP

```

```

;
;SDAD: ;DAD B
1E1E CDFC1E  CALL   SHDREG ;SCAN AND EMIT DOUBLE PRECISION REGISTER
1E21 C3B11E  JMP    INCP

```

```

;
;SPUSH: ;PUSH B POP D
1E24 CDF21E  CALL   SHREG  ;SCAN SINGLE PRECISION REGISTER TO A
1E27 FE38    CPI    111000B ;MAY BE PSW
1E29 CA311E  JZ     SPUB

```

```

;
;SPUB: ;NOT PSW, MUST BE B,D, OR H
1E2C E688    ANI   001000B ;LOW BIT MUST BE 0
1E2E C48D20  CNZ   ERRR   ;REGISTER ERROR IF NOT
1E31 79      MOV   A,C    ;RECALL REGISTER AND MASK IN CASE OF ERROR
1E32 E630    ANI   110000B ;MASK IN OPCODE FOR PUSH OR POP
1E34 E8      ORA   B      ;FILL HEX VALUE AND INCREMENT PC
1E35 C3AE1E  JMP   FILINC

```

```

;
;SJMP: ;JMP 16B/ CALL 16B
1E38 CD4820  CALL   FILHB ;EMIT JMP OR CALL OPCODE

```

CP/M VERSION \_\_\_\_\_  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # \_\_\_\_\_

```

1E3B CD111F  CALL   SETADR ;EMIT 16BIT OPERAND
1E3E C3B11E  JMP    INCP

```

```

;
;SMOV: ;MOV A,B
1E41 CDF21E  CALL   SHREG
1E44 B0      ORA   B      ;MASK IN OPCODE
1E45 47      MOV   B,A    ;SAVE IN B TEMPORARILY
1E46 CD171F  CALL   CHCOM  ;MUST BE COMMA SEPARATOR
1E49 CDE71E  CALL   EXP3   ;VALUE MUST BE 0-7
1E4C B0      ORA   B      ;MASK IN OPCODE
1E4D C3AE1E  JMP   FILINC

```

```

;
;SMVI: ;MVI A,8B
1E50 CDF21E  CALL   SHREG
1E53 B0      ORA   B      ;MASK IN OPCODE
1E54 CD4720  CALL   FILHEX ;EMIT OPCODE
1E57 CD171F  CALL   CHCOM  ;SCAN COMMA
1E5A CD0B1F  CALL   SETBYTE ;EMIT 8BIT VALUE
1E5D C3B11E  JMP    INCP

```

```

;
;SACCI: ;ADI 8B
1E60 CD4820  CALL   FILHB ;EMIT IMMEDIATE OPCODE
1E63 CD0B1F  CALL   SETBYTE ;EMIT 8BIT OPERAND
1E66 C3B11E  JMP    INCP

```

```

;
;SLDAX: ;LDAX B/STAX D
1E69 CDF21E  CALL   SHREG
1E6C E628    ANI   101000B ;MUST BE B OR D
1E6E C48D20  CNZ   ERRR   ;REGISTER ERROR IF NOT
1E71 79      MOV   A,C    ;RECOVER REGISTER NUMBER
1E72 E610    ANI   010000B ;CHANGE TO B OR D IF ERROR
1E74 B0      ORA   B      ;MASK IN OPCODE
1E75 C3AE1E  JMP   FILINC ;EMIT OPCODE

```

```

;
;SLHLD: ;LHLD 16B/ SHLD 16B/ LDA 16B/ STA 16B
1E78 CD4820  CALL   FILHB ;EMIT OPCODE
1E7B CD111F  CALL   SETADR ;EMIT OPERAND
1E7E C3B11E  JMP    INCP

```

```

;
;SACCR: ;ADD B
1E81 CDE71E  CALL   EXP3   ;RIGHT ADJUSTED 3BIT VALUE FOR REGISTER
1E84 B0      ORA   B      ;MASK IN OPCODE
1E85 C3AE1E  JMP   FILINC

```

```

;
;SINC: ;INR B/DCR D
1E88 CDF21E  CALL   SHREG ;GET REGISTER
1E8B B0      ORA   B
1E8C C3AE1E  JMP   FILINC

```

```

;
;SINX: ;INX H/DCX B

```

CP/M VERSION \_\_\_\_\_  
COPYRIGHT © 1976  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA. 93950  
SER. # \_\_\_\_\_

```

1E8F CDF21E      CALL  SHREG
1E92 E608        ANI   001000B ;MUST BE B D M OR SP
1E94 C4BD20      CNZ   ERRR   ;REGISTER ERROR IF NOT
1E97 79          MOV   A,C    ;RECOVER REGISTER
1E98 E630        ANI   110000B ;IN CASE OF ERROR
1E9A B0          ORA   B      ;MASK IN OPCODE
1E9B C3AE1E      JMP   FILINC

```

```

;
SRST: ;RESTART 4
      CALL  SHREG ;VALUE IS 0-7
      ORA   B    ;OPCODE MASKED
      JMP   FILINC

```

```

;
SIN:  ;IN 8B/OUT 8B
      CALL  FILHB ;EMIT OPCODE
      CALL  SETBYTE ;EMIT 8BIT OPERAND
      JMP   INPC

```

```

;
FILINC: ;FILL HEX VALUE FROM A BEFORE INCREMENTING PC
        CALL  FILHEX

```

```

;
INPC:  ;CHANGE ASSEMBLER'S PSEUDO PROGRAM COUNTER
        CALL  FILAB ;SET ANY LABELS WHICH OCCUR ON THE LINE
        CALL  SETAS ;ASPC=FPC
        JMP   CHEND ;END OF LINE SCAN

```

```

1EA5 CD4820
1EA8 CD831F
1EAB C3B11E

```

```

1EAE CD4720

```

```

;
;
;
UTILITY SUBROUTINES FOR OPERATION CODES

```

```

;
DELIM: ;CHECK DELIMITER SYNTAX FOR DATA STATEMENTS

```

```

1EBA 3A8501      LDA   TOKEN
1EBD FE34        CPI   SPECL
1EBF C4D120      CNZ   ERRD
1EC2 3A8901      LDA   ACCUM
1EC5 FE2C        CPI   ' '
1EC7 C8          RZ
1EC8 FE33        CPI   ' '
1ECA C8          RZ
1ECB FE8D        CPI   CR
1ECD C4D120      CNZ   ERRD
1ED8 C9          RET

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

;
EXP16: ;GET 16BIT VALUE TO H,L
        PUSH  B
        CALL  SCAN ;START SCANNING OPERAND FIELD
        CALL  OPAND
        LHLD  EVALUE ;VALUE TO H,L
        POP  B
        RET

```

```

1ED1 C5
1ED2 CD0611
1ED5 CD6318
1ED8 2AC901
1EDB C1
1EDC C9

```

```

1EDD CDD11E
1EE0 7C
1EE1 B7
1EE2 C4C720
1EE5 7D
1EE6 C9

```

```

EXP8: ;GET 8BIT VALUE TO REG A
      CALL  EXP16
      MOV   A,H
      ORA   A
      CNZ   ERRV ;VALUE ERROR IF HIGH BYTE NOT ZERO
      MOV   A,L
      RET

```

```

1EE7 CDD11E
1EEA FE08
1EEC D4C720
1EEF E607
1EF1 C9

```

```

;
EXP3: ;GET 3BIT VALUE TO REG A
      CALL  EXP8
      CPI   8
      CNC  ERRV ;VALUE ERROR IF >=8
      ANI  111B ;REDUCE IF ERROR OCCURS
      RET

```

```

1EF2 CDE71E
1EF5 17
1EF6 17
1EF7 17
1EF8 E638
1EFA 4F
1EFB C9

```

```

;
SHREG: ;GET 3BIT VALUE AND SHIFT LEFT BY 3
        CALL  EXP3
        RAL
        RAL
        RAL
        ANI  111000B
        MOV  C,A ;COPY TO C
        RET

```

```

1EFC CDF21E
1EFF E608
1F01 C4BD20
1F04 79
1F05 E630
1F07 B0
1F08 C34720

```

```

;
SHDREG: ;GET DOUBLE REGISTER TO A
        CALL  SHREG
        ANI  001000B ;CHECK FOR A,C,E, OR L
        ERRR ;REGISTER ERROR
        MOV  A,C ;RECOVER REGISTER
        ANI  110000B ;FIX IT IF ERROR OCCURRED
        ORA  B ;MASK OPCODE
        JMP  FILHEX ;EMIT IT

```

```

1F0B CDD11E
1F0E C34720

```

```

;
SETBYTE: ;EMIT 16BIT OPERAND
         CALL  EXP8
         JMP  FILHEX

```

```

1F11 CDD11E
1F14 C37420

```

```

;
SETADR: ;EMIT 16BIT OPERAND
        CALL  EXP16
        JMP  FILADR

```

```

1F17 F5
1F18 C5
1F19 3A8501
1F1C FE04
1F1E C2291F

```

```

;
CHCOM: ;CHECK FOR COMMA FOLLOWING EXPRESSION
        PUSH  PSW
        PUSH  B
        LDA  TOKEN
        CPI  SPECL
        JNZ  COMER
        SPECIAL CHARACTER, CHECK FOR COMMA
        LDA  ACCUM
        CPI  ' '

```

```

1F21 3A8901
1F24 FE2C

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

1F26 CA2E1F      JZ      COMRET ;RETURN IF COMMA POUND
COMER: ;COMMA ERROR
1F29 3E43      MVI     A,'C'
1F2B CD1882     CALL    PERR
COMRET:
1F2E C1        POP     B
1F2F F1        POP     PSW
1F30 C9        RET

```

```

;
CHEND: ;END OF LINE CHECK
1F31 CD9A28     CALL    FILAB ;IN CASE OF A LABEL
1F34 3A9501     LDA     TOKEN
1F37 FE34      CPI     SPECL
1F39 C27C1F     JNZ     STERR ;MUST BE A SPECIAL CHARACTER
1F3C 3A8921     LDA     ACCUM
1F3F FE3D      CPI     CR ;CARRIAGE RETURN
1F41 C24A1F     JNZ     CHEN0
;
CARRIAGE RETURN FOUND, SCAN PICKS UP LF AND PUSHES LINE
1F44 CD0611     CALL    SCAN
1F47 C3BC1B     JMP     SCNEXT

```

```

;
CHEN0: ;NOT CR, CHECK FOR COMMENT
1F4A FE3B      CPI     ' '
1F4C C2721F     JNZ     CHEN2
1F4F CD9A28     CALL    FILAB ;IN CASE LABELLED EMPTY LINE
;
CLEAR COMMENT TO END OF LINE

```

```

CHEN1: CALL    SCAN
1F52 CD0611     LDA     TOKEN
1F55 3A8501     CPI     SPECL
1F58 FE94      CPI     SPECL
1F5A C2521F     JNZ     CHEN1
1F5D 3A8921     LDA     ACCUM
1F60 FE8A      CPI     LF
1F62 CA9C1B     JZ      SCNEXT
1F65 FE1A      CPI     EOF
1F67 CA651F     JZ     ENDAS ;END OF ASSEMBLY IF EOF
1F6A FE21     CPI     ' '
1F6C CA9C1B     JZ     SCNEXT ;LOGICAL END OF LINE
1F6F C3521F     JMP     CHEN1 ;NONE OF THE ABOVE

```

```

;
NOT CR OR LF, MAY BE LOGICAL END OF LINE
CHEN2: CPI     ' '
1F72 FE21     JZ     SCNEXT
1F74 CA9C1B     JZ     SCNEXT
1F77 FE1A      CPI     EOF
1F79 CA9B1F     JZ     ENDAS

```

```

;
STATEMENT ERROR IN OPERAND FIELD
STERR: MVI     A,'S'
1F7E CD1882     CALL    PERR
1F81 C3521F     JMP     CHEN1 ;TO DUMP LINE

```

```

1F84 7B      MOV     A,E
1F85 95      SUB     L
1F86 6F      MOV     L,A
1F87 7A      MOV     A,D
1F88 9C      SBB     H
1F89 67      MOV     H,A
1F8A C9      RET

```

```

1F8B 21CF01   LXI     H,PASS
1F8E 7E      MOV     A,M
1F8F 34      INR     M ;PASS NUMBER INCREMENTED
1F90 B7      ORA     A
1F91 CAA71B   JZ      RESTART
1F94 CD0611   CALL    SCAN ;TO CLEAR LAST LINE FEED
1F97 CDA620   CALL    PADD ;WRITE LAST ADDRESS
1F9A 211101   LXI     H,PBUFF+5
1F9D 3600     MVI     M,CR ;SET TO CR FOR END OF MESSAGE
1F9F 210D01   LXI     H,PBUFF+1
1FA2 CD1202   CALL    PCON ;PRINT LAST ADDRESS

```

```

;
COMPUTE REMAINING SPACE
1FA5 2ACB01   LHL    SYTOP
1FA8 E8      XCHG
1FA9 2AD401   LHL    SYBAS
1FAC CD841F   CALL    DIFF ;DIFFERENCE TO H,L
1FAF E5      PUSH   H ;SYTOP-SYBAS TO STACK
1FB0 2ACD01   LHL    SYMAX
1FB3 EB      XCHG
1FB4 2AD401   LHL    SYBAS
1FB7 CD841F   CALL    DIFF ;SYMAX-SYBAS TO H,L
1FBA 5C      MOV     E,H
1FBB 1600     MVI     D,0 ;DIVIDED BY 256
1FBD E1      POP     H ;SYTOP-SYBAS TO H,L
1FBE CD6918   CALL    DIVF ;RESULT TO DE
1FC1 EB      XCHG
1FC2 CDA920   CALL    PADDR ;PRINT H,L TO PBUFF
1FC5 211101   LXI     H,PBUFF+5 ;MESSAGE
1FC8 11D61F   LXI     D,EMSG ;END MESSAGE
ENDAS: LDAX   D
1FCB 1A      ORA     A ;ZERO?
1FCC B7      JZ     ENDA1
1FCD CAE41F   MOV     M,A
1FD0 77      MOV     H,A
1FD1 23      INX     H
1FD2 13      INX     D
1FD3 C3CB1F   JMP     ENDA0

```

```

1FD6 4820555345MSG: DB 'H USE FACTOR',CR,0

```

```
DIFF: ;COMPUTE DE-HL TO HL
```

```

MOV     A,E
SUB     L
MOV     L,A
MOV     A,D
SBB     H
MOV     H,A
RET

```

```
;
ENDAS: ;END OF ASSEMBLY FOR THIS PASS
```

```

LXI     H,PASS
MOV     A,M
INR     M ;PASS NUMBER INCREMENTED
ORA     A
JZ      RESTART
CALL    SCAN ;TO CLEAR LAST LINE FEED
CALL    PADD ;WRITE LAST ADDRESS
LXI     H,PBUFF+5
MVI     M,CR ;SET TO CR FOR END OF MESSAGE
LXI     H,PBUFF+1
CALL    PCON ;PRINT LAST ADDRESS

```

```
;
COMPUTE REMAINING SPACE
```

```

LHL    SYTOP
XCHG
LHL    SYBAS
CALL    DIFF ;DIFFERENCE TO H,L
PUSH   H ;SYTOP-SYBAS TO STACK
LHL    SYMAX
XCHG
LHL    SYBAS
CALL    DIFF ;SYMAX-SYBAS TO H,L
MOV     E,H
MVI     D,0 ;DIVIDED BY 256
POP     H ;SYTOP-SYBAS TO H,L
CALL    DIVF ;RESULT TO DE
XCHG
CALL    PADDR ;PRINT H,L TO PBUFF
LXI     H,PBUFF+5 ;MESSAGE
LXI     D,EMSG ;END MESSAGE
ENDAS: LDAX   D
ORA     A ;ZERO?
JZ     ENDA1
MOV     M,A
MOV     H,A
INX     H
INX     D
JMP     ENDA0

```

```
ENDAS: LDAX   D
ORA     A ;ZERO?
JZ     ENDA1
MOV     M,A
MOV     H,A
INX     H
INX     D
JMP     ENDA0
```

```
1FD6 4820555345MSG: DB 'H USE FACTOR',CR,0
```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

CP/M VERSION _____
COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA. 93950
SER. # _____

```

```

1FE4 21E001  ENDAL: LXI  H, PBUFF+2 ;BEGINNING OF RATIO
1FE7 CD1202  CALL  PCON
1FEA 2AED20  LHL  EPC
1FED 22D001  SHLD FPC ;END PROGRAM COUNTER
1FF0 C31E02  JMP  EOR

```

```

1FF3 7A
1FF4 BC
1FF5 C0
1FF6 7B
1FF7 B0
1FF8 C9

```

```

;
; UTILITY SUBROUTINES
; COMDH: ;COMPARE D,E WITH H,L FOR EQUALITY (NZ FLAG IF NOT EQUAL)

```

```

MOV  A,D
CMP  H
RNZ  H
MOV  A,E
CMP  L
RET

```

```

; SETAS: ;ASPC=FPC

```

```

1FF9 2AD001  LHL  FPC
1FFC 22D201  SHLD ASPC
1FFF C9  RET

```

```

; SETLA: ;SYADR=SYLAB, FOLLOWED BY CHECK FOR ZERO

```

```

2000 2AEB20  LHL  SYLAB
2003 22D601  SHLD SYADR
2006 CD4913  CALL  FOUND
2009 C9  RET

```

```

; FILAB: ;FILL LABEL VALUE WITH CURRENT ASPC, IF LABEL FOUND

```

```

200A CD0020  CALL  SETLA
200D C8  RZ ;RETURN IF NO LABEL DETECTED

```

```

; LABEL FOUND, MUST BE DEFINED ON PASS-1
; TO MARK NEXT STATEMENT WITH NO LABEL

```

```

200E 21E000  LXI  SYLAB
2011 22E320  SHLD SYLAB ;TO MARK NEXT STATEMENT WITH NO LABEL
2014 3ACF01  LDA  PASS
2017 B7  ORA  A
2018 C23120  JNZ  FIL1

```

```

; PASS 0

```

```

201B CD5213  CALL  GETTY
201E F5  PUSH PSW ;SAVE A COPY OF TYPE
201F E607  ANI  111B ;CHECK FOR UNDEFINED
2021 C4DD20  CNZ  ERR1 ;LABEL ERROR
2024 F1  POP  PSW ;RESTORE TYPE
2025 F601  ORI  PLABT ;SET TO LABEL TYPE
2027 CD4F13  CALL  SETTY ;SET TYPE FIELD
202A 2AD201  LHL  ASPC ;GET CURRENT PC
202D CD5513  CALL  SETVAL ;PLACE INTO VALUE FIELD
2030 C9  RET

```

```

; FIL1: ;CHECK FOR DEFINED VALUE

```

```

2031 CD5213
2034 E607
2036 CCD720

```

```

2039 CD5813
203C E8
203D 2AD201
2040 CDF31F
2043 C4D720
2046 C9

```

```

2047 47
2048 3ACF01
204B B7
204C 78
204D CA6C20

```

```

2050 C5
2051 CD1B02

```

```

2054 3A0D01
2057 FE20
2059 2AD201
205C CCA920

```

```

205F 3AEF20
2062 FE10
2064 C1
2065 D26C20

```

```

2068 78
2069 CD9620
206C 2AD001
206F 23
2070 22D001
2073 C9

```

```

2074 E5
2075 45
2076 CD4820
2079 E1
207A 44
207B C34820

```

```

207E C630

```

```

CALL  GETTY
ANI  111B
CZ  ERRP ;PHASE ERROR
GET VALUE AND COMPARE WITH ASPC
CALL  GETVAL ;TO H,L
XCHG
LHL  ASPC
CALL  COMDH
CNZ  ERRP ;PHASE ERROR IF NOT THE SAME
RET

```

```

; FILHEX: ;WRITE HEX BYTE IN REGISTER A TO MACHINE CODE FILE IF PASS

```

```

MOV  B,A
FILHB: LDA  PASS
ORA  A
MOV  A,B
JZ  FILHI

```

```

; PASS - 1, WRITE HEX AND PRINT DATA
; PUSH B ;SAVE A COPY
CALL  DHEX ;INTO MACHINE CODE FILE
; MAY BE COMPLETELY EMPTY LINE, SO CHECK ADDRESS

```

```

LDA  PBUFF+1
CPI  ' '
LHL  ASPC
CZ  PADDR ;PRINT ADDRESS FIELD

```

```

; LDA  NBP
; CPI  NBMAX ;TRUNCATE CODE IF TOO MUCH ON THIS LINE
; POP  B ;RECALL HEX DIGIT
; JNC  FILHI
; ROOM FOR DIGIT ON THIS LINE

```

```

MOV  A,B
CALL  WHEXB ;WRITE HEX BYTE TO PRINT LINE
FILHI: LHL  FPC
INX  H
SHLD FPC ;READY FOR NEXT BYTE
RET

```

```

; FILADR: ;EMIT DOUBLE PRECISION VALUE FROM H,L

```

```

PUSH H ;SAVE A COPY
MOV  B,L
CALL  FILHB ;LOW BYTE EMITTED
POP  H ;RECOVER A COPY OF H,L
MOV  B,H
JMP  FILHB ;EMIT HIGH BYTE AND RETURN

```

```

; UTILITY FUNCTIONS FOR PRINTING HEX ADDRESSES AND DATA
; CHEX: ;CONVERT TO HEX
ADI  '0'

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950

SER. # \_\_\_\_\_

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA. 93950  
 SER. # \_\_\_\_\_

```

2080 FE3A      CPI      '0'+10
2082 D8       RC
2083 C607     ADI      'A'-'0'-10
2085 C9       RET

;
WHEXN: ;WRITE HEX NIBBLE
CALL CHEX      ;CONVERT TO ASCII FROM HEX
LXI H,NBP
MOV E,M        ;NEXT POSITION TO PRINT
MVI D,0        ;DOUBLE PRECISION
INR M          ;NBP=NBP+1
LXI H,PBUFF
DAD D
MOV M,A        ;STORE IN PRINT BUFFER
RET

;
WHEXB: ;WRITE HEX BYTE TO PRINT BUFFER
PUSH PSW
RAR
RAR
RAR
RAR
ANI 0FH        ;HIGH ORDER NIBBLE NORMALIZE IN A
CALL WHEXN     ;WRITE IT
POP PSW
ANI 0FH
JMP WHEXN     ;WRITE AND RETURN

;
PADD:  LHLD    ASPC
PADDR: ;PRINT ADDRESS FIELD OF PRINT LINE FROM H,L
XCHG
LXI H,NBP     ;INITIALIZE NEXT TO PRINT ADDRESS
MVI H,D       ;PRINT HIGH BYTE
PUSH D        ;SAVE A COPY
CALL WHEXB
POP D
MOV A,E
CALL WHEXB
POP H         ;ADDRESSING NBP
INR M        ;SKIP A SPACE AFTER ADDRESS FIELD
RET

;
ERRR: ;EMIT REGISTER ERROR
PUSH PSW
PUSH B
MVI A,'R'
CALL PERR
POP B

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA 93950  
 SER. # \_\_\_\_\_

```

20C5 F1      POP      PSW
20C6 C9      RET

;
ERRV: ;EMIT VALUE ERROR
PUSH PSW
PUSH H
MVI A,'V'
CALL PERR
POP H
POP PSW
RET

;
ERRD: PUSH PSW
MVI A,'D' ;DATA ERROR
JMP ERR

;
ERRP: PUSH PSW
MVI A,'P'
JMP ERR

;
ERRL: PUSH PSW
MVI A,'L' ;LABEL ERROR
JMP ERR

;
ERRN: PUSH PSW
MVI A,'N' ;NOT IMPLEMENTED

;
ERR: CALL PERR
POP PSW
RET

;
SYLAB: DS 2 ;ADDRESS OF LINE LABEL
EPC: DS 2 ;END PC VALUE
NBP: DS 1 ;NEXT BYTE POSITION TO WRITE FOR MACHINE CODE
END

```

CP/M VERSION \_\_\_\_\_  
 COPYRIGHT © 1976  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA 93950  
 SER. # \_\_\_\_\_