

# || DIGITAL RESEARCH

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

STATUS PROGRAM (STAT)

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

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

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

```

1  STAT,
  DO,
  /* CP/M STATUS COMMAND (STAT)
     COPYRIGHT(C) 1975,1976 DIGITAL RESEARCH
  */
2  1  STATUS, PROCEDURE PUBLIC;
3  2  DECLARE COPYRIGHT(+) BYTE DATA (
4  2  ' COPYRIGHT (C) 1976, DIGITAL RESEARCH');
  DECLARE
  BUFFA LITERALLY '80H', /* DEFAULT BUFFER */
  FCBA LITERALLY '5CH', /* DEFAULT FCB */
  IOBA LITERALLY '3H', /* IOBYTE ADDRESS */
5  2  DECLARE MAM LITERALLY '241' /* MAX BLOCK NUMBER - 1 */
  /* DUMMY OUTER PROCEDURE 'STATUS' WILL START AT 100H */
  /* DETERMINE STATUS OF CURRENTLY SELECTED DISK */
6  2  DECLARE ALLOCA ADDRESS,
  /* ALLOCA IS THE ADDRESS OF THE DISK ALLOCATION VECTOR */
  ALLOC BASED ALLOCA (32) BYTE; /* ALLOCATION VECTOR */
7  2  MON1, PROCEDURE(F,A);
8  3  DECLARE F BYTE,
  A ADDRESS;
  /* PATCH TO JMP 0605 */
9  3  BDOSE: GO TO BDOSE;
10 3  END MON1;
11 2  MON2, PROCEDURE(F,A) BYTE;
12 3  DECLARE F BYTE,
  A ADDRESS;
  /* PATCH TO JMP 2605 */
13 3  BDOSE: GO TO BDOSE;
14 3  RETURN 0;
15 3  END MON2;
16 2  MON3, PROCEDURE(FUNC,INFO) ADDRESS;
17 3  DECLARE FUNC BYTE, INFO ADDRESS;
  /* PATCH TO CALL 0005 */
18 3  BDOSE: GO TO BDOSE;
*** ERROR #2, STATEMENT #18, NEAR 'BDOSE', UNPRINTABLE ASCII CHARACTER IGNORED
  /* PATCH TO MOV H,B MOV L,C */
19 3  RETURN 0;
20 3  END MON3;
21 2  DECLARE
  TRUE LITERALLY '1',
  FALSE LITERALLY '0',
  FOREVER LITERALLY 'WHILE TRUE',
  CR LITERALLY '13',
  LF LITERALLY '10',
  WHAT LITERALLY '63';
22 2  PRINTCHAR, PROCEDURE(CHAR);
23 3  DECLARE CHAR BYTE;
24 3  CALL MON1(2,CHAR);
25 3  END PRINTCHAR;

```

```

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

```

```

26 2  CRLF, PROCEDURE;
27 3  CALL PRINTCHAR(CR);
28 3  CALL PRINTCHAR(LF);
29 3  END CRLF;
30 2  PRINTB, PROCEDURE;
  /* PRINT BLANK CHARACTER */
31 3  CALL PRINTCHAR(' ');
32 3  END PRINTB;
33 2  PRINT, PROCEDURE(A);
  DECLARE A ADDRESS;
  /* PRINT THE STRING STARTING AT ADDRESS A UNTIL THE
  NEXT DOLLAR SIGN IS ENCOUNTERED */
34 3  CALL CRLF;
  CALL MON1(9,A);
  END PRINT;
35 3
36 3
37 3
38 2  PRINTX, PROCEDURE(A);
  DECLARE A ADDRESS;
39 3  CALL MON1(9,A);
40 3  END PRINTX;
41 3
42 2  DECLARE DCNT BYTE;
43 2  SELECT, PROCEDURE(D);
  DECLARE D BYTE;
44 3  CALL MON1(14,D);
45 3  END SELECT;
46 3
47 2  OPEN, PROCEDURE(FCB);
  DECLARE FCB ADDRESS;
48 3  DCNT = MON2(15,FCB);
49 3  END OPEN;
50 3
51 2  SEARCH, PROCEDURE(FCB);
  DECLARE FCB ADDRESS;
52 3  DCNT = MON2(17,FCB);
53 3  END SEARCH;
54 3
55 2  SEARCHN, PROCEDURE;
  DCNT = MON2(18,0);
56 3  END SEARCHN;
57 3
58 2  CSELECT, PROCEDURE BYTE;
  /* RETURN CURRENT DISK NUMBER */
59 3  RETURN MON2(25,0);
60 3  END CSELECT;
61 2  GETALLOCA, PROCEDURE ADDRESS;
  /* GET BASE ADDRESS OF ALLOC VECTOR */
62 3  RETURN MON3(27,0);
63 3  END GETALLOCA;
64 2  DECLARE OLDSP ADDRESS; /* SP ON ENTRY */
  STACK(16) ADDRESS; /* THIS PROGRAM'S STACK */
65 2  DECLARE
  FCB(33) BYTE AT (FCBA), /* DEFAULT FILE CONTROL BLOCK */
  BUFF(128) BYTE AT (BUFFA), /* DEFAULT BUFFER */
  IOVAL BYTE AT (IOBA); /* 10 BYTE */

```

```

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

```

```

66 2  GETALLOC, PROCEDURE(I) BYTE;
    /* RETURN THE ITH BIT OF THE ALLOC VECTOR */
67 3  DECLARE I BYTE;
68 3  RETURN
    ROL(ALLOC(SHR(I,3)), (I AND 111B) + 1);
69 3  END GETALLOC;

78 2  DEVREQ, PROCEDURE BYTE;
    /* PROCESS DEVICE REQUEST, RETURN TRUE IF FOUND */
    /* DEVICE TABLES */
71 3  DECLARE DEVL(*) BYTE DATA
    ('COH,PDR,PUN,LST,DEV,VAL,');
    DEVR(*) BYTE DATA
    (/* CONSOLE */ 'TTY,CRT,BAT,UC1,');
    /* READER */ 'TTY,PTR,URI,UR2,');
    /* PUNCH */ 'TTY,PTP,UP1,UP2,');
    /* LISTING */ 'TTY,CRT,LPT,UL1,');

72 3  DECLARE ACCUM(4) BYTE;
    (IBP, I, J, IOBYTE, ITEMS) BYTE;

73 3  SCAN, PROCEDURE;
    /* FILL ACCUM WITH NEXT INPUT VALUE */
74 4  DECLARE (I,B) BYTE;
75 4  SETACC, PROCEDURE(B);
76 5  DECLARE B BYTE;
77 5  ACCUM(I) = B; I=I+1;
79 5  END SETACC;

    /* DEBLANK INPUT */
80 4  DO WHILE BUFF(IBP) = ' ' IBP=IBP+1;
92 5  END;

    /* INITIALIZE ACCUM LENGTH */
93 4  I = 0;
94 4  DO WHILE I < 4;
95 5  IF (B := BUFF(IBP)) > 1 THEN /* VALID */
96 5  CALL SETACC(B); ELSE /* BLANK FILL */
97 5  CALL SETACC(' ');
98 5  IF B <= 1 OR
    B = '.' OR
    B = ',' OR
    B = '=' OR
    B = '/' OR
    B = '-' OR
    B = '>' OR
    B = '<' THEN BUFF(IBP) = ' ';
    ELSE
99 5  IBP = IBP + 1;
101 5  END;
102 4  IBP = IBP + 1;
103 4  END SCAN;

94 3  MATCH, PROCEDURE(VA,VL) BYTE;
    /* RETURN INDEX+1 TO VECTOR AT VA IF MATCH */
95 4  DECLARE VA ADDRESS;
    V BASED VA (16) BYTE;
    VL BYTE;
96 4  DECLARE (I,J,MATCH,SYNC) BYTE;
97 4  J,SYNC = 0;

```

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

```

98 4  DO SYNC = 1 TO VL;
99 5  MATCH = TRUE;
100 5  DO I = 0 TO 3;
101 6  IF V(I) <> ACCUM(I) THEN MATCH=FALSE;
102 6  J = J + 1;
103 6  END;
104 6  IF MATCH THEN RETURN SYNC;
105 5  END;
106 4  RETURN 0; /* NO MATCH */
107 4  END MATCH;
108 4  PRNAME, PROCEDURE(A);
109 4  DECLARE A ADDRESS;
110 3  X BASED A BYTE;
111 4  /* PRINT DEVICE NAME AT A */
    DO WHILE X <> ',';
    CALL PRINTCHAR(X); A=A+1;
    END;
    CALL PRINTCHAR(',');
    END PRNAME;

118 3  IBP=1; ITEMS = 0;
120 3  DO FOREVER;
121 4  CALL SCAN;
122 4  IF (I:=MATCH(.DEVL,6)) = 0 THEN RETURN ITEMS<>0;
124 4  ITEMS = ITEMS+1; /* FOUND FIRST/NEXT ITEM */
125 4  IF I = 5 THEN /* DEVICE STATUS REQUEST */
126 4  DO;
127 5  IOBYTE = IOVAL; J = 0;
128 5  DO I = 0 TO 3;
129 6  CALL PRNAME(.DEVL(SHL(I,2)));
130 6  CALL PRINTX(, ' IS $');
131 6  CALL PRNAME(.DEVR(SHL(IOBYTE AND 1:B,2)+J));
132 6  J = J + 16; IOBYTE = SHR(IOBYTE,2);
133 6  CALL CRLF;
134 6  END;
135 6  END; ELSE /* NOT DEV. */
136 6  IF I = 6 THEN /* LIST POSSIBLE ASSIGNMENT */
137 4  DO I = 0 TO 3; /* EACH LINE SHOWS ONE DEVICE */
138 5  CALL CRLF;
139 5  CALL PRNAME(.DEVL(SHL(I,2)));
140 5  CALL PRINTX(, ' = $');
141 5  DO J = 0 TO 12 BY 4;
142 6  CALL PRINTCHAR(' ');
143 6  CALL PRNAME(.DEVR(SHL(I,4)+J));
144 6  END;
145 6  END; ELSE
146 6  /* SCAN ITEM I-1 IN DEVICE TABLE */
147 5  DO; /* FIND BASE OF DESTINATION */
148 4  J = SHL(I,=I-1,4);
149 5  CALL SCAN;
150 5  IF ACCUM(0) <> '=' THEN
151 5  DO; CALL PRINT(, 'BAD DELIMITERS');
152 6  RETURN TRUE;
153 6  END;
154 6  CALL SCAN;
155 5  IF (J:=MATCH(.DEVR(J),4)-1) = 255 THEN
156 5  DO; CALL PRINT(, 'INVALID ASSIGNMENTS');
157 6  RETURN TRUE;
158 6  END;
159 6  IOBYTE = 1111$1100B; /* CONSTRUCT MASK */

```

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

```

163 5      DO WHILE (I:=I-1) (<) 255;
164 6      IOBYTE = ROL(IOBYTE,2);
165 6      J = SHL(J,2);
166 6      END;
167 5      IOVAL = (IOVAL AND IOBYTE) OR J;
168 5      END;
169 4      /* END OF CURRENT ITEM, LOOK FOR MORE */
170 4      CALL SCAN;
171 4      IF ACCUM(0) = ' ' THEN RETURN TRUE;
172 4      IF ACCUM(0) (<) ' ' THEN
173 4          DO; CALL PRINT('BAD DELIMITER*');
174 5          RETURN TRUE;
175 5          END;
176 5      END;
177 4      END; /* OF DO FOREVER */
178 3      END DEVREQ;

179 2      COUNT; PROCEDURE;
180 3      /* COUNT RETURNS THE NUMBER OF BLOCKS REMAINING */
181 3      DECLARE C BYTE; /* COUNT */
182 3      I BYTE; /* SEARCH */
183 3      C = 0;
184 3      DO I = 0 TO MADM;
185 4          IF NOT GETALLO(I) THEN C = C + 1;
186 4          END;
187 3      RETURN C;
188 3      END COUNT;

188 2      PVALUE; PROCEDURE(V);
189 3      DECLARE (K,V,D,ZERO) BYTE;
190 3      K = 100;
191 3      ZERO = FALSE;
192 3      DO WHILE K (<) 0;
193 4          D = V / K; V = V MOD K;
194 4          K = K / 10;
195 4          IF ZERO OR K (<) 0 THEN
196 5              DO; ZERO = TRUE; CALL PRINTCHAR('0'+D);
197 5              END;
198 4          END;
199 3      CALL PRINTCHAR('K');
200 3      CALL CRLF;
201 3      END PVALUE;

205 2      PRALLOC; PROCEDURE;
206 3      /* PRINT ALLOCATION FOR CURRENT DISK */
207 3      CALL PRINT('BYTES REMAINING ON $');
208 3      ALLOCA = GETALLOCA;
209 3      CALL PRINTCHAR(CSELECT+'A');
210 3      CALL PRINT(' ');
211 3      CALL PVALUE(COUNT);
212 3      END PRALLOC;

212 2      GETFILE; PROCEDURE;
213 3      /* PROCESS FILE REQUEST */
214 3      DECLARE
215 4          FNAME LITERALLY '11', FEXT LITERALLY '12',
216 4          FRC LITERALLY '15', FLN LITERALLY '15',
217 4          FDM LITERALLY '16', FDL LITERALLY '31',
218 4          FTYPE LITERALLY '9';

```

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

```

214 3      DECLARE
215 4          FCBN BYTE; /* NUMBER OF FCB'S COLLECTED SO FAR */
216 4          FCB(3840) BYTE; /* FCB STORAGE - 16*255*3840 */
217 4          FINX(255) BYTE; /* INDEX VECTOR USED DURING SORT */
218 4          FCB(255) BYTE; /* EXTENT COUNTS */
219 4          FCBK(255) BYTE; /* KILOBYTE COUNT */
220 4          FCB(255) ADDRESS; /* RECORD COUNT */
221 3      DECLARE
222 4          BFCBA ADDRESS; /* INDEX INTO DIRECTORY BUFFER */
223 4          FCBSA ADDRESS; /* INDEX INTO FCBS */
224 4          BFCB BASED BFCBA (32) BYTE; /* TEMPLATE OVER DIRECTORY */
225 4          FCBV BASED FCBSA (16) BYTE; /* TEMPLATE OVER FCBS ENTRY */
226 3      DECLARE
227 4          I BYTE; /* FCB COUNTER DURING COLLECTION AND DISPLAY */
228 4          (K,L,M) BYTE; /* LOOP COUNTERS */
229 4          (B,F) BYTE; /* TEMPS USED DURING SORT */
230 4          MATCHED BYTE; /* USED DURING FCBS SEARCH */
231 3      MULTI16; PROCEDURE;
232 4      /* UTILITY TO COMPUTE FCBS ADDRESS FROM I */
233 4          FCBSA = SHL(DOUBLE(I),4) + FCBS;
234 4          END MULTI16;
235 3      PDECIMAL; PROCEDURE(V,PREC);
236 4      /* PRINT VALUE V WITH PRECISION PREC (10,100,1000)
237 4      WITH LEADING ZERO SUPPRESSION */
238 4      DECLARE
239 5          V ADDRESS; /* VALUE TO PRINT */
240 5          PREC ADDRESS; /* PRECISION */
241 5          ZEROSUP BYTE; /* ZERO SUPPRESSION FLAG */
242 5          D BYTE; /* CURRENT DECIMAL DIGIT */
243 4          ZEROSUP = TRUE;
244 4          DO WHILE PREC (<) 0;
245 5              D = V / PREC; /* GET NEXT DIGIT */
246 5              V = V MOD PREC; /* GET REMAINDER BACK TO V */
247 5              PREC = PREC / 10; /* READY FOR NEXT DIGIT */
248 5              IF PREC (<) 0 AND ZEROSUP AND D = 0 THEN CALL PRINTB; ELSE
249 5                  DO; ZEROSUP = FALSE; CALL PRINTCHAR('0'+D);
250 5                  END;
251 4          END;
252 3      END PDECIMAL;
253 3      /* READ THE DIRECTORY, COLLECT ALL COMMON FILE NAMES */
254 3          FCBN,FCB(0) = 0;
255 3          FCB(FEXT) = 63; /* QUESTION MARK MATCHES ALL */
256 3          CALL SEARCH(FCBA); /* FILL DIRECTORY BUFFER */
257 3          DO WHILE DCNT (<) 255;
258 4              /* ANOTHER ITEM FOUND, COMPARE IT FOR COMMON ENTRY */
259 4              BFCBA = SHL(DCNT AND 118.5)+BUFFA; /* DCNT MOD 4 = 32 */
260 4              MATCHED = FALSE; I = 0;
261 4              DO WHILE NOT MATCHED AND I < FCBN;
262 5                  /* COMPARE CURRENT ENTRY */
263 5                  CALL MULTI16;
264 5                  DO K = 1 TO FNAME;
265 6                      IF BFCB(K) (<) FCBV(K) THEN K = FNAME; ELSE
266 6                          /* COMPLETE MATCH IF AT END */
267 6                          MATCHED = K = FNAME;
268 5                  END;
269 4                  I = I + 1;
270 4              END;

```

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

```

251 4 IF MATCHED THEN I = I - 1; ELSE
253 4 DO; /* COPY TO NEW POSITION IN FCBS */
254 5 FCBN = (I := FCBN) + 1;
255 5 CALL MULTI16;
/* SAVE INDEX TO ELEMENT FOR LATER SORT */
256 5 FINX(I) = I;
257 5 DO K = 0 TO FNAM;
258 6 FCBV(K) = BFCB(K);
259 6 END;
260 5 FCBE(I),FCBK(I),FCBR(I) = 0;
261 5 END;
/* ENTRY IS AT, OR WAS PLACED AT LOCATION I IN FCBS */
262 4 FCBE(I) = FCBE(I) + 1; /* EXTENT INCREMENTED */
263 4 FCBR(I) = FCBR(I) + BFCB(FRC); /* RECORD COUNT */
/* COUNT KILOBYTES */
264 4 DO K = FDM TO FDL;
265 5 IF BFCB(K) <> 0 THEN
266 5 FCBK(I) = FCBK(I) + 1;
267 5 END;
268 4 CALL SEARCHN; /* TO NEXT ENTRY IN DIRECTORY */
269 4 END; /* OF DO WHILE DCNT <> 255 */

/* NOW DISPLAY THE COLLECTED DATA */
270 3 IF FCBN = 0 THEN CALL PRINT(,('FILE NOT FOUND')); ELSE
272 3 DO;
/* SORT THE FILE NAMES IN ASCENDING ORDER */
273 4 IF FCBN > 1 THEN /* REQUIRES AT LEAST TWO TO SORT */
274 4 DO; L = 1;
275 5 DO WHILE L > 0; /* BUBBLE SORT */
276 6 L = 0;
277 6 DO M = 0 TO FCBN - 2;
278 7 I = FINX(M+1); CALL MULTI16; BFCBA = FCBSA; I = FINX(M);
279 7 CALL MULTI16; /* SETS FCBSA, BASING FCBV */
280 7 DO K = 1 TO FNAM; /* COMPARE FOR LESS OR EQUAL */
281 8 IF (B:=BFCB(K)) < (F:=FCBV(K)) THEN /* SWITCH */
282 8 DO; K = FINX(M); FINX(M) = FINX(M + 1);
283 9 FINX(M + 1) = K; L = L + 1; K = FNAM;
284 9 END;
285 8 ELSE IF B > F THEN K = FNAM; /* STOP COMPARE */
286 8 END;
287 7 END;
288 6 END;
289 5 END;
290 4 CALL PRINT(,('RECS BYTS EX D:FILENAME.TYP'));
291 4 L = 0;
292 4 DO WHILE L < FCBN;
293 5 I = FINX(L); /* I IS THE INDEX TO NEXT IN ORDER */
294 5 CALL MULTI16; CALL CRLF;
295 5 CALL PDECIMAL(FCBR(I),1000); /* RRRR */
296 5 CALL PRINTB; /* BLANK */
297 5 CALL PDECIMAL(FCBK(I),100); /* BBB */
298 5 CALL PRINTCHAR('K'); CALL PRINTB;
299 5 CALL PDECIMAL(FCBE(I),10); /* EE */
300 5 CALL PRINTB;
301 5 CALL PRINTCHAR('A'+CSELECT); CALL PRINTCHAR(',');
/* PRINT FILENAME.TYP */
302 5 DO K = 1 TO FNAM;
303 6 IF FCBV(K) <> ' ' THEN
304 6 DO; IF K = FTYF THEN CALL PRINTCHAR(',');
305 6 CALL PRINTCHAR(FCBV(K));

```

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

```

320 7 END;
321 6 END;
322 5 L = L + 1;
323 5 END;
324 4 IF FCBN = 63 OR FCB(FTYP) = 63 THEN CALL PRALLO;
325 4 END;
326 4 END GETFILE;
327 3

/* SAVE STACK POINTER AND RESET */
328 2 OLDSP = STACKPTR;
329 2 STACKPTR = .STACK(LENGTH(STACK));

/* PROCESS REQUEST */
330 2 IF FCB(0) <> 0 THEN CALL SELECT(FCB(0)-1);
331 2 IF FCB(1) = ' ' THEN CALL PRALLO; /* PRINT ALLOCATION */
332 2 ELSE
333 2 IF NOT DEVREQ THEN /* MUST BE FILE NAME */
334 2 CALL GETFILE;
335 2 /* RESTORE OLD STACK BEFORE EXIT */
/* STACKPTR = OLDSP;
END STATUS;
337 2 END;
338 1 END;

```

MODULE INFORMATION:

```

CODE AREA SIZE = 09C9H 2505D
VARIABLE AREA SIZE = 1461H 5217D
MAXIMUM STACK SIZE = 000CH 12D
439 LINES READ
1 PROGRAM ERROR(S)

```

END OF PL/M-80 COMPILATION

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

00000000  
 00000000  
 00000000  
 00000000

1F42H SYM MEMORY  
 01F0H SYM STATUS  
 0100H SYM COPYRIGHT  
 0AE1H SYM ALLOCA  
 0231H SYM MON1  
 0AE3H SYM F  
 0AE4H SYM A  
 0239H SYM BDOSE  
 023DH SYM MON2  
 0AE6H SYM F  
 0AE7H SYM A  
 0245H SYM BDOSE  
 0248H SYM MON3  
 0AE9H SYM FUNC  
 0AEA H SYM INFO  
 0253H SYM BDOSE  
 025AH SYM PRINTCHAR  
 0AEC H SYM CHAR  
 025AH SYM CPLF  
 0275H SYM PRINTB  
 0278H SYM PRINT  
 0AEDH SYM A  
 028EH SYM PRINTX  
 0AEFH SYM A  
 0AF1H SYM DCNT  
 029EH SYM SELECT  
 0AF2H SYM D  
 02AEH SYM OPEN  
 0AF3H SYM FCB  
 02C1H SYM SEARCH  
 0AF5H SYM FCB  
 02D4H SYM SEARCHN  
 02E0H SYM CSELECT  
 02E9H SYM GETALLOCA  
 0AF7H SYM OLDS  
 0AF9H SYM STACK  
 0B5CH SYM FCB  
 0B88H SYM BUFF  
 0B83H SYM IOVAL  
 02F2H SYM GETALLOC  
 0B19H SYM I  
 0318H SYM DEVREQ  
 0127H SYM DEVL  
 013FH SYM DEVR  
 0B1AH SYM ACCUM  
 0B1EH SYM IOP  
 0D1FH SYM I  
 0B28H SYM J  
 0B21H SYM IOBYTE  
 0B22H SYM ITEMS  
 048AH SYM SCAN  
 0B23H SYM I  
 0B24H SYM B  
 056CH SYM SETACC  
 0B25H SYM B  
 0582H SYM MATCH  
 0B26H SYM VA  
 0B28H SYM VL

0B29H SYM I  
 0B2AH SYM J  
 0B2BH SYM MATCH  
 0B2CH SYM SYNC  
 05F0H SYM PRNAME  
 0B2DH SYM A  
 0616H SYM COUNT  
 0B2FH SYM C  
 0B30H SYM I  
 0642H SYM PVALUE  
 0631H SYM V  
 0632H SYM K  
 0B33H SYM D  
 0B34H SYM ZERO  
 06ACH SYM FRALLOCA  
 06CFH SYM GETFILE  
 0B35H SYM FCBN  
 0B36H SYM FCBS  
 1A36H SYM FINX  
 1B35H SYM FCDE  
 1C34H SYM FCBK  
 1D33H SYM FCBR  
 1F31H SYM BFCBA  
 1F33H SYM FCBSA  
 1F35H SYM I  
 1F36H SYM K  
 1F37H SYM L  
 1F38H SYM M  
 1F39H SYM B  
 1F3AH SYM F  
 1F3BH SYM MATCHED  
 0A05H SYM MULTI16  
 0A16H SYM PDECIMAL  
 1F3CH SYM V  
 1F3EH SYM PREC  
 1F40H SYM ZEROSUP  
 1F41H SYM D

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