

|| 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) 1975, DIGITAL RESEARCH');
5  2  DECLARE
6  2  BUFFA LITERALLY '80H', /* DEFAULT BUFFER */
7  2  FCBA LITERALLY '5CH', /* DEFAULT FCB */
8  2  IOBA LITERALLY '3H', /* IOBYTE ADDRESS */
9  2  DECLARE MAM LITERALLY '241' /* MAX BLOCK NUMBER - 1 */
10 2  /* DUMMY OUTER PROCEDURE 'STATUS' WILL START AT 100H */
11 2  /* DETERMINE STATUS OF CURRENTLY SELECTED DISK */
12 2  DECLARE ALLOCA ADDRESS,
13 2  /* ALLOCA IS THE ADDRESS OF THE DISK ALLOCATION VECTOR */
14 2  /* ALLOC BASED ALLOCA (32) BYTE; /* ALLOCATION VECTOR */
15 2  MON1: PROCEDURE(F,A);
16 3  DECLARE F BYTE,
17 3  A ADDRESS;
18 3  /* PATCH TO JMP 0005 */
19 3  BDOSE: GO TO BDOSE;
20 3  END MON1;
21 2  MON2: PROCEDURE(F,A) BYTE;
22 3  DECLARE F BYTE,
23 3  A ADDRESS;
24 3  /* PATCH TO JMP 2005 */
25 3  BDOSE: GO TO BDOSE;
26 3  RETURN 0;
27 3  END MON2;
28 2  MON3: PROCEDURE(FUNC,INFO) ADDRESS;
29 3  DECLARE FUNC BYTE, INFO ADDRESS;
30 3  /* PATCH TO CALL 0005 */
31 3  BDOSE: GO TO BDOSE;
32 3  RETURN 0;
33 3  END MON3;
34 2  *** ERROR #2, STATEMENT #18, NEAR 'BDOSE', UNPRINTABLE ASCII CHARACTER IGNORED
35 2  /* PATCH TO MOV H,B MOV L,C */
36 2  RETURN 0;
37 2  END MON3;
38 2  DECLARE
39 2  TRUE LITERALLY '1',
40 2  FALSE LITERALLY '0',
41 2  FOREVER LITERALLY 'WHILE TRUE',
42 2  CR LITERALLY '13',
43 2  LF LITERALLY '10',
44 2  WHAT LITERALLY '63';
45 2  PRINTCHAR: PROCEDURE(CHAR);
46 3  DECLARE CHAR BYTE;
47 3  CALL MON1(2,CHAR);
48 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;
31 3  /* PRINT BLANK CHARACTER */
32 3  CALL PRINTCHAR(' ');
33 3  END PRINTB;
34 2  PRINT, PROCEDURE(A);
35 3  DECLARE A ADDRESS;
36 3  /* PRINT THE STRING STARTING AT ADDRESS A UNTIL THE
37 3  NEXT DOLLAR SIGN IS ENCOUNTERED */
38 3  CALL CRLF;
39 3  CALL MON1(9,A);
40 3  END PRINT;
41 2  PRINTX, PROCEDURE(A);
42 3  DECLARE A ADDRESS;
43 3  CALL MON1(9,A);
44 3  END PRINTX;
45 2  DECLARE DCHT BYTE;
46 2  SELECT, PROCEDURE(D);
47 3  DECLARE D BYTE;
48 3  CALL MON1(14,D);
49 3  END SELECT;
50 2  OPEN, PROCEDURE(FCB);
51 3  DECLARE FCB ADDRESS;
52 3  DCHT = MON2(15,FCB);
53 3  END OPEN;
54 2  SEARCH, PROCEDURE(FCB);
55 3  DECLARE FCB ADDRESS;
56 3  DCHT = MON2(17,FCB);
57 3  END SEARCH;
58 2  SEARCHN, PROCEDURE;
59 3  DCHT = MON2(18,0);
60 3  END SEARCHN;
61 2  CSELECT, PROCEDURE BYTE;
62 3  /* RETURN CURRENT DISK NUMBER */
63 3  RETURN MON2(25,0);
64 3  END CSELECT;
65 2  GETALLOCA, PROCEDURE ADDRESS;
66 3  /* GET BASE ADDRESS OF ALLOC VECTOR */
67 3  RETURN MON3(27,0);
68 3  END GETALLOCA;
69 2  DECLARE OLDDSP ADDRESS, /* SP ON ENTRY */
70 2  STACK(16) ADDRESS; /* THIS PROGRAM'S STACK */
71 2  DECLARE
72 2  FCB(33) BYTE AT (FCBA), /* DEFAULT FILE CONTROL BLOCK */
73 2  BUFF(128) BYTE AT (BUFFA), /* DEFAULT BUFFER */
74 2  IOVAL BYTE AT (IOBA); /* IO 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
    ('CON,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 */
    DECLARE (I,B) BYTE;
    SETACC, PROCEDURE(B);
    DECLARE B BYTE;
    ACCUM(I) = B; I=I+1;
    END SETACC;

    /* DEBLANK INPUT */
    DO WHILE BUFF(IBP) = ' ', IBP=IBP+1;
    END;
    /* INITIALIZE ACCUM LENGTH */
    I = 0;
    DO WHILE I < 4;
    IF (B := BUFF(IBP)) > 1 THEN /* VALID */
        CALL SETACC(B); ELSE /* BLANK FILL */
        CALL SETACC(' ');
    IF B <= 1 OR
        B = '.' OR
        B = ',' OR
        B = '=' OR
        B = '*' OR
        B = '-' OR
        B = '>' OR
        B = '<' THEN BUFF(IBP) = ' ';
    ELSE
        IBP = IBP + 1;
    END;
    IBP = IBP + 1;
    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(J) <> ACCUM(I) THEN MATCH=FALSE;
102 6  J = J + 1;
103 6  END;
104 6  IF MATCH THEN RETURN SYNC;
105 5  END;
106 5  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;

112 4  IBP=1; ITEMS = 0;
120 3  DO FOREVER;
121 4  CALL SCAN;
122 4  IF (I:=MATCH(DEVL,6)) = 0 THEN RETURN ITEMS<>0;
123 4  ITEMS = ITEMS+1; /* FOUND FIRST/NEXT ITEM */
124 4  IF I = 5 THEN /* DEVICE STATUS REQUEST */
125 4  DO;
126 5  IOBYTE = IOVAL; J = 0;
127 5  DO I = 0 TO 3;
128 6  CALL PRNAME(DEVL(SHL(I,2)));
129 6  CALL PRINTX(' IS $');
130 6  CALL PRNAME(DEVR(SHL(IOBYTE AND 11B,2)+J));
131 6  J = J + 16; IOBYTE = SHR(IOBYTE,2);
132 6  CALL CRLF;
133 6  END;
134 5  END; ELSE /* NOT DEV. */
135 4  IF I = 6 THEN /* LIST POSSIBLE ASSIGNMENT */
136 4  DO I = 0 TO 3; /* EACH LINE SHOWS ONE DEVICE */
137 5  CALL CRLF;
138 5  CALL PRNAME(DEVL(SHL(I,2)));
139 5  CALL PRINTX(' = $');
140 5  DO J = 0 TO 12 BY 4;
141 6  CALL PRINTCHAR(' ');
142 6  CALL PRNAME(DEVR(SHL(I,4)+J));
143 6  END;
144 5  END; ELSE
145 4  /* SCAN ITEM I-1 IN DEVICE TABLE */
146 4  DO; /* FIND BASE OF DESTINATION */
147 5  J = SHL(I:=I-1,4);
148 5  CALL SCAN;
149 5  IF ACCUM(0) <> '=' THEN
150 6  DO; CALL PRINT('BAD DELIMITERS');
151 6  RETURN TRUE;
152 6  END;
153 5  CALL SCAN;
154 5  IF (J:=MATCH(DEVR(J),4)-1) = 255 THEN
155 6  DO; CALL PRINT('INVALID ASSIGNMENTS');
156 6  RETURN TRUE;
157 6  END;
158 5  IOBYTE = 1111$1100B; /* CONSTRUCT MASK */
159 5
160 5
161 5
162 5

```

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 SC4H;
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 BYTE;
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 M4LM;
185 4      IF NOT GETALLOCC(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      PRALLOCC: PROCEDURE;
206 3      /* PRINT ALLOCATION FOR CURRENT DISK */
207 3      CALL PRINT('BYTES REMAINING ON $');
208 3      ALLOCA = GETALLOCA;
209 3      CALL PRINTCHAR(CHAR(1));
210 3      CALL PRINT(' ');
211 3      CALL PVALUE(COUNT);
212 3      END PRALLOCC;

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 3      FCBN BYTE; /* NUMBER OF FCB'S COLLECTED SO FAR */
216 3      FCB(3840) BYTE; /* FCB STORAGE - 16*255*3840 */
217 3      FINX(255) BYTE; /* INDEX VECTOR USED DURING SORT */
218 3      FCB(255) BYTE; /* EXTENT COUNTS */
219 3      FCBK(255) BYTE; /* KILOBYTE COUNT */
220 3      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 4      DECLARE
227 5      I BYTE; /* FCB COUNTER DURING COLLECTION AND DISPLAY */
228 5      (K,L,M) BYTE; /* LOOP COUNTERS */
229 5      (B,F) BYTE; /* TEMPS USED DURING SORT */
230 5      MATCHED BYTE; /* USED DURING FCBS SEARCH */
231 5      MULTI16: PROCEDURE;
232 6      /* UTILITY TO COMPUTE FCBS ADDRESS FROM I */
233 6      FCBSA = SHL(DOUBLE(I),4) + .FCBS;
234 6      END MULTI16;
235 5      PDECIMAL: PROCEDURE(V,PREC);
236 6      /* PRINT VALUE V WITH PRECISION PREC (10,100,1000) */
237 6      WITH LEADING ZERO SUPPRESSION */
238 6      DECLARE
239 7      V ADDRESS; /* VALUE TO PRINT */
240 7      PREC ADDRESS; /* PRECISION */
241 7      ZEROSUP BYTE; /* ZERO SUPPRESSION FLAG */
242 7      D BYTE; /* CURRENT DECIMAL DIGIT */
243 7      ZEROSUP = TRUE;
244 7      DO WHILE PREC <> 0;
245 8      D = V / PREC; /* GET NEXT DIGIT */
246 8      V = V MOD PREC; /* GET REMAINDER BACK TO V */
247 8      PREC = PREC / 10; /* READY FOR NEXT DIGIT */
248 8      IF PREC <> 0 AND ZEROSUP AND D = 0 THEN CALL PRINTB; ELSE
249 9      DO; ZEROSUP = FALSE; CALL PRINTCHAR('0'+D);
250 9      END;
251 8      END;
252 7      END;
253 6      END PDECIMAL;

254 5      /* READ THE DIRECTORY, COLLECT ALL COMMON FILE NAMES */
255 5      FCBN,FCB(0) = 0;
256 5      FCB(FEXT) = 63; /* QUESTION MARK MATCHES ALL */
257 5      CALL SEARCH(FCBA); /* FILL DIRECTORY BUFFER */
258 5      DO WHILE DCNT <> 255;
259 6      /* ANOTHER ITEM FOUND, COMPARE IT FOR COMMON ENTRY */
260 6      BFCBA = SHL(DCNT AND 118.5)+BUFFA; /* DCNT MOD 4 = 32 */
261 6      MATCHED = FALSE; I = 0;
262 6      DO WHILE NOT MATCHED AND I < FCBN;
263 7      /* COMPARE CURRENT ENTRY */
264 7      CALL MULTI16;
265 7      DO K = 1 TO FNAME;
266 8      IF BFCB(K) <> FCBV(K) THEN K = FNAME; ELSE
267 9      /* COMPLETE MATCH IF AT END */
268 9      MATCHED = K = FNAME;
269 8      END;
270 7      I = I + 1;
271 6      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 FNAME;
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 */
          IF FCBN > 1 THEN /* REQUIRES AT LEAST TWO TO SORT */
273 4      DO; L = 1;
274 4      DO WHILE L > 0; /* BUBBLE SORT */
275 5      L = 0;
276 5      DO M = 0 TO FCBN - 2;
277 6      I = FINX(M+1); CALL MULTI16; BFCBA = FCBSA; I = FINX(M);
278 6      CALL MULTI16; /* SETS FCBSA, BASING FCBV */
279 7      DO K = 1 TO FNAME; /* COMPARE FOR LESS OR EQUAL */
280 8      IF (B:=BFCB(K)) < (F:=FCBV(K)) THEN /* SWITCH */
281 8      DO; K = FINX(M); FINX(M) = FINX(M + 1);
282 8      FINX(M + 1) = K; L = L + 1; K = FNAME;
283 8      END;
284 8      ELSE IF B > F THEN K = FNAME; /* STOP COMPARE */
285 8      END;
286 8      END;
287 8      END;
288 8      END;
289 8      END;
290 8      END;
291 8      END;
292 8      END;
293 8      END;
294 8      END;
295 8      END;
296 8      END;
297 8      END;
298 8      END;
299 8      END;
300 8      END;
301 8      END;
302 8      END;
303 8      END;
304 8      END;
305 8      END;
306 8      END;
307 8      END;
308 8      END;
309 8      END;
310 8      END;
311 8      END;
312 8      END;
313 8      END;
314 8      END;
315 8      END;
316 8      END;
317 8      END;
318 8      END;
319 8      END;

```

SER. #

PACIFIC GROVE, CA 93950

COPYRIGHT © 1976
DIGITAL RESEARCH
P. O. BOX 579

CP/M VERSION

```

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

          /* SAVE STACK POINTER AND RESET */
328 2      OLDSB = 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 = OLDSB;
          END STATUS;
336 2      END;
337 1      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. #

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
 026AH SYM CRLF
 0275H SYM PRINTB
 027BH SYM PRINT
 0AEDH SYM A
 026EH 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
 005CH SYM FCB
 0060H SYM BUFF
 0063H SYM IOVAL
 02F2H SYM GETALLOC
 0019H SYM I
 0310H SYM DEVREQ
 0127H SYM DEVL
 013FH SYM DEVR
 0B1AH SYM ACCUM
 001EH SYM IOP
 001FH SYM I
 0B20H SYM J
 0B21H SYM IOBYTE
 0022H SYM ITEMS
 04EAH SYM SCAN
 0B23H SYM I
 0B24H SYM B
 050CH SYM SETACC
 0B25H SYM B
 0502H SYM MATCH
 0B26H SYM VA
 0B28H SYM VL

0029H SYM I
 002AH SYM J
 002BH SYM MATCH
 002CH SYM SYNC
 05F0H SYM PRNAME
 002DH SYM A
 0016H SYM COUNT
 002FH SYM C
 0030H SYM I
 0042H SYM PVALUE
 0031H SYM V
 0032H SYM K
 0033H SYM D
 0034H SYM ZERO
 00ACH SYM FRALLOP
 00CFH SYM GETFILE
 0035H SYM FCBN
 0036H SYM FCB
 1A36H SYM FINX
 1035H SYM FCDE
 1C34H SYM FCBK
 1D33H SYM FCB
 1F31H SYM BFCBA
 1F33H SYM FCB
 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

CP/M VERSION _____

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. # _____