

```

1  %BEGIN
2  %OWNINTEGER MIN = 1;
3  %OWNINTEGER MOUT = 1;
4  %OWNINTEGER SIN = 2;
5  %OWNINTEGER MON = 1;
6  %OWNINTEGER PRINTED;
7  %OWNINTEGER CLOSE;
8  %OWNINTEGER STOP = -2000;
9  %OWNINTEGER INV = -2001;
10 %INTEGER I, J, K, PP1, SYM
11 %INTEGER CI;
12 %INTEGER TI;
13 %INTEGER CODE;
14 %INTEGER TEXT;
15 %INTEGER NUM;
16 %INTEGER LAST;
17 %INTEGER TOP;
18 %INTEGER PE;
19 %INTEGER PP;
20 %INTEGER FP;
21 %INTEGER END;
22 %INTEGER LIM;
23 %OWNINTEGER MS;
24 %OWNINTEGER ML;
25 %INTEGER IP;
26 %INTEGER TB;
27 %SWITCH S('A': 'Z')
28 %INTEGERARRAY C(1:91);
29 %INTEGERARRAY T(1:80);
30 %INTEGER AMAX
31 %FAULT 9 ->0; SELECT INPUT(SIN); K = NEXT SYMBOL
32 0: AMAX = FREESTORE-520
33 %INTEGERARRAY A(0:AMAX)
34
35 %ROUTINE LOAD PP(%INTEGER K)
36 %INTEGER P
37 %LAC PP; %OPR 2064;!RCR; %DAC P
38 %OPR 768;!SZL; %JMP L1
39 %LAC K; %OPR 1032;!RTL; %OPR 1032
40 %OPR 1032; %OPR 1032; %DAC K
41 %LAC (-256; %OPR 513;!SKP!CMA
42 L1: %LAC (-256; %AND* P; %XOR K; %DAC* P
43 %END
44
45 %ROUTINE LOAD FP(%INTEGER K)
46 %INTEGER P
47 %LAC FP; %OPR 2064;!RCR; %DAC P
48 %OPR 768;!SZL; %JMP L1
49 %LAC K; %OPR 1032;!RTL; %OPR 1032
50 %OPR 1032; %OPR 1032; %DAC K
51 %LAC (-256; %OPR 513;!SKP!CMA
52 L1: %LAC (-256; %AND* P; %XOR K; %DAC* P
53 %END
54
55 %INTEGERFN BYTE(%INTEGER P)
56 %OPR 2064;!RCR; %DAC P
57 %LAC* P; %OPR 768;!SZL; %JMP L1
58 %OPR 1040;!RTR; %OPR 1040
59 %OPR 1040; %OPR 1040
60 L1: %AND (255; %JMP* BYTE

```

```

!MAIN FILE INPUT STREAM
!MAIN FILE OUTPUT STREAM
!SECONDARY INPUT STREAM
!MONITOR SWITCH
!PRINT INDICATOR
!CLOSE INDICATOR
!LOOP STOP
!INVERSION INDICATOR
!COMMAND INDEX
!TEXT INDEX AD
!COMMAND CODE
!TEXT STRING AD
!REPETITION NUMBER
!LAST COMMAND CODE
!TOP OF BUFF
!PREVIOUS LINE END
!PREVIOUS POINTER
!FILE POINTER
!END OF BUFF INF
!BUFF LIMIT
!MATCH START
!MATCH LIMIT
!INPUT FILE POINTER
!TEXT STRING BASE
!COMMAND SEQUENCE
!TEXT STRINGS

```

```

61      %STOP
62 %END
63
64 %INTEGERMAP INTEGER(%INTEGER A)
65      %DAC 1; %JMP* INTEGER
66      %STOP
67 %END
68
69 %ROUTINE MAKE SPACE
70 %INTEGER K,P1,P2
71      %RETURN %IF FP-PP-240 > 0
72      SELECT OUTPUT(MOUT)
73      P1 = TOP; P2 = (P1+PE)>>1
74      P2 = PE %IF CLOSE # 0; %MONITOR 20 %IF P2 = TOP
75 1: K = BYTE(P1); PRINT SYMBOL(K); P1 = P1+1
76     ->1 %UNLESS K = NL %AND P1-P2 >= 0
77     SELECT OUTPUT(0)
78     PE = TOP+PE-P1; P2 = PP; PP = TOP
79 2: %RETURN %IF P1 = P2
80     LOAD PP(BYTE(P1)); PP = PP+1; P1 = P1+1
81     ->2
82 %END
83
84 %ROUTINE READ COMMAND
85 %OWNINTEGER CLIM
86 %INTEGER I,TYPE,CTYPE,QUOTE,CHAIN
87 %SWITCH T(2:10)
88
89 %INTEGERFN NEXT ITEM TYPE
90 %SWITCH S(32:95)
91 S(' '): READ SYMBOL(SYM); SYM = SYM-32 %IF SYM >= 96
92     ->S(SYM) %IF SYM >= 32
93 S(';'): %RESULT = 1
94 S('('): %RESULT = 2
95 S(','): %RESULT = 3
96 S(')'): %RESULT = 4
97 S('D'): %RESULT = 5
98 S('I'): %RESULT = 6
99 S('S'): %RESULT = 7
100 S('F'):S('L'):S('T'):S('U'):S('V'): %RESULT = 8
101 S('A'):S('B'):S('E'):S('G'):S('M'):S('P'):S('R'): %RESULT = 9
102 S('C'):S('H'):S('J'):S('K'):S('O'):S('Q'):
103 S('W'):S('X'):S('Y'):S('Z'): %RESULT = 10
104 S('0'):S('1'):S('2'):S('3'):S('4'):S('5'):S('6'):S('7'):S('8'):S('9'):
105     NUM = SYM-'0'
106 1: I = NEXT SYMBOL
107     %RESULT = 0 %UNLESS '0' <= I <= '9'
108     NUM = (NUM<<2+NUM)<<1+I-'0'; SKIP SYMBOL
109     ->1
110 S('*'): NUM = 0; %RESULT = 0
111 S('?'): NUM = STOP+1; %RESULT = 0
112 S(92): NUM = INV+1; %RESULT = 0
113 S('!'):
114 2: READ SYMBOL(SYM); ->2 %UNLESS SYM = NL
115     %RESULT = 1
116 S('N'): TEXT = -1
117 S(34):S('#'):S('$'):S('%'):S('&'):S('||'):
118 S('+'):S('-'):S('.')S('/'):S(':'):S('<'):
119 S('='):S('>'):S('@'):S(91):S(93):S(94):S('_'):
120     %RESULT = -1

```

```

121 %END
122
123 %ROUTINE UNCHAIN
124 1: %RETURN %IF CHAIN >= 0
125     TEXT = -CHAIN; CHAIN = C(TEXT); C(TEXT) = CI
126     ->1
127 %END
128
129 %ROUTINE STACK(%INTEGER V)
130     CI = CI+1; C(CI) = V
131 %END
132
133 1: PROMPT('>'); TYPE = NEXT ITEM TYPE
134     ->1 %IF TYPE = 1
135     CI = 0; TI = TB; CHAIN = 0
136     %IF TYPE = 0 %AND CLIM # 0 %THEN %START
137         C(CLIM) = NUM; ->ER1 %UNLESS NEXT ITEM TYPE = 1
138     %RETURN
139     %FINISH
140     %IF SYM = '%' %THEN %START
141         READ SYMBOL(SYM)
142         ->ER1 %UNLESS SYM='C' %OR SYM='M' %OR SYM='S'
143         CODE = SYM; ->ER1 %UNLESS NEXT ITEM TYPE = 1
144         %IF CODE # 'C' %THEN %START
145             MON = CODE-'S'; ->1
146         %FINISH
147         CLOSE = 1; %RETURN
148     %FINISH
149 2: ->ER1 %UNLESS TYPE > 0
150     CODE = SYM; TEXT = 0; NUM = 1
151     CTYPE = TYPE; TYPE = NEXT ITEM TYPE
152     ->T(CTYPE)
153 T(2):                                     !LEFT BRACKET
154     TEXT = CHAIN; CHAIN = CI+2; CODE = 'Y'
155     ->6
156 T(3):                                     !COMMA
157     TEXT = CHAIN; CHAIN = -(CI+2)
158     NUM = INV; CODE = 'Z'
159     ->6
160 T(4):                                     !RIGHT BRACKET
161     UNCHAIN; ->ER2 %IF CHAIN = 0
162     TEXT = CHAIN; CHAIN = C(TEXT); C(TEXT) = CI+3
163     TEXT = TEXT+1; C(TEXT) = NUM; CODE = 'Z'
164     ->5
165 T(5):                                     !DELETE
166     ->ER1 %IF NUM = 0
167 T(6):                                     !OPTIONAL TEXT
168     ->5 %IF TYPE >= 0
169 T(7):                                     !TEXT OR N
170     ->4 %IF TEXT < 0
171 T(8):                                     !TEXT
172     ->ER3 %UNLESS TYPE < 0 %AND TEXT = 0
173     QUOTE = SYM; TEXT = TI
174 3: READ SYMBOL(SYM)
175     %IF SYM # QUOTE %THEN %START
176         ->ER3 %IF SYM = NL %OR TI-TB > 78
177         INTEGER(TI) = SYM; TI = TI+1
178         ->3
179     %FINISH
180     ->ER3 %IF TI = TEXT %AND CODE # 'S'

```

```

181     INTEGER(TI) = 0;  TI = TI+1
182 4:  TYPE = NEXT ITEM TYPE
183 T(9):
184 5:  TYPE = NEXT ITEM TYPE %IF TYPE = 0
185 6:  STACK(CODE);  STACK(TEXT);  STACK(NUM)
186     ->2 %UNLESS TYPE = 1
187     UNCHAIN;  ->ER2 %UNLESS CHAIN = 0
188     STACK('Z');  STACK(0);  STACK(1)
189     CLIM = CI;  STACK(0)
190     %RETURN
191
192 ER1:CODE = SYM
193     ->ER5
194 ER2:%PRINTTEXT ' BRACKETS?'
195     ->9
196 ER3:%PRINTTEXT ' TEXT FOR'
197 T(10):
198 ER5:SPACE;  PRINT SYMBOL(CODE);  PRINT SYMBOL('?')
199 9:  NEWLINE;  CLIM = 0 %IF CI # 0
200 10: ->1 %IF SYM = NL;  READ SYMBOL(SYM)
201     ->10
202 %END
203
204 %ROUTINE PRINT LINE
205 %INTEGER P
206     PRINTED = 1
207     ->3 %IF FP = END
208     P = PE
209     ->2 %UNLESS P = PP
210 1:  P = FP
211 2:  PRINT SYMBOL(BYTE(P));  %RETURN %IF BYTE(P) = NL
212     P = P+1
213     ->2 %UNLESS P = PP
214     PRINT SYMBOL(94) %IF NUM = 0
215     ->1
216 3:  %PRINTTEXT '**END**';  NEWLINE
217 %END
218
219 %ROUTINE READ LINE
220 %INTEGER K
221     PRINTED = 0
222     %RETURN %UNLESS FP = END # IP
223     SELECT INPUT(MIN)
224     FP = IP;  MS = 0
225 1:  K = NL
226     READ SYMBOL(K) %UNLESS FP = LIM %AND NEXT SYMBOL # NL
227     LOAD FP(K);  FP = FP+1
228     ->1 %UNLESS K = NL
229     END = FP;  FP = IP
230     SELECT INPUT(0)
231 %END
232
233 %ROUTINE INSERT LINE
234 %INTEGER K
235 1:  ->2 %IF PP = PE
236     FP = FP-1;  PP = PP-1;  LOAD FP(BYTE(PP))
237     ->1
238 2:  READ SYMBOL(K);  LOAD PP(K);  PP = PP+1
239     ->2 %UNLESS K = NL
240     PE = PP

```

```

241 %END
242
243 %ROUTINE MOVE BACK
244 %INTEGER K
245 1: K = BYTE(PP-1)
246     ->2 %IF K = NL %AND PP # PE
247     FP = FP-1; PP = PP-1; LOAD FP(K)
248     ->1
249 2: PE = PP; MS = 0; PRINTED = 0
250 %END
251
252 %ROUTINE MOVE
253 %INTEGER K
254     MAKE SPACE
255 1: K = BYTE(FP); LOAD PP(K)
256     PP = PP+1; FP = FP+1
257     ->1 %UNLESS K = NL
258     PE = PP
259     READ LINE
260 %END
261
262 %INTEGERFN MATCHED(%INTEGER IND)
263 %INTEGER I,J,K,L,FP1
264     PP1 = PP; FP1 = FP
265     J = INTEGER(TEXT)
266     ->2 %UNLESS FP = MS %AND IND >= 0
267     K = BYTE(FP)
268 1: LOAD PP(K); PP = PP+1; FP = FP+1
269 2: K = BYTE(FP)
270     ->5 %IF K = NL
271     ->1 %UNLESS K = J
272     L = 1
273 4: I = INTEGER(TEXT+L); ->6 %IF I = 0
274     ->1 %IF BYTE(FP+L) # I
275     L = L+1
276     ->4
277 5: ->9 %IF IND <= 0; ->10 %IF FP = END
278     LOAD PP(K); PP = PP+1; FP = FP+1
279     PE = PP; PRINTED = 0
280     MAKE SPACE; READ LINE
281     ->2
282 6: MS = FP; ML = FP+L
283     %RESULT = 1
284 9: PP = PP1; FP = FP1
285 10: %RESULT = 0
286 %END
287
288 *INITIALISE
289     %FAULT 9 ->EOF
290     TOP = ADDR(A(0))<<1+1; LIM = ADDR(A(AMAX))<<1+1
291     IP = LIM-120; FP = IP; END = FP+1
292     PP = TOP-1; TB = ADDR(T(1))
293     LOAD FP(NL); MOVE
294 %OWNINTEGER KT=35;%LAC* KT;%OPR 640;%LAC (1);%DAC* KT
295 !PROCESS COMMAND SEQUENCE
296 1: SELECT INPUT(0); SELECT OUTPUT(0)
297 2: READ COMMAND; ->FIN %IF CLOSE # 0
298     CI = 0; LAST = 0
299 3: CODE = C(CI+1); ->9 %IF CODE = 0
300     TEXT = C(CI+2); NUM = C(CI+3)

```

```

301      CI = CI+3
302  4:  NUM = NUM-1
303      ->S(CODE)
304  OK:  LAST = CODE
305      ->5 %IF NUM = INV
306      ->4 %UNLESS NUM = 0 %OR NUM = STOP
307      ->3
308  NO:  ->3 %IF NUM < 0
309  5:  I = C(CI+1); CI = CI+3; CI = C(CI-1) %IF I = 'Y'
310      ->5 %UNLESS I = 0 %OR (I='Z' %AND C(CI)<=0)
311      ->3 %IF I # 0
312      %PRINTTEXT 'FAILURE: '; PRINT SYMBOL(CODE)
313      %IF TEXT > 0 %THEN %START
314          PRINT SYMBOL('')
315  6:  ->7 %IF INTEGER(TEXT) = 0
316          PRINT SYMBOL(INTEGER(TEXT)); TEXT = TEXT+1
317          ->6
318  7:  PRINT SYMBOL('')
319      %FINISH
320      PRINT SYMBOL('\') %IF NUM = INV; NEWLINE
321  8:  ->9 %IF SYM = NL; READ SYMBOL(SYM); ->8
322  9:  NUM = 0
323      PRINT LINE %IF MON # 0 %AND PRINTED = 0 %AND SYM = NL
324      ->2
325  EOF: I = INPUT; SELECT INPUT(O)
326      ->NO %IF I = SIN
327      %IF I = MIN %THEN %START
328          FP = LIM; FND = FP; LOAD FP(NL)
329          %JUMP* READ LINE
330      %FINISH
331      CLOSE INPUT
332      ->2
333
334  !EXECUTE COMMANDS
335  S('Y'):                                     !OPEN BRACKET
336      C(TEXT) = NUM+1
337      ->3
338  S('Z'):                                     !CLOSE BRACKET
339      ->5 %IF NUM = INV
340      %IF NUM # 0 %AND NUM # STOP %THEN %START
341          C(CI) = NUM; CI = TEXT
342      %FINISH
343      ->3
344  S('A'):                                     !ADVANCE
345      K = BYTE(FP)
346      ->NO %IF K = NL
347      LOAD PP(K); PP = PP+1; FP = FP+1
348      ->OK
349  S('R'):                                     !RETRACT
350      ->NO %IF PP = PE
351      FP = FP-1; PP = PP-1; LOAD FP(BYTE(PP))
352      MS = 0
353      ->OK
354  S('E'):                                     !ERASE
355      ->NO %IF BYTE(FP) = NL
356      FP = FP+1
357      ->OK
358  S('V'):                                     !VERIFY
359      I = FP; J = TEXT
360  V1: ->NO %IF BYTE(I) # INTEGER(J)

```

```

361      I = I+1;  J = J+1
362      ->V1 %UNLESS INTEGER(J) = 0
363      MS = FP;  ML = I
364      ->OK
365  S('F'):                                     !FIND
366      ->NO %IF MATCHED(1) = 0
367      ->OK
368  S('L'):                                     !LOCATE
369      ->NO %IF MATCHED(0) = 0
370      ->OK
371  S('U'):                                     !UNCOVER
372      ->NO %IF MATCHED(0) = 0;  PP = PP1
373      ->OK
374  S('T'):                                     !TRAVERSE
375      ->NO %IF MATCHED(-1) = 0;  FP = ML
376      ->I1
377  S('D'):                                     !DELETE
378      ->D3 %IF TEXT <= 0
379      ->NO %IF MATCHED(-1) = 0;  FP = ML;
380      ->OK
381  D3: ->NO %IF FP = END
382      ->D5 %IF TEXT = 0
383  D4: K = BYTE(FP)
384      LOAD PP(K);  PP = PP+1;  FP = FP+1
385      ->D4 %UNLESS K = NL
386      READ LINE;  PP = PP-1
387      ->OK %UNLESS PP-PE > 80 %OR (FP=END %AND PP#PE)
388      PP = PP+1;  PE = PP
389      ->NO
390  D5: PP = PE;                                     !DELETE LINE
391  D6: FP = FP+1;  ->D6 %UNLESS BYTE(FP-1) = NL
392      READ LINE
393      ->OK
394  S('S'):                                     !SUBSTITUTE
395      ->NO %IF FP # MS;  FP = ML
396  S('I'):                                     !INSERT
397      MAKE SPACE
398      ->I3 %IF TEXT <= 0
399      ->NO %IF PP-PE > 80 %OR FP = END
400  I1: I = TEXT;                                     !INSERT TEXT
401  I2: ->OK %IF INTEGER(I) = 0
402      LOAD PP(INTEGER(I));  PP = PP+1;  I = I+1
403      ->I2
404  I3: ->I5 %IF TEXT = 0
405      LOAD PP(NL);  PP = PP+1;  PE = PP;
406      ->OK
407  I5: PROMPT(':');                                     !INSERT LINE
408      ->I9 %IF NEXY SYMBOL = ':'
409      INSERT LINE;  SYM = NL
410      ->OK
411  I9: READ SYMBOL(SYM);  ->I9 %UNLESS SYM = NL
412      ->NO
413  S('M'):                                     !MOVE
414      ->NO %IF FP = END
415      MOVE
416      ->OK
417  S('B'):                                     !BACKOFF
418      MAKE SPACE
419      ->NO %IF PE = TOP
420      MOVE BACK

```

```

421     ->OK
422 S('P'):                                !PRINT
423     %IF LAST = 'P' %THEN %START
424     ->NO %IF FP = END; MOVE
425     %FINISH
426     PRINT LINE
427     ->OK
428 S('G'):                                !GET
429     MAKE SPACE
430     SELECT INPUT(SIN); INSERT LINE
431     SELECT INPUT(O)
432     ->OK
433 FIN:                                    !CLOSE
434 90: ->91 %IF FP = END; MOVE; ->90
435 91: SELECT OUTPUT(MOUT)
436 92: %IF TOP # PP %THEN %START
437     PRINT SYMBOL(BYTE(TOP)); TOP = TOP+1
438     ->92
439     %FINISH
440 %ENDOFPROGRAM

```