```
 1
 2  !                    ** EDIT4C **
 3  !
 4  !
 5  !
 6  !
 7      %FAULT 30 -> 0     ;!  RE-START COMMAND DECODER
 8      %FAULT 31 -> STOP      ;!  USED BY 'ABORT' TO KILL BATCH EDITOR RUN
 9      %FAULT 9 -> FAULT9  ;!  -> FAULT9 WHEN 'INPUT ENDED' FAULT OCCURS
10      OTC(0)
11  !    WRITE(AMAX,1)
12      NEWLINE ; %PRINTTEXT 'IMP EDIT    29/10/70' ; NEWLINE
13      INIT ; FILL MAIN
14  PW('X'):                          ;!  IN CASE FILE IS SHORT( <= ONE BUFFER )
15  !
16  !   'BATCH' AND 'LISTING' ARE '%OWN' VARIABLES INITIALISED TO 'OFF'
17  !
18  !
19  !
20  !   SEQUENCING MECHANISM FOLLOWS
21  !
22  0: LVL = -1 ; PFLAG = 1
23  1: READ COMMAND STRING %IF LVL < 0
24  2: WK1 = CB(CBP) ; CBP = CBP - 1
25  !
26  ! NOW SEPARATE EDITOR FN. AND PARAMETER POINTER INTO 'I' AND 'WK1' RESP.
27      I = WK1 & 127 ; WK1 = WK1 >> 7
28  !
29      %IF I = '(' %THENSTART
30          LVL = LVL + 1
31          REPT(LVL) = CB(WK1)
32          EXIT(LVL) = CB(WK1+1)
33          RESET(LVL) = CBP ; -> 2
34      %FINISH
35  !
36      %IF I = ')' %THENSTART
37          REPT(LVL) = REPT(LVL) - 1
38          %IF REPT(LVL) = 0 %THEN LVL = LVL - 1 %ELSE CBP = RESET(LVL)
39          -> 1
40      %FINISH
41  !
42      -> SW(I)          ;! GO TO APPROPRIATE EDITING FUNCTION
43  !
44  3: WK1 = 1
45  4: LVL = LVL - WK1
46      %IF LVL < 0 %THENSTART
47          ABORT(2) %IF BATCH = ON %AND CBP # EXIT(0) ; -> 1
48      %FINISH
49      CBP = EXIT(LVL+1) ; -> 2
50  !
51  !
52  !
53  !
54  EOF: ERROR(2)
55      %IF STATUS(IN) = 2 %THENSTART
56          STATUS(IN) = 1 ; IN = 1 ; -> 0
57      %FINISH
58  !
59      STATUS(IN) = STATUS(IN) + 1 ; IN = 1 ; -> 3
60  !
```

```
61    !
62    EOL: ERROR(3) ; -> 3
63    !
64    !
65    !
66    !
67    SEEK FAIL: %IF F = -1 %THEN -> 3 %ELSESTART
68          ERROR(4) ; -> 0     ;! GIVES SPURIOUS '*LOOP TRAP' IF -1 # F # -2
69       %FINISH
70    !
71    !
72    !
73    !
74    FAULT9:  OTC(0)
75       -> 0 %IF F = 2        ;! IF READING COMMAND
76       %IF STATUS(IN) # 0 %THENSTART  ;! READ REQUEST HAS BEEN ISSUED ON ..
77          NEWLINE     ;! ... CHANNEL WHICH IS ALREADY 'INPUT ENDED'
78          %PRINTTEXT 'STAT(' ; PRINTSYMBOL('0'+IN)
79          %PRINTTEXT ') # 0' ; %STOP  ;! ** THIS SHOULD NEVER HAPPEN **
80       %FINISH
81    !
82       STATUS(IN) = 1
83       STORE(NLDF) %IF P # 0 %AND LIST(P)&DF # NLDF  ;! CLOSE WITH NEWLINE
84    !
85    !  DEAL WITH ABORTIVE ATTEMPTS TO READ FROM INPUT
86       TEMP = 0 %IF TEMP = ASL
87    !
88       %IF IN = 1 %THENSTART    ;! OTHERWISE LEAVE IT TO ROUTINE 'G'
89          CHOP IT OFF ; JOIN ON(TEMP)
90       %FINISH
91    !
92       -> PW(I)
93    !
94    !
95    !
96    !
97    !
98    !
99    !
100   SW('A'):PW('A'):  SEEK ; -> SEEK FAIL %IF F < 0
101      WK2 = CB(WK1)
102      CL = CL + ( WK2 >> 8 ) %IF WK2 & NOT255 # 0
103      FP = FP2 ; SETLP     ;!  IN CASE SPECIFIED STRING INCLUDES NEWLINE(S)
104      -> 2
105   !
106   SW('B'):PW('B'): SEEK ; -> SEEK FAIL %IF F < 0
107      FP = FP1       ;! NO CHANCE OF 'FP' GOING TO NEXT LINE, SO NO 'SETLP'
108      -> 2
109   !
110   !
111   SW('U'):PW('U'):  SEEK ; -> SEEK FAIL %IF F < 0
112   U1: -> 2 %IF FP = FP1
113      WK1 = LIST(FP) & PF
114      LIST(FP) = LIST(FP1)
115      WK2 = WK1
116   U2: FREE = FREE + 1
117      -> U3 %IF WK2 = FP1
118      WK2 = LIST(WK2) & PF ; -> U2
119   U3: LIST(WK2) = ASL ; ASL = WK1
120      MOD1 = 1       ;! SET IT NON-ZERO TO MARK MODIFICATION TO FILE
```

```
121         -> 2
122     !
123     !
124     ! 'DELETE' AND 'UNTIL' ARE VERY SIMILAR WHEN DEFINED AS LINE-BY-LINE
125     ! OPERATIONS.
126     !   'UNTIL' REMOVES FP -> FP1, 'DELETE' FP1 -> FP2, SO:-
127     !
128     SW('D'):PW('D'): SEEK ; -> 3 %IF F = -1   ;! CAN'T LOOP SIMPLY  ******
129         WK2 = CB(WK1)
130         CL = CL + ( WK2 >> 8 ) %IF WK2 & NOT255 # 0
131         FP = FP1 ; FP1 = FP2 ; -> U1
132     !
133     !
134     !
135     !
136     !   FOR REASONS OF EFFICIENCY, ROUTINES 'I' AND 'G' DO NOT FORMALLY
137     ! SPLIT THE FILE THEN USE 'JOIN ON' TO REASSEMBLE IT..
138     !     THE TECHNIQUE USED IS TO SAVE THE CELL CURRENTLY IN LIST(FP)
139     ! OVERWRITE LIST(FP) WITH THE FIRST CELL OF THE INSERTION THEN RESTORE
140     ! THE ORIGINAL LIST(FP) AT THE END OF THE INSERTION.
141     ! IN 'G' THIS CAN BE PROGRAMMED RATHER MORE BRIEFLY BECAUSE THE
142     ! INSERTION IS ALREADY IN A LIST, NOT HELD AS A PARAMETER IN THE
143     ! COMMAND BUFFER.
144     SW('I'): WK2 = WK1 + (CB(WK1) & 255)
145         CHECK SPACE
146         WK1 = WK1 + 1
147         WK3 = LIST(FP)
148         LIST(FP) = ( CB(WK1) & DF )!!( ASL & PF )
149         %IF WK1 # WK2 %THENSTART    ;! MORE THAN SINGLE CHARACTER
150         WK1 = WK1 + 1
151     %CYCLE  WK1 = WK1,1,WK2
152         STORE( CB(WK1) )
153     %REPEAT
154         %FINISH
155     !
156         STORE(0)          ;!      GET FREE CELL FROM ASL .......
157         LIST(P) = WK3     ;!      ...... AND RESTORE ORIGINAL 'LIST(FP)'
158     !
159         FP = P ; P = 0
160         SET LP
161         MOD1 = 1
162         -> 2
163     !
164     !
165     SW('G'): IN = CB(WK1)     ;!  FETCH ONE LINE FROM CHANNEL 'IN'
166         CHECK SPACE
167         STATUS(0) = 0 ; -> EOF %IF STATUS(IN) # 0
168         FP = LP
169         P = 0 ; TEMP = ASL ; READ INPUT
170     G1: WK1 = LIST(TEMP) ; LIST(TEMP) = LIST(FP) ; LIST(FP) = WK1
171         P = FP %IF P = TEMP     ;! P IS NEVER = 0 HERE
172         LIST(P) = ( LIST(P) & DF )!!( TEMP & PF )
173     !
174         LP = TEMP ; FP = LP
175         TEMP = 0 ; P = 0 ; IN = 1
176         MOD1 = 1
177         -> 2
178     PW('G'): %IF IN = 0 %THENSTART
179           CLOSE INPUT ; IFC(0) ; ABORT(3) %IF BATCH = ON
180         %FINISH
```

```
181       -> G1 %IF P # 0 ; -> EOF
182    !
183    !
184    !
185    !
186    !    THIS ROUTINE MOVES FP BACKWARD THE NUMBER OF LINES SPECIFIED
187    ! IN CB(WK1). FP IS LEFT AT THE START OF THE TARGET LINE.  L0
188    ! HAS THE SPECIALLY DEFINED EFFECT OF MOVING FP TO THE START OF THE
189    ! CURRENT LINE
190    SW('L'):  LTEMP = CB(WK1)
191       %IF LTEMP = 0 %THENSTART
192          FP = LP ; -> 2      ;!  GO TO START OF LINE
193       %FINISH
194    !
195       CL = CL - LTEMP  ;!  BATCH MACRO'S USE 'M-' EVEN THOUGH USER CAN'T
196    L0: WK1 = MAIN ; WK3 = 0
197    !
198       %IF LIST(MAIN) # 0 %THENSTART    ;! IF 'MAIN' IS NON-EMPTY
199    L1:    %IF WK1 # LP %THENSTART      ;! IS THIS THE 'CURRENT LINE'
200              WK2 = LIST(WK1) ; WK1 = WK2 & PF
201              WK3 = WK3 + 1 %IF WK2&DF = NLDF
202              -> L1
203          %FINISH
204       %FINISH
205       LTEMP = WK3 - LTEMP
206       %IF LTEMP < 0 %THENSTART
207          LTEMP = LTEMP + LC
208          -> TL
209       %FINISH
210       LP = MAIN ; FP = LP
211    L2: -> 2 %IF LTEMP = 0     ;!  NOTHING TO DO
212       %IF LTEMP < 0 %THENSTART  ;! ALREADY AT TOP SO CAN'T GO BACK
213          ERROR(1) ; -> 0
214       %FINISH
215       I = 'M' ; WK1 = LTEMP ; -> MO
216    !
217    !
218    !
219    !
220    !
221    !
222    !    THIS ROUTINE MOVES FP FORWARD THE NUMBER OF LINES SPECIFIED
223    ! BY CB(WK1) . FP IS LEFT AT THE START OF THE TARGET LINE.
224    ! MO HAS THE SPECIALLY DEFINED EFFECT OF MOVING FP TO THE END OF THE
225    ! CURRENT LINE
226    SW('M'):  WK1 = CB(WK1)
227    MN: CL = CL + WK1    ;!  UPDATE BATCH CURRENT LINE NO.
228    MO: WK2 = 0
229    M1: WK3 = LIST(FP)
230       %IF WK3 # 0 %THENSTART   ;!  BUFFER END MARKER? - NO
231          %IF WK3&DF # NLDF %THENSTART
232             FP = WK3 & PF ; -> M1
233          %FINISH
234          -> 2 %IF WK1 = 0      ;! GO TO END OF LINE ?
235          LP = WK3 & PF ; FP = LP
236          WK2 = WK2 + 1
237    M2:    -> M1 %IF WK2 # WK1     ;! COUNT NOT EXHAUSTED
238          -> 2
239       %FINISH
240    !
```

```
241       %IF STATUS(1) # 0 %THENSTART
242          STATUS(1) = 2 %IF LVL = 0 ; -> EOF
243       %FINISH
244    !
245       FILL MAIN
246    PW('M'):PW('N'): -> M1 %IF WK1 = 0 ; -> M2
247    !
248    !
249    !
250    !
251    !  FIND A PARTICULAR LINE, GIVEN ITS NUMBER ON A LISTING PRODUCED
252    !  BY THE '$LIST' FACILITY
253    SW('N'):  %STOP %UNLESS BATCH = ON      ;!  ONLY AVAILABLE IN BATCH MODE
254       WK1 = CB(WK1) - CL     ;!  HOW MANY LINES TO MOVE ?
255       ABORT(2) %IF WK1 < 0                ;!  CAN'T GO BACKWARDS
256       -> MN %IF WK1 > 0              ;! MOVE REQ'D NO OF LINES
257       FP = LP ; -> 2                 ;!  ALREADY AT REQ'D LINE
258    !
259    !
260    !
261    !
262    !     THIS ROUTINE PRINTS THE NUMBER OF LINES SPECIFIED IN CB(WK1)
263    ! STARTING WITH THE CURRENT LINE . FP IS LEFT AT THE START OF THE
264    ! LAST LINE PRINTED.  FP IS MARKED BY AN UP-ARROW WHICH IS SUPPRESSED
265    ! IF AT THE BEGINNING OF A NON-NULL LINE.
266    !    OUTPUT GOES TO CHANNEL(0). THIS WOULD BE THE USER'S CONSOLE IN
267    ! AN INTERACTIVE ENVIRONMENT, OTHERWISE A FAST LISTING DEVICE.
268    SW('P'):  WK1 = CB(WK1)
269       CL = CL + WK1 - 1      ;!  UPDATE  BATCH CURRENT LINE NUMBER
270       PFLAG = 1 ; WK1 = -WK1 + 1 ; WK2 = LP
271    !
272    PW('P'):P1: -> P2 %IF LIST(WK2) # 0    ;! IF NOT AT BOTTOM OF BUFFER
273       %IF STATUS(1) # 0 %THENSTART
274          STATUS(1) = 2 %IF LVL = 0 ; -> EOF
275       %FINISH
276       FILL MAIN
277    P2: WK3 = LIST(WK2) ; -> P1 %IF WK3 = 0  ;! IF AT END OF BUFFER
278       WK2 = WK3 & PF ; WK3 = WK3 >> 11   ;!  SEPARATE POINTER FROM CHARACTER
279       PRINT SYMBOL(WK3)
280       %IF WK1 = 0 %THENSTART                  ;!  COUNT EXHAUSTED ?
281          PRINT SYMBOL(UP ARROW) %IF WK2 = FP %AND (FP # LP %OR WK3 = NL)
282       %FINISH
283       %IF WK3 = NL %THENSTART                  ;! AT END OF LINE ?
284          -> 2 %IF WK1 = 0            ;! COUNT EXHAUSTED ?
285          LP = WK2 ; FP = LP
286          WK1 = WK1 + 1
287       %FINISH
288       -> P2
289    !
290    !
291    !
292    !
293    !    ROUTINES 'R'/'S' STEP FP BKWD/FWD THE NUMBER OF CHARACTERS
294    ! SPECIFIED IN CB(WK1).  AN ERROR MESSAGE IS PRODUCED IF AN ATTEMPT
295    ! IS MADE TO MOVE FP OUTSIDE THE CURRENT LINE BY THIS MEANS.
296    SW('R'):  WK1 = CB(WK1)
297       WK2 = LP ; WK3 = 0
298    R1: -> R2 %IF WK2 = FP
299       WK2 = LIST(WK2) & PF
300       WK3 = WK3 + 1 ; -> R1
```

```
!
R2: WK3 = WK3 - WK1
    -> EOL %IF WK3 < 0
    FP = LP ; WK1 = 0
R3: -> 2 %IF WK1 = WK3
    FP = LIST(FP) & PF ; WK1 = WK1 + 1
    -> R3
!
SW('S'): WK1 = CB(WK1)
    WK4 = FP
%CYCLE  WK2 = 1,1,WK1
PW('S'):S1: WK3 = LIST(WK4)
    %IF WK3 = 0 %THENSTART
        %IF STATUS(IN) # 0 %THENSTART
            ERROR(2) ; -> 0
        %FINISH
        FILL MAIN ; -> S1
    %FINISH
    -> EOL %IF WK3 & DF = NLDF
    WK4 = WK3 & PF
%REPEAT
    FP = WK4
    -> 2
!
!
!
!
!
!   THIS ROUTINE TRANSFERS THE OUTPUT FILE TO OUTPUT STREAM 1 THEN
! RETURNS CONTROL TO THE OPERATING SYSTEM
! PUT OUTPUT STRAIGHT ONTO FINAL OUTPUT IF WE ARE AT THE TOP OF
! THE FILE AND EITHER, (1) THE FILE IS SHORTER THAN ONE COMPLETE
! BUFFER LOAD, OR (2) IT IS ALREADY ON ONE OF THE WORK FILES, OTHERWISE
! FIRST COPY IT OVER TO A WORK FILE THEN DO THE 'CLOSE'.
!
SW('C'): %IF LC = 0 %THENSTART
    OUTPUT(1) = SOURCE FILE %IF STATUS(IN) # 0 %OR INPUT(1) # SOURCE FILE
    %FINISH
    IFC(0) ; CLOSE INPUT  ;!  CLOSE OFF COMMAND STREAM
    %IF LISTING = OFF %THEN -> T1 %ELSESTART
C1:   PUNCH(MAIN)
      MAIN = ASL ; ASL = LIST(ASL) & PF ; LIST(MAIN) = 0
      LP = MAIN ; FP = LP
      FILL MAIN ; -> C1 %IF STATUS(1) = 0
PW('C'): STATUS(1) = 2 ; -> TC
    %FINISH
!
!
!
!
!   THIS ROUTINE EITHER SETS FP TO THE TOP OF THE FILE OR REWINDS
! THE SUBSIDIARY INPUT FILE DEPENDING ON CB(WK1) BEING #2 AND =2
! RESPECTIVELY
SW('T'): WK1 = CB(WK1)
    %IF WK1 = 2 %THENSTART    ;! REWIND SECONDARY INPUT
    IFC(2) ; CLOSE INPUT ; STATUS(2) = 0 ; -> 2
    %FINISH
!
    LISTING = OFF
!
```

```
TL: -> QUICK TOP %IF LC = 0
!
T1: %IF MOD1 # 0 %OR INPUT(1) = SOURCE FILE %THENSTART
TC: PUNCH(MAIN)      ;!  'PUNCH' MUST TAKE CARE OF EMPTY FILES
   %IF I = 'C' %THENSTART
      OTC(2) ; PLC = PLC - 1 ; PAGE %IF PLC >= 0
      CLOSE OUTPUT         ;!  ON LISTING DEVICE
   %FINISH
T2: %FAULT9 -> T5
   IFC(1) ; OTC(1)
!
T3: %IF STATUS(1) # 2 %THENSTART
   %CYCLE  WK1 = 1,1,AMAX
      WK2 = 0 ; WK3 = 0 ; WK4 = 0
      READ SYMBOL(WK2) ; READ SYMBOL(WK3) ; READ SYMBOL(WK4)
     LIST(WK1) = ( ( ( SIXBIT(WK4)<<6 )!!SIXBIT(WK3) )<<6 )!!SIXBIT(WK2)
   %REPEAT
T4:%CYCLE  WK2 = 1,1,WK1
      WK3 = LIST(WK2)
      %CYCLE  WK4 = 1,1,3
         PRINT SYMBOL( SEVENBIT(WK3) ) %IF WK3 & 63 # 0
         WK3 = WK3 >> 6
      %REPEAT
   %REPEAT
      -> T3
!
T5:  LIST(WK1) = ( ( ( SIXBIT(WK4)<<6 )!!SIXBIT(WK3) )<<6 )!!SIXBIT(WK2)
      STATUS(1) = 2 ; -> T4
   %FINISH
!
   CLOSE INPUT ; CLOSE OUTPUT ; OTC(0)
   -> STOP %IF OUTPUT(1) = SOURCE FILE        CLOSING
.1 COMPLETE                                   AND START NEXT PASS
T3: %IF STATUS(1) ≠ 2 %THENSTART              ONLY
   WK4 = ADDR( LIST(1) )
   %CYCLE  WK1 = 1,1,AMAX
      WK2 = 0 ; WK3 = 0
      READ SYMBOL(WK2) ; READ SYMBOL(WK3)     E %PRINTTEXT '.2'
      %LAC WK3 ; %OPR 1048 ;!  SWHA
      %TAD WK2
      %DAC* WK4 ; %ISZ WK4
   %REPEAT
T4: WK4 = ADDR( LIST(1) )                     .E
   %CYCLE  WK2 = 1,1,WK1                      T FILE
      %LAC* WK4 ; %DAC WK3
      %AND (127 ; %JMS PRINT SYMBOL
      %LAC WK3 ; %OPR 1048 ;!  SWHA
      %AND (127 ; %JMS PRINT SYMBOL
      %ISZ WK4
   %REPEAT
      -> T3
!
T5:   %LAC WK3 ; %OPR 1048 ; %TAD WK2 ; %DAC* WK4

>MP2
      STATUS(1) = 2 ; -> T4                   ) ZERO
   %FINISH

!
QUICK TOP:
!
```

```
361   TL: -> QUICK TOP %IF LC = 0
362   !
363   T1: %IF MOD1 # 0 %OR INPUT(1) = SOURCE FILE %THENSTART
364   TC: PUNCH(MAIN)     ;!  'PUNCH' MUST TAKE CARE OF EMPTY FILES
365       %IF I = 'C' %THENSTART
366          OTC(2) ; PLC = PLC - 1 ; PAGE %IF PLC >= 0
367          CLOSE OUTPUT           ;!  ON LISTING DEVICE
368       %FINISH
369   T2: %FAULT9 -> T5
370       IFC(1) ; OTC(1)
371   !
372   T3: %IF STATUS(1) # 2 %THENSTART
373       %CYCLE  WK1 = 1,1,AMAX
374          WK2 = 0 ; WK3 = 0 ; WK4 = 0
375          READ SYMBOL(WK2) ; READ SYMBOL(WK3) ; READ SYMBOL(WK4)
376        LIST(WK1) = ( ( ( SIXBIT(WK4)<<6 )!!SIXBIT(WK3) )<<6 )!!SIXBIT(WK2)
377       %REPEAT
378   T4: %CYCLE  WK2 = 1,1,WK1
379          WK3 = LIST(WK2)
380          %CYCLE  WK4 = 1,1,3
381             PRINT SYMBOL( SEVENBIT(WK3) ) %IF WK3 & 63 # 0
382             WK3 = WK3 >> 6
383          %REPEAT
384       %REPEAT
385          -> T3
386   !
387   T5:  LIST(WK1) = ( ( ( SIXBIT(WK4)<<6 )!!SIXBIT(WK3) )<<6 )!!SIXBIT(WK2)
388          STATUS(1) = 2 ; -> T4
389       %FINISH
390   !
391       CLOSE INPUT ; CLOSE OUTPUT ; OTC(0)
392       -> STOP %IF OUTPUT(1) = SOURCE FILE    ;! WE'RE CLOSING
393       WK1 = INPUT(1)      ;! OTHERWISE SWOP WORK FILES AND START NEXT PASS
394       WK1 = WORK2 %IF WK1 = SOURCE FILE ;! FIRST PASS ONLY
395       INPUT(1) = OUTPUT(1) ; OUTPUT(1) = WK1
396   !
397       PRINT SYMBOL('*')
398       %IF INPUT(1) = WORK1 %THEN %PRINTTEXT '.1' %ELSE %PRINTTEXT '.2'
399       %PRINTTEXT ' COMPLETE'
400       NEWLINE
401   !
402       %FINISH%ELSESTART
403          IFC(1) ; CLOSE INPUT       ;! REWIND INPUT FILE
404          OTC(1) ; CLOSE OUTPUT      ;! START NEW OUTPUT FILE
405          OTC(0)
406       %FINISH
407   !
408       STATUS(1) = 0
409   !
410       %IF I = 'C' %THENSTART        ;! INITIATE 'CLOSE'
411          OUTPUT(1) = SOURCE FILE ; -> T2
412       %FINISH
413   !
414       %FAULT 9 -> FAULT9     ;! IT'S 'T' OR 'L'
415       IN = 1          ;! STATUS(1) = 0 AT THIS POINT
416       INIT ;!     THIS ALSO RESETS LINE-COUNTER 'LC' TO ZERO
417       FILL MAIN
418   !
419   QUICK TOP:
420   !
```

```
421   PW('T'):PW('L'): LP = MAIN ; FP = LP
422       -> 2 %IF I = 'T' ; -> L2
423   !
424   !
425   !
426   !
427   !    THIS IS EFFECTIVELY A JUMP INSTRUCTION FOR THE SEQUENCING
428   ! MECHANISM . CONTROL IS TRANSFERRED TO THE COMMAND IMMEDIATELY
429   ! FOLLOWING THE NUMBER OF CLOSING BRACKETS SPECIFIED IN CB(WK1).
430   ! PAIRED BRACKETS ARE IGNORED
431   SW('X'): WK1 = CB(WK1) ; -> 4
432   !
433   !
434   !
435   !
436   ! SET BATCH MODE FLAG.  THIS SHOULD CHECK(SOMEHOW?) THAT IT IS THE
437   ! VERY FIRST COMMAND ISSUED ON A PARTICULAR RUN
438   SW('Y'): ABORT(0) %IF FREE # LISTMAX-1
439   !   OUTPUT(0) = OUTPUT(2)     ;!  SWITCH MONITOR OUTPUT TO PRINTER
440         BATCH = ON ; -> 2
441   ! CALL:  $ONEPASS = YZ2     :-  SWITCH TO BATCH MODE AND TURN ON LISTING
442   !
443   !
444   !
445   ! TURN ON LISTING.
446   SW('Z'): FILL MAIN %IF MAIN # LP
447   PW('Z'): WK2 = CB(WK1)
448       %IF WK2 # 0 %THENSTART
449           LISTING = LISTSW %IF WK2 = 1
450           LISTING = ON %IF WK2 = 2
451           -> 2 %IF WK2 # 3
452           ABORT(0) %IF FREE # LISTMAX-1
453           LISTING = OFF
454       %FINISH
455   !
456       LISTSW = LISTING
457       -> 2 %IF LISTING = OFF
458       LISTING = OFF
459       OTC(2)
460       LP LIST
461       %PRINTTEXT '*****' ; NEWLINE ; PLC = PLC - 1
462       OTC(0)
463       -> 2
464   !
465   ! CALLS:  ALL CALLS MOVE THE WINDOW SO THAT LP = MAIN
466   ! Z0  SAVE CURRENT STATE THEN TURN OFF LISTING
467   ! Z1  RESET LISTING
468   ! Z2  TURN LISTING ON
469   ! Z3  TURN LISTING OFF
470   !
471   !
472   !
473   !
474   STOP: %ENDOFPROGRAM
```

```
>EDIT  EDIT4C/EDIT4C DSK


::
****  GENERATING A SPECIAL VERSION OF  EDIT4C   29/10/70   TO
****  EXPLOIT MORE FULLY A PDP-15 WITH DISK OR DRUM
****  THIS CONTAINS MACHINE CODE AND IS NOT MACHINE INDEPENDENT


::
M-5(PK)5


F/T3:/ P
$U/T5:/P
K::
T3: %IF STATUS(1) # 2 %THENSTART
      WK4 = ADDR( LIST(1) )
      %CYCLE  WK1 = 1,1,AMAX
         WK2 = 0 ; WK3 = 0
         READ SYMBOL(WK2) ; READ SYMBOL(WK3)
         %LAC WK3 ; %OPR 1048    ;!  SWHA
         %TAD WK2
         %DAC* WK4 ; %ISZ WK4
      %REPEAT
T4: WK4 = ADDR( LIST(1) )
      %CYCLE  WK2 = 1,1,WK1
         %LAC* WK4 ; %DAC WK3
         %AND (127) ; %JMS PRINT SYMBOL
         %LAC WK3 ; %OPR 1048    ;!  SWHA
         %AND (127) ; %JMS PRINT SYMBOL
         %ISZ WK4
      %REPEAT
      -> T3
!
T5:   %LAC WK3 ; %OPR 1048 ; %TAD WK2 ; %DAC* WK4
    OUTPUT FILE IS CALLED  'EDIT4C DSK'
::
M-PK
$CLOSE
```