```
1   !  FORMAT MK. 2                JULY '72
2   !
3   !
4   !           CONTROL SYMBOLS AVAILABLE
5   ! <>D: ENTER DIAGRAM MODE - E.G. <>D10
6   ! <>E: END OF INPUT FILE - E.G. <>E
7   ! <>F: DECODE REST OF LINE AS OPTION SETTING FLAGS( SEE BELOW)
8   ! <>I: INDENT SPECIFIED NUMBER OF TAB STOPS - E.G. <>I3
9   !      OVERRIDDEN FOR IMMEDIATELY FOLLOWING LINE BY <>L
10  ! <>L: NEWLINE($) - E.G. <>L6
11  ! <>N: NEW PAGE - E.G. <>N
12  ! <>P: START NEW PARAGRAPH - E.G. <>P
13  ! <>S: NEW SECTION - E.G. <>S
14  ! <>T: TAB TO EITHER - (A) THE SPECIFIED STOP - E.G. <>T4
15  !                  OR - (B) THE NEXT STOP - E.G. <>T
16  ! <>V: VERIFY THAT AT LEAST THE SPECIFIED NUMBER OF LINES ARE
17  !      LEFT ON THE CURRENT PAGE - E.G.  <>V3
18  !
19  !
20  !         OPTION SETTING - FLAG LINES
21  !    THE BEHAVIOUR OF THE PROGRAM MAY BE MODIFIED BY SETTING OR
22  ! UNSETTING VARIOUS OPTIONS.  THIS IS DONE BY A FLAG LINE CONSISTING OF
23  ! THE CONTROL SYMBOL <>F FOLLOWED BY VARIOUS LETTERS TO DENOTE THE
24  ! OPTIONS REQUIRED.  THE LETTER ALONE SETS THE OPTION, THE SAME
25  ! LETTER PRECEDED BY '\' UNSETS IT.  IF IT IS PRECEDED BY '?', A
26  ! QUERY WILL BE PRINTED ON THE CONSOLE ASKING HOW THE OPTION IS
27  ! TO BE SET ON THIS RUN OF THE PROGRAM.  THESE QUESTIONS MAY BE ANSWERED
28  ! EITHER WITH Y(ES)/N(O) OR WITH AN EXPLICIT SETTING OF A PARTICULAR
29  ! OPTION.  YES/NO IS TAKEN AS A REPLY TO THE QUESTION ASKED BUT
30  ! A REPLY SPECIFIC TO A PARTICULAR OPTION SETS THAT OPTION WHICH
31  ! MAY BE DIFFERENT THAN THE ONE TO WHICH THE CONSOLE QUESTION
32  ! REFERRED.  NOTE THAT SINCE REDUNDANT QUESTIONS ARE SUPPRESSED
33  ! THE TOTAL NUMBER OF REPLIES REQUIRED WILL IN GENERAL DEPEND
34  ! ON THE ORDER IN WHICH THEY ARE RECEIVED.
35  !
36  ! P: PRIMARY(FORMATTED) OUTPUT IS REQUIRED
37  ! I: PRIMARY OUTPUT IS TO BE IN ISO NOT THE SECONDARY CODE
38  ! J: PRIMARY OUTPUT IS NOT TO BE RIGHT-JUSTIFIED
39  ! C: CARRY-OVER WORDS ARE TO BE PRINTED IN PRIMARY OUTPUT
40  !
41  ! S: A NEW SOURCE FILE IS REQUIRED
42  ! L: A PAGED, LINE-NUMBER LISTING IS REQUIRED OF THIS NEW FILE.
43  !    ** 'L' IS UNAVAILABLE UNLESS 'S' IS ALSO REQUESTED **
44  !
45  ! E: PRINT END-OF-PAGE LINE NUMBERS ON THE CONSOLE
46  !
47  ! R: TABS ARE TO BE RELATIVE(NOT ABSOLUTE)
48  ! T: SET TAB STOPS
49  ! M: SET LEFT-HAND MARGIN
50  ! O: SET ORIGIN - SECTION,PAGE;
51  ! D: SET PAGE DIMENSIONS - LINES/PAGE,CHARS/LINE;
52  ! U: SET SHIFT CHARACTERS - UPPERCASE,UNDERLINE;
53  ! X: SET NUMBER OF EXTRA SPACES AFTER -----. (SENTENCE TERMINATOR)
54  ! Z: SET NUMBER OF PHYSICAL LINES PER LOGICAL NEW LINE
55  ! G: SET INTER-PARAGRAPH GAP( NUMBER OF LOGICAL NEWLINES )
56  !
57  ! W: GENERATE WHOLE FILE(NOT SINGLE PAGES)
58  !    ** \W IMPLIES P \S \L **
59  !
60  !
```

```
61    !        DEFAULT OPTION SETTINGS
62    !
63    ! <>F P I S \L \E
64    ! <>F J C \R
65    ! <>F W
66    ! <>F D 72,60:  0 0,0:
67    ! <>F T 8,16,24,32,40,48,56,64,72;
68    ! <>F U @,_: X2: Z1: G2: M0:
69    !
70    !     ** COLONS IN THE ABOVE SHOULD BE READ AS SEMI-COLONS **
71    !
72    !
73    !
74    !
75    !
76    %BEGIN
77    ! DEFINE I/O STREAMS
78    %OWNINTEGER MONITOR=0,INPUT=1,OUTPUT1=1,NEW SOURCE=2,LISTING=3
79    !
80    !
81    !    ** DEFAULTS **
82    !
83    !  -1 = 77777...(8) => 'ON', 0 => 'OFF'
84    !
85    ! F:     FORMATTED(PRIMARY) OUTPUT                    IMPLICIT
86    ! J:     RIGHT JUSTIFIED LINES                        ON
87    ! ISO:   PRIMARY OUTPUT IN ISO CODE                   YES => ON
88    ! REL:   RELATIVE(NOT ABSOLUTE) TABS                  NO => OFF
89    ! OVER:  CARRY-OVER WORD PRINTED AT END OF PAGE       YES => ON
90    ! COPY:  IMAGE COPY('DIAGRAM') MODE                   OFF
91    ! W:     WHOLE FILE(NOT SINGLE PAGES) GENERATED       YES => ON
92    ! E:     AT END OF EACH PAGE OF (P) OUTPUT, PRINT
93    !        INPUT LINE NO. ON MONITOR STREAM             OFF
94    !
95    ! S:     GENERATE NEW SOURCE FILE                     IMPLICIT
96    !
97    ! L:     GENERATE PAGED,LINE-NUMBERED LISTING OF (S) IMPLICIT
98    !
99    !  **IMPLICIT DEFAULTS ARE SET 'ON' IF AN OUTPUT STREAM IS CONNECTED,
100   !    'OFF' IF NOT**
101   !
102   !
103   !
104   !  DEFAULTS: SECTION = 0, PAGE = 0
105   !            CHARS ON LINE = 72, LINES ON PAGE = 60
106   !            UPPERCASE SHIFT('UCS') = '@', UNDERLINE SHIFT('ULS') = '_'
107   !            SENTENCE SEPARATING SPACES('DOTSP') = 2
108   !
109   !
110   !
111   !
112   %OWNINTEGER  ON=-1,OFF=0,UC=128,UL=256,SP=512
113   %OWNINTEGER  RH9=511 ; %INTEGER  LH9; LH9 = \RH9
114   !
115   ! ** THIS DECLARATION SETS THE SIZE OF THE PROGRAM'S DATA BUFFERS **
116   %OWNINTEGER  AL=100, BCL=256
117   !
118   !
119   %INTEGER  VSKIPB,VSKIPC,HSKIPB,TOTAL EXTRA B,HSKIPC
120   %INTEGER  RUN FLAG, CSYM,LAST CSYM,INDENT,REL,DOTSP,PGAP
```

```
121   !
122   %OWNINTEGER  J=0,F=-1,L=0,S=0,ISO=-1,COPY=0,OVER=-1,E=0
123   !
124   %INTEGER  NLFACTOR, CHARS ON LINE, LINES ON PAGE
125   %OWNINTEGER  UC$='@', UL$='_'    ;!  UPPER CASE SHIFT, UNDERLINE SHIFT
126   !
127   %INTEGER  EMPTY LINES
128   %INTEGER  INPUT LINE NO, FORMAT LINE NO, LISTING LINE NO
129   %OWNINTEGER  W = -1        ;!  INITIALISED TO 'ON'
130   %INTEGER  PAGE,SECTION,REQD PAGE,REQD SECTION,WSTAT
131   %INTEGER  COPY START LINE,SPEC COPY SIZE,ACTUAL COPY SIZE
132   !
133   %INTEGERARRAY  A(0:AL+2), B,C(0:BCL-1)
134   %INTEGER  APTR,AMAX,BPTR,B1,B2,CPTR,C1,C2
135   !
136   !
137   !  THE FOLLOWING ARRAY DEFINES THE POSITIONS OF UP TO 20 TAB-STOPS
138   !  IN ELEMENTS 1:20 AND A LEFT-HAND MARGIN INDENT IN ELEMENT 0.
139   !  THIS LATTER IS DIFFERENT FROM THE INDENT FACILITY(<>I) IN
140   !  THAT IT MOVES EVERYTHING TABTAB(0) PHYSICAL SPACES TO THE RIGHT,
141   !  TAB-STOPS INCLUDED. THE <>I FACILITY IS DEFINED IN TERMS OF TAB-STOPS
142   !  NOT PHYSICAL SPACES.
143   %OWNINTEGERARRAY  TABTAB(0:20) =  0,7,15,23,31,39,47,55,63,71,-1,
144                                     -1,-1,-1,-1,-1,-1,-1,-1,-1,-1
145   !
146   !
147   !
148   !
149   !
150   !
151   %INTEGERFN  CASE SHIFT(%INTEGER  I)
152   %INTEGER  J
153      J = I&127
154      %RESULT = J %UNLESS I&UC = 0 %AND 'A' <= J <= 'Z'
155      %RESULT = J+32     ;!  IT'S A LOWER CASE LETTER
156   %END
157   !
158   !
159   %ROUTINE  INITIALISE
160      %INTEGERFN  DEFAULT(%INTEGER  STREAM)  ;! SET DEFAULT OUTPUT OPTIONS
161         SELECT OUTPUT(STREAM)
162         %RESULT = OFF %IF OUTDEV = 0     ;!  IF NO STREAM CONNECTED
163         %RESULT = ON
164      %END
165   !
166      F = DEFAULT(OUTPUT1); S = DEFAULT(NEW SOURCE)
167      L = DEFAULT(LISTING) & S    ;!  NO LISTING WITHOUT NEW SOURCE FILE!
168   !
169      RUN FLAG = ON ; CSYM = -1 ; LAST CSYM = -1 ; INDENT = 0 ; REL = 0
170      BPTR = -1 ; CPTR = -1
171      VSKIPB = 0 ; VSKIPC = 0 ; HSKIPC = 0
172      HSKIPB = TABTAB(0); TOTAL EXTRA B = HSKIPB
173      INPUT LINE NO = 0 ; FORMAT LINE NO = 0 ; LISTING LINE NO = 0
174      PAGE = 0 ; SECTION = 0     ;!      ** RESET DEFAULT VALUES **
175      COPY START LINE = 0 ; SPEC COPY SIZE = 0 ; ACTUAL COPY SIZE = 0
176      CHARS ON LINE = 72 ; LINES ON PAGE = 60
177      NLFACTOR = 1 ; DOTSP = 2 ; PGAP = 2
178      B1 = 0 ; B2 = 1 ; C1 = 0 ; C2 = 1
179      B(0) = -1 ; B(1) = 0 ; C(0) = -1 ; C(1) = 0
180   %END
```

```
!
!
%ROUTINE  PRINT SOURCE LINE
%INTEGER  I
   SELECT OUTPUT(MONITOR) ; NEWLINE
   WRITE(INPUT LINE NO,4) ; SPACES(2)
   %CYCLE  I = 2,1,AMAX
      PRINT SYMBOL(94) %IF I = APTR     ;!  UP-ARROW
      PRINT SYMBOL( A(I) ) ; %RETURN %IF A(I) = NL
   %REPEAT
   NEWLINE
%END
!
!
%ROUTINE  READ INPUT LINE
%INTEGER  I,P
   SELECT INPUT(INPUT)
   EMPTY LINES = -1   ;! 'EMPTY LINES' USED BY DIAGRAMS
1: EMPTY LINES = EMPTY LINES + 1
   %CYCLE  I = 2,1,AL+1
      READ SYMBOL(P)
      %IF P = NL %THENSTART
         INPUT LINE NO = INPUT LINE NO + 1
         -> 1 %IF I = 1 ; -> 2
      %FINISH
      P = P-32 %IF 'A' <= P-32 <= 'Z'     ;!  LOWER -> UPPER CASE
      A(I) = P
   %REPEAT
   PRINT SOURCE LINE ; %PRINTTEXT '*TOO LONG' ; NEWLINE
2: A(I) = ' ' ; AMAX = I
%END
!
!
!
%ROUTINE  RUNOUT(%INTEGER  I)
1: I = I-1 ; %RETURN %IF I < 0
   PRINT SYMBOL(0)
   -> 1
%END
!
!
!
%ROUTINE  PRINT ISO LINE(%INTEGERARRAYNAME  A, %INTEGER  X,Y)
%OWNINTEGER  CR=13, STOPCODE=20
%INTEGER  I,J,P,Q,UFLAG
%OWNINTEGER  STOP = -1     ;!  -1 -> 'ON'
!
%ROUTINE  UNDERLINE(%INTEGER  I)
1: %IF I > 0 %THENSTART
      PRINT SYMBOL('_') ; I = I - 1 ; -> 1
   %FINISH
%END
!
%ROUTINE  PAGE
   NEWLINES(5)
   %IF OUTDEV = 1 %THENSTART    ;!  IF IT'S THE TELETYPE
      ! PRINT PAGE SEPARATOR
      PRINT SYMBOL('-'); SPACES(70); PRINT SYMBOL('-'); NEWLINES(7)
   %FINISHELSESTART
      ! SEPARATE PAGES ON E.G. PAPER TAPE WITH RUNOUT + STOP CODE
```

```
241         RUNOUT(5); PRINT SYMBOL(STOP CODE); RUNOUT(5); PRINT SYMBOL(CR)
242       %FINISH
243     %END
244     !
245     !
246     !
247     !
248       PAGE %IF STOP = ON          ;! .. ON FIRST CALL ONLY
249       STOP = OFF      ;!  TURN OFF SEPARATOR PRINTING MEANTIME
250       J = A(X) ; I = J&RH9 ; J = J>>9
251       NEWLINES(I) %IF I # 0
252       X = X+1                 ;!  SKIP OVER NEWLINE COUNT CELL
253       UFLAG = 0 ; SPACES(J)
254       %CYCLE  I = X,1,Y
255         P = A(I) ; UFLAG = I %IF P & UL # 0  ;! UNDERLINING REQ'D?
256         Q = 0 ; Q = P>>9 %IF P&LH9 # 0
257         %IF P = STOP CODE %THEN STOP=ON %ELSE PRINT SYMBOL(CASESHIFT(P))-
258         SPACES(Q) %IF Q # 0
259       %REPEAT
260       %IF UFLAG # 0 %THENSTART
261         PRINT SYMBOL(CR) ; PRINT SYMBOL(CR) ; SPACES(J)
262         %CYCLE  I = X,1,UFLAG
263           P = A(I) ; J = (P>>9) + 1
264           %IF P&UL = 0 %THEN SPACES(J) %ELSE UNDERLINE(J)
265         %REPEAT
266       %FINISH
267       %IF STOP = ON %THEN PAGE %ELSE NEWLINE ;!  IF A STOP CODE WAS FOUND
268     !
269     ! SEPARATOR PRINTING IS CURRENTLY TURNED ON: CLEAR IT IF IN
270     ! WHOLE PAGE MODE(W) SO THAT ONLY A SINGLE SEPARATOR IS
271     ! PRINTED BETWEEN CONSECUTIVE PAGES.  OTHERWISE LEAVE IT ON SO THAT
272     ! ON THE NEXT CALL, A SEPARATOR PRECEDES THE SINGLE PAGE AS WELL
273     ! AS FOLLOWING THIS ONE.
274       STOP = OFF %IF W = ON
275     %END
276     !
277     !
278     !
279     %ROUTINE  PRINT CODE2 LINE(%INTEGERARRAYNAME  A, %INTEGER  X,Y)
280     %OWNINTEGER  RESET=1        ;!  DO/DON'T INITIALISE ATLAS OUTPUT ROUTINE
281     %INTEGER  I,J,P,Q,UFLAG
282     !
283     %ROUTINE  PRINT ATLAS(%INTEGER  P)
284     ! ATLAS FLEXOWRITERS USE THE SAME TAPE CODE FOR RUNOUT AS FOR
285     ! UPPERCASE SHIFT.  IN ORDER TO BE SURE THAT THE PRINTER
286     ! ACTUALLY IS IN UPPERCASE BEFORE STARTING PRINTING FOR SERIOUS
287     ! IT IS DESIRABLE TO SEND AT LEAST ONE RUNOUT CHARACTER( = UPPER CASE
288     ! SHIFT )
289     %OWNINTEGERARRAY  A(0: 33)= 30,
290       18464, 18698, 7073, 22187, 22317, 22446, 21824, 19988, 19080, 7846,
291       8108, 7208, 7337, 7978, 6183, 4026, 6588, 6845, 6718, 7743,
292       6363, 6493, 7007, 18351, 20528, 24769, 10465,-127069,-126917,-126812,
293       13711, 7999, 14307
294     %OWNINTEGER  CASE=128
295     %INTEGER  I,K,X
296     !
297     %ROUTINE  PRINT(%INTEGER  P)
298     %INTEGER  Q,N
299       N = 0 ; Q = P
300     1: %IF Q # 0 %THENSTART
```

```
301          N = (Q&1)+N ; Q = Q >>1 ; -> 1
302       %FINISH
303       P = P!!16 %IF N&1 = 0     ;!  SET ODD PARITY IN TRACK 5
304       PRINT SYMBOL(P)
305    %END
306    !
307    %ROUTINE  SHIFT UP
308       PRINT(7) ; CASE = 128
309    %END
310    !
311    %ROUTINE  SHIFT DOWN
312       PRINT(6) ; CASE = 16
313    %END
314    !
315    !
316       P = P&127
317    !    UPPERCASE SHIFT CODE IS USED AS RUNOUT
318       %IF P = 0 %THENSTART ; SHIFT UP ; %RETURN ; %FINISH
319    !
320       K = P
321       K = 'A' %IF 'A' <= P <= 'Z'
322       K = 'A'+32 %IF 'A' <= P-32 <= 'Z'
323       K = '0' %IF '0' <= P <= '9'
324       %CYCLE  I = A(0),-1,1
325          -> 1 %IF A(I)&127 = K
326       %REPEAT
327    !  'SPACE' IS THE FIRST ENTRY IN 'A' SO IF THE CHARACTER IN
328    !  'K' IS NOT MATCHED, A SPACE IS AUTOMATICALLY SUBSTITUTED
329    !
330    1: I = A(I)
331       X = ((I<<1)>>8)     ;!  >>7  AND GET RID OF SIGN BIT
332       X = A(X) %IF I < 0  ;!  IS IT A POINTER? :  < 0 : YES
333    !
334    2: %IF X&CASE = 0 %THENSTART      ;!  MUST SHIFT CASE
335          %IF CASE = 128 %THEN SHIFT DOWN %ELSE SHIFT UP
336       %FINISH
337    !  NOTE THAT BIT 5 MUST BE SET SO THAT ANY CARRIES SKIP TO POSN. 6
338       PRINT( ((X!16)&127)+P-K )    ;!   DEALS WITH 'A'-'Z' ETC.
339       X = X>>8 ; %RETURN %IF X = 0
340       PRINT(21) ; -> 2     ;!  BACKSPACE TO OVERPRINT
341    %END;!ATLAS
342    !
343    !
344    !
345    !
346    %ROUTINE  NEWLINES(%INTEGER  I)
347    1: %IF I > 0 %THENSTART
348          PRINT ATLAS(NL) ; I = I - 1 ; -> 1
349       %FINISH
350    %END
351    !
352    %OWNINTEGER  BACKSPACE=8, UNDERSCORE=95, STOPCODE=20
353    !
354    %ROUTINE  UNDERLINE     ;!   USES 'J'
355       %IF UFLAG > 0 %THENSTART
356          %CYCLE  J = 1,1,UFLAG ; PRINT ATLAS(BACKSPACE) ; %REPEAT
357          %CYCLE  J = 1,1,UFLAG ; PRINT ATLAS(UNDERSCORE) ; %REPEAT
358       %FINISH
359       UFLAG = 0
360    %END
```

```
!
!
    %IF RESET # 0 %THENSTART
       PRINT ATLAS(0) ; RESET = 0     ;!  SET UPPER CASE SHIFT
    %FINISH
    J = A(X) ; I = J&RH9 ; J = J>>9
    NEWLINES(I) %IF I # 0
    X = X+1                ;!    SKIP OVER SPACE/NEWLINE COUNT CELL
    UFLAG = 0
1:  %IF J > 0 %THENSTART
       PRINT ATLAS(' ') ; J = J-1 ; -> 1
    %FINISH
    %CYCLE  I = X,1,Y
       P = A(I) ; Q = P>>9
       UNDERLINE %IF UFLAG > 0 %AND P&UL = 0
       UFLAG = UFLAG+1 %IF P&UL # 0
       P = CASE SHIFT(P)
       %IF P&127 # STOPCODE %THENSTART
          PRINT ATLAS(P)
2:        %IF Q > 0 %THENSTART
             PRINT ATLAS(' ') ; UFLAG = UFLAG+1 %IF UFLAG # 0
             Q = Q-1 ; -> 2
          %FINISH
       %FINISHELSESTART
          UNDERLINE %IF UFLAG > 0
          NEWLINES(5) ; RUNOUT(5)
          PRINT ATLAS(STOPCODE) ; RUNOUT(5)
          RESET = 1    ;!  PRECEDE NEXT OUTPUT WITH UPPERCASE SHIFT
          %RETURN
       %FINISH
    %REPEAT
3:  UNDERLINE %IF UFLAG > 0 ; NEWLINES(1)
%END      ;!   PRINT CODE2 LINE
!
!
!
%ROUTINE  PRINT(%INTEGERARRAYNAME  A,%INTEGERNAME P)
%INTEGER  I
! ***DESIGN FAILING***: PROGRAM WOULD GENERATE SINGLE PAGES MUCH
!            FASTER IF PRINTING WERE TURNED OFF AT A HIGHER LEVEL,
!              !!            AND ESPECIALLY BEFORE LINE STRETCHING.
! DON'T PRINT IF IN SINGLE PAGE MODE AND THE PAGE NO. IS WRONG
    %RETURN %IF W # ON %AND ( SECTION # REQD SECTION %OR %C
       PAGE # REQD PAGE )
!
    %RETURN %IF P < 0
    SELECT OUTPUT(OUTPUT1)
1:  I = P ; P = A(P)
    %IF P >= 0 %THENSTART
       %IF ISO = ON %THEN PRINT ISO LINE(A,I+1,A(I)-1) %C
          %ELSE PRINT CODE2 LINE(A,I+1,A(I)-1)
       -> 1
    %FINISH
%END
!
!
%ROUTINE  LIST(%INTEGERARRAYNAME  A, %INTEGERNAME  P)
%INTEGER  I,K,X,Y,Z,S
!
%ROUTINE  PAGE
```

```
421     !
422     %INTEGERFN  MODULO60(%INTEGER  N)
423     1: %RESULT = N+60 %IF N < 0 ; N = N-60 ; -> 1
424     %END
425     !
426     !
427        %IF  MODULO60(LISTING LINE NO) = 0 %THENSTART
428           NEWLINES(2)
429           PRINT SYMBOL('-') ; SPACES(78) ; PRINT SYMBOL('-')
430           NEWLINES(2)
431        %FINISH
432        LISTING LINE NO = LISTING LINE NO + 1 ; NEWLINE
433     %END
434     !
435        %RETURN %IF P < 0
436     1: I = P ; P = A(P)
437        %IF P >= 0 %THENSTART
438           S = A(I+1) ; X = S&RH9 ; S = S>>9
439           Y = ( I+2 ) ; Z = P
440           %IF L = ON %THENSTART
441              SELECT OUTPUT(LISTING)
442              I = -1
443     2:       %IF I # X %THENSTART
444                 PAGE ; WRITE(LISTING LINE NO,4)
445                 I = I + 1 ; -> 2
446              %FINISH
447              %IF Y # Z %THENSTART
448                 SPACES(S+2)
449                 %CYCLE  I = Y,1,Z-1
450                    K = C( I )
451                    PRINT SYMBOL(K&127) ; SPACES( K>>9 )
452                 %REPEAT
453              %FINISH
454           %FINISH
455           SELECT OUTPUT(NEW SOURCE)
456           NEWLINES(X) %IF X > 0 ; SPACES(S)
457           %IF Y # Z %THENSTART
458              %CYCLE  I = Y,1,Z-1
459                 K = C( I )
460                 PRINT SYMBOL(K&127) ; SPACES( K>>9 )
461              %REPEAT
462           %FINISH
463           NEWLINE
464           -> 1
465        %FINISH
466     %END
467     !
468     !
469     !
470     !
471     %ROUTINE  PROCESS FLAG LINE
472     %SWITCH  SW('A':'Z')
473     %INTEGER  I,K,N,QUAL,QUERY,QM
474     !
475     %ROUTINE  GET I
476        %IF QM = OFF %THENSTART
477           I = 0          ;!  TO MARK APTR >= AMAX ON RETURN
478     1:    %RETURN %IF APTR >= AMAX
479           %IF A(APTR) = ' ' %THENSTART
480              APTR = APTR+1 ; -> 1
```

```
481         %FINISH
482         I = A(APTR) ; APTR = APTR+1
483      %FINISHELSESTART
484   2:    PROMPT(':') ; READ SYMBOL(I)
485         -> 2 %IF I = ' ' %OR I&QM = NL   ;!  IGNORE TT NEWLINES
486      %FINISH
487      I = I-32 %IF 'A' <= I-32 <= 'Z'      ;!  IGNORE CASE SHIFT ON LETTERS
488   %END
489   !
490   %ROUTINE  READ NO
491      N = 0
492   1: %IF '0' <= I <= '9' %THENSTART       ;!    ** 'I' MUST BE INITIALISED **
493         N = 10*N + I-'0'
494         GET I ; -> 1
495      %FINISH
496   %END
497   !
498   %ROUTINE  SKIP
499      GET I %IF I = ','       ;!  SKIP OVER COMMA
500   %END
501   !
502   !
503   !
504      SELECT INPUT(MONITOR) ; SELECT OUTPUT(MONITOR)
505   0: QM = OFF
506   1: QUAL = OFF ; QUERY = OFF
507   2: GET I
508      -> END %IF I = 0      ;!   FROM GET I ONLY
509      %IF I = '\' %OR I = '?' %THENSTART
510         -> CONTEXT %IF QUAL!QUERY = ON
511         %IF I = '\' %THEN QUAL = ON %ELSE QUERY = ON
512         -> 2 %IF  QUERY&QM # ON
513      %FINISH
514   !
515      -> SW(I) %IF 'A' <= I <= 'Z'
516      ! TRAP ALL UNUSED SWITCH ELEMENTS
517      SW('A'):SW('B'):SW('F'):SW('H'):SW('K'):SW('Q'):SW('V'):
518   !
519   !  UNRECOGNISABLE ITEM IN FLAG LINE
520   CONTEXT: PRINT SOURCE LINE ; %PRINTTEXT '*WHAT?' ; NEWLINE
521         -> 1 %IF QM = ON
522   20:GET I
523      %IF I = '\' %OR I = '?' %OR 'A' <= I <= 'Z' %THENSTART
524         APTR = APTR-1 ; -> 0
525      %FINISH
526      -> 20 %IF I # 0
527      -> END
528   !
529   !
530   SW('I'): %IF QUERY&F = ON %THENSTART      ;!  NOT NEEDED IF F = OFF
531         %PRINTTEXT 'ISO OUTPUT(I)?'
532   3:    QM = ON ; K = I ; -> 1
533      %FINISH
534      ISO = ON!!QUAL ; -> 0
535   !
536   SW('J'): %IF QUERY&F = ON %THENSTART      ;! REDUNDANT UNLESS F = ON
537         %PRINTTEXT 'RIGHT JUSTIFY(J)?' ; -> 3
538      %FINISH
539      J = ON!!QUAL ; -> 0
540   !
```

```
541   SW('P'): %IF QUERY&W = ON %THENSTART  ;! REDUNDANT IF SINGLE PAGES(\W)
542         %PRINTTEXT 'PRIMARY OUTPUT(P)?' ; -> 3
543      %FINISH
544      F = ON!!QUAL ; -> 0
545   !
546   SW('L'): %IF S = ON %THENSTART     ;!  NOT ALLOWED UNLESS S IS ON
547         %IF QUERY&W = ON %THENSTART
548            %PRINTTEXT 'SOURCE LISTING(L)?' ; -> 3
549         %FINISH
550         L = ON!!QUAL       ;!  CHECK FOR L = ON # S AT END
551      %FINISH
552      -> 0
553   !
554   SW('S'): %IF QUERY&W = ON %THENSTART
555         %PRINTTEXT 'NEW SOURCE FILE(S)?' ; -> 3
556      %FINISH
557      S = ON!!QUAL ; -> 0
558   !
559   SW('C'): %IF QUERY&F = ON %THENSTART        ;!  PRINT CARRY-OVER WORDS?
560         %PRINTTEXT 'CARRY-OVER WORDS(C)?' ; -> 3
561      %FINISH
562      OVER = ON!!QUAL
563      -> 0
564   !
565   SW('R'): %IF QUERY = ON %THENSTART      ;! SET REL ON/OFF
566         %PRINTTEXT 'RELATIVE TABS(\R)?' ; -> 3
567      %FINISH
568      REL = ON!!QUAL
569      -> 0
570   !
571   SW('E'): %IF F # ON %THENSTART      ;! PRIMARY OUTPUT REQUESTED?
572         %IF QUERY = ON %THENSTART     ;!  NO:- IS QUERY MARKER SET?
573            %PRINTTEXT 'END-OF-PAGE LINE NOS(\E)?' ; -> 3
574         %FINISH
575         E = ON!!QUAL
576      %FINISH                          ;!  YES:- IGNORE 'E' ALTOGETHER
577      -> 0
578   !
579   !
580   SW('Y'): QUAL = OFF ; -> 4
581   SW('N'): QUAL = ON
582   4: -> CONTEXT %IF QM # ON ; -> SW(K)
583   !
584   SW('T'): %IF QUERY = ON %THENSTART
585         %PRINTTEXT 'TABS=?' ; QM = ON
586      %FINISH
587      %CYCLE K = 1,1,20     ;!  UNSET DEFAULT TAB VALUES
588         TABTAB(K) = -1
589      %REPEAT
590      GET I
591      %CYCLE  K = 1,1,20
592         READ NO; N = N-1 + TABTAB(0)
593         -> CONTEXT %IF TABTAB(K-1) > N
594         TABTAB(K) = N
595         -> 0 %IF I # ','; SKIP  ;! OVER ','
596      %REPEAT
597      PRINT SOURCE LINE %IF QM # ON
598      %PRINTTEXT '* >20 TABS' ; -> 0
599   !
600   SW('M'):  %IF QUERY = ON %THENSTART
```

```
601         %PRINTTEXT 'L.H. MARGIN(0)=?'; QM = ON
602       %FINISH
603    5: GET I; READ NO; -> 5 %IF N < 0
604       N = N - TABTAB(0)
605       %IF B2 = B1+1 %THENSTART       ;! IF CURRENT OUTPUT LINE EMPTY
606          HSKIPB = HSKIPB+N; TOTAL EXTRA B = TOTAL EXTRA B + N
607       %FINISH
608       %CYCLE  K = 0,1,20
609          -> 0 %IF TABTAB(K) = -1
610          TABTAB(K) = TABTAB(K) + N
611       %REPEAT
612       -> 0
613    !
614    SW('O'): %IF QUERY = ON %THENSTART
615          %PRINTTEXT 'INITIAL SECTION,PAGE(0,0)=?' ; QM = ON
616       %FINISH
617       GET I ; READ NO ; SECTION = N ; SKIP ; READ NO ; PAGE = N
618       -> 0
619    !
620    SW('D'): %IF QUERY = ON %THENSTART
621    6:    %PRINTTEXT 'LINES/PAGE,CHARS/LINE(72,60)=?' ; QM = ON
622       %FINISH
623       GET I ; READ NO ; LINES ON PAGE = N
624       SKIP ; READ NO ; CHARS ON LINE = N
625       %IF CHARS ON LINE = 0 %OR LINES ON PAGE = 0 %THENSTART
626          -> 6 %IF QM = ON ; -> CONTEXT
627       %FINISH
628       -> 0
629    !
630    SW('W'): %IF W = ON  %THENSTART  ;!  IF PROCESSING WHOLE FILE
631          %IF QUERY = ON %THENSTART    ;!  IF ?W
632             %PRINTTEXT 'WHOLE FILE(W)?' ; -> 3
633          %FINISH
634          W = ON!!QUAL ; WSTAT = ON
635       %FINISH
636       -> 0
637    !
638    SW('U'):  %IF QUERY = ON %THENSTART      ;! UCS, ULS RESET
639          %PRINTTEXT 'SHIFT CHARACTERS(@,_)=?' ; QM = ON
640       %FINISH
641       GET I
642       %IF I = ',' %THEN UCS = -1 %ELSE UCS = I
643       GET I ; SKIP          ;!  I = NEXT NON-COMMA
644       %IF I # ';' %THENSTART
645          ULS = I            ;!  RESET UNDER LINE MARKER
646          GET I ; -> CONTEXT %IF I # ';'    ;!  CHECK COMMAND SYNTAX
647       %FINISHELSE  ULS = -1        ;!   DISABLE UNDERLINE
648       -> 0
649    !
650    SW('X'):  %IF QUERY = ON %THENSTART          ;!  RESET DOTSP
651    7:    %PRINTTEXT 'SENTENCE GAP(3)=?' ; QM = ON
652       %FINISH
653       GET I ; READ NO ; -> 7 %IF N <= 0 ; DOTSP = N
654       -> 0
655    !
656    SW('Z'):  %IF QUERY&F = ON %THENSTART          ;!  RESET NL FACTOR
657    8:    %PRINTTEXT 'NLFACTOR(1)=?' ; QM = ON
658       %FINISH
659       GET I ; READ NO ; N = 1 %IF N <= 0 ; NL FACTOR = N
660       %IF FORMAT LINE NO # 0 %AND NLFACTOR # 1 %THENSTART
```

```
661         N = FORMAT LINE NO + VSKIPB + 1
662         %IF (N//NLFACTOR)*NLFACTOR # N %THENSTART
663            PRINT SOURCE LINE
664            %PRINTTEXT 'OUT OF STEP AT' ; WRITE(N,1) ; NEWLINE
665         %FINISH
666      %FINISH
667      -> 0
668   !
669   SW('G'): %IF QUERY = ON %THENSTART  ;! SET INTERPARA. GAP ('PGAP')
670   9:    %PRINTTEXT 'PARAGRAPH GAP(2)=?' ; QM = ON
671      %FINISH
672      GET I ; READ NO ; -> 9 %IF N <= 0
673      PGAP = N ; -> 0
674   !
675   END: L = OFF %IF L = ON # S     ;!  NO LISTING WITHOUT NEW SOURCE!
676   %END
677   !
678   !
679   !
680   !
681   %ROUTINE  FORMAT
682   %OWNINTEGER  LAST TAB = -1, TOTAL EXTRA C = 0
683   !
684   %INTEGER  LVL,OLDLVL,X,Y,SPACES,MOD,REALCH
685   %INTEGER  I
686   !
687   !
688   %OWNINTEGERARRAY  RBR(0:10) = -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1
689   %OWNINTEGERARRAY  SHIFT(0:10) = 0,0,0,0,0,0,0,0,0,0,0
690   !
691   !
692   %INTEGERFN OPPOSITE(%INTEGER  I)
693      %RESULT = ']' %IF I = '['
694      %RESULT = ')' %IF I = '('
695      %RESULT = '>' %IF I = '<'
696      %RESULT = '[' %IF I = ']'
697      %RESULT = '(' %IF I = ')'
698      %RESULT = '<' %IF I = '>'
699      %RESULT = -1
700   %END
701   !
702   !
703   !
704   %ROUTINE  CLOSE B
705   ! ***DESIGN FAILING***
706   !  THIS ROUTINE TURNS THE PAGE WHEN THE FIRST LINE OF THIS NEXT
707   !  PAGE IS COMPLETE AND READY TO PRINT RATHER THAN WHEN THE LAST
708   !  OF THE CURRENT PAGE IS READY.  THIS GIVES RISE TO THE PROBLEMS
709   !  WITH FORMATTING CONSECUTIVE SINGLE PAGES AND PRINTING THE
710   !  END-OF-PAGE LINES CORRECTLY
711   !
712   !
713   %ROUTINE  NEWPAGE
714   %INTEGER  I,P
715   !
716   %ROUTINE  PUT(%INTEGER  I)
717      B2 = B2+1; %MONITOR 32 %IF B2 = BCL  ;! OUTPUT BUFFER OVERFLOW?
718      B(B2) = I
719   %END
720   !
```

```
721   !
722   %ROUTINE  STOPCODE
723      PUT(20)
724   %END
725   !
726   !
727   %INTEGERFN  UPPERCASE(%INTEGER  I)
728      %RESULT = (I&127)+UC          ;!  SET UPPERCASE SHIFT BIT
729   %END
730   !
731   !
732   %ROUTINE  WRITE ARABIC(%INTEGER N)
733   %INTEGER  I,J,K
734      J = 1
735      %CYCLE  I= 1,1,5
736         -> 1 %IF J > N ; J = 10*J
737      %REPEAT
738      %PRINTTEXT 'PAGE NUMBER? ' ; WRITE(N,1) ; %STOP
739   !
740   1: J = J // 10
741   2: %IF J # 0 %THENSTART
742         K = '0'
743   3:    %IF N >= J %THENSTART
744            N = N-J ; K = K+1 ; -> 3
745         %FINISH
746         PUT(K) ; J = J//10 ; -> 2
747      %FINISH
748   %END
749   !
750   !
751   %ROUTINE  WRITE ROMAN(%INTEGER  N)
752   %INTEGER  J,K
753   %OWNINTEGERARRAY  S(0:3) = 'I','X','C','M'
754   %OWNINTEGERARRAY  T(0:2) = 'V','L','D'
755      J = 1000 ; K = 3
756   1: %IF N >= J %THENSTART
757         PUT( UPPERCASE( S(K) ) ) ; N = N - J ; -> 1
758      %FINISH
759      K = K - 1 ; %RETURN %IF K < 0
760      J = J // 10
761      %IF N >= 9*J %THENSTART
762         PUT( UPPERCASE( S(K) ) ) ; PUT( UPPERCASE( S(K+1) ) )
763         N = N - 9*J
764      %FINISHELSESTART
765         %IF N >= 5*J %THENSTART
766            PUT( UPPERCASE( T(K) ) ) ; N = N - 5*J
767         %FINISHELSESTART
768            %IF N >= 4*J %THENSTART
769              PUT( UPPERCASE( S(K) ) ) ; PUT( UPPERCASE( T(K) ) )
770              N = N - 4*J
771            %FINISH
772         %FINISH
773      %FINISH
774      -> 1
775   %END
776
777   !
778   !
779      %IF E = ON %THENSTART      ;!  END-OF-PAGE LINE NOS.
780         PRINT SOURCE LINE
```

```
781         WRITE(SECTION,1) ; WRITE(-PAGE,1)
782         %PRINTTEXT ' ENDS' ; NEWLINE
783       %FINISH
784
785     !
786     !
787     !
788       P = LINES ON PAGE - FORMAT LINE NO
789       %IF P < 0 %THENSTART
790         PRINT SOURCE LINE
791         %PRINTTEXT '***OVERFLOW  LINES ON PAGE,FORMAT LINE NO,P ='
792         WRITE(LINES ON PAGE,1) ; WRITE(FORMAT LINE NO,1) ; WRITE(P,1)
793         %STOP
794       %FINISH
795     !
796     !   B1   AND    B2    ARE FREE AT THIS TIME.  USE THEM TO SAVE SPACE
797     !
798       %IF BPTR < 0 %THEN  B1 = 0  %ELSE  B1 = B(BPTR)+1
799       B2 = B1
800     ! PRINT CARRYOVER WORD IF (1) IT IS REQUESTED, (2) THIS IS NOT AN
801     !   EXPLICIT PAGE THROW(VSKIPB#-1) AND (3) NOT A NEW PARAGRAPH
802       %IF OVER=ON %AND VSKIPB >= 0 %AND HSKIPB=TABTAB(INDENT) %THENSTART
803         PUT( HSKIPB*SP + P ) ; P = NLFACTOR-1
804         PUT('/')
805         %CYCLE  I = BPTR+2,1,B(BPTR)-1     ;!  NEXT REAL OUTPUT LINE
806           -> 1 %IF B(I)&127 = ' ' ; PUT( B(I) )
807         %REPEAT
808     1:    B2 = B2+1          ;!   MOVE TO NEXT BLANK CELL
809         B(B1) = B2 ; B1 = B2           ;! SET UP NEXT LINE
810       %FINISH %ELSE %START
811         P = P + NLFACTOR    ;!  INSERT BLANK LINE OTHERWISE
812       %FINISH
813     !
814       %IF PAGE > 0 %THENSTART          ;!   A PAGE NO. IS REQUIRED
815         PUT( ((CHARS ON LINE//2)-2)*SP + P )
816         %IF SECTION > 0 %THENSTART
817           WRITE ROMAN(SECTION) ; PUT('-')
818         %FINISH
819         WRITE ARABIC(PAGE)
820       %FINISHELSE  PUT(P)       ;!  NULL LINE SO NO 'HSKIP' JUST 'VSKIP'
821     !
822       STOP CODE          ;!   PRINT ROUTINES  INSERT RUNOUT
823       PUT(-1) ; B(B1) = B2
824       %IF BPTR < 0 %THEN B1 = 0 %ELSE B1 = B(BPTR)+1
825       PRINT(B,B1) %IF F # OFF
826       PAGE = PAGE+1 %IF PAGE > 0          ;! MUST BE AFTER 'PRINT'!
827       FORMAT LINE NO = 0 ; VSKIPB = 0
828     !
829     ! REQUEST NEXT PAGE NO, IF IN SINGLE PAGE MODE AND THE PAGE JUST
830     ! COMPLETED WAS THE REQUESTED ONE.
831       WSTAT=ON %IF W#ON %AND SECTION=REQD SECTION %AND PAGE=REQD PAGE + 1
832     %END      ;!     NEWPAGE
833     !
834     !
835     !
836     !
837       %IF B2 = B1+1 %THENSTART      ;!  NULL LINE
838         B(B1) = -1
839         NEWPAGE %IF VSKIPB < 0
840       %FINISHELSESTART
```

```
841   1:     %IF B(B2)&127 = ' ' %THENSTART
842             B2 = B2-1 ; -> 1
843          %FINISH
844          BPTR = B1 ; B(BPTR) = B2+1 ; B(B2+1) = -1
845   !
846          %IF FORMAT LINE NO + VSKIPB >= LINES ON PAGE %THEN NEWPAGE %C
847          %ELSE  FORMAT LINE NO = FORMAT LINE NO + VSKIPB
848   !
849          VSKIPB = 0 %IF FORMAT LINE NO = 0
850          B(BPTR+1) = HSKIPB*SP + VSKIPB ; PRINT(B,BPTR) %IF F = ON
851          FORMAT LINE NO = FORMAT LINE NO + 1
852   !
853          %IF COPY # ON %THEN VSKIPB = NLFACTOR-1 %ELSE VSKIPB = 0
854          HSKIPB = TABTAB(INDENT) ; TOTAL EXTRA B = HSKIPB
855       %FINISH
856       B1 = 0 ; B2 = 1 ; B(0) = -1 ; BPTR = -1
857       LAST TAB = -1
858   %END      ;!  CLOSE B
859   !
860   !
861   %ROUTINE  CLOSE C
862   %INTEGER  I,K,X
863   !
864      %IF C2 = C1+1 %THENSTART
865         C(C1) = -1 ; VSKIPC = VSKIPC+1
866      %FINISHELSESTART
867   1:    %IF C(C2)&127 = ' ' %THENSTART
868            C2 = C2-1 ; -> 1
869         %FINISH
870         %IF OLDLVL <= 0 %THEN X = 0 %ELSESTART
871            X = OLDLVL
872   ! DETECT CASE WHERE BRACKETTED ATOM JUST MISSED BEING INCLUDED IN LINE
873   ! AND SO EVEN THOUGH 'LVL' IS NON-ZERO, THERE IS NO NEED TO INSERT
874   ! EXTRA BRACKETS
875            %IF X > 0 %THENSTART
876               %CYCLE  I = X,-1,1
877                  C2 = C2+1 ; C(C2) = RBR(I)
878               %REPEAT
879            %FINISH
880         %FINISH
881         CPTR = C1 ; C(CPTR) = C2+1 ; C(C2+1) = -1
882   !
883         C(CPTR+1) = HSKIPC*SP + VSKIPC ; LIST(C,CPTR) %IF S = ON
884         VSKIPC = 0 ; HSKIPC = (2*INDENT)&(\COPY) ; TOTAL EXTRA C = 0
885         C1 = 0 ; C2 = 1 ; C(0) = -1 ; CPTR = -1
886         %IF X > 0 %THENSTART     ;!  INSERT SHIFT AND UNDERLINE MARKERS
887            K = 0
888            %CYCLE  I = 1,1,X
889               K = K!!SHIFT(I)
890               %IF K # 0 %THENSTART
891                  %IF K = UL %THEN K = ULS %ELSE K = UCS
892                  C2 = C2+1 ; C(C2) = K
893               %FINISH
894               C2 = C2+1 ; C(C2) = OPPOSITE( RBR(I) )
895               K = SHIFT(I)
896            %REPEAT
897         %FINISH
898      %FINISH
899   %END
900   !
```

```
!
%ROUTINE  COPY TO C(%INTEGER  FROM,TO)
%INTEGER  I
   CLOSE C %IF C2 # C1+1
   %RETURN %IF FROM > TO
   %CYCLE  I = FROM,1,TO
      C2 = C2+1 ; C(C2) = A(I)
   %REPEAT
   VSKIPC = EMPTY LINES %IF COPY = ON  ;! 'EMPTY LINES' SET BY READ IN
   CLOSE C
%END
!
!
!
%ROUTINE  CHECK CSYM
   %RETURN %IF CSYM = 'D' %OR CSYM = 'E' %OR CSYM = 'V'
   %RETURN %IF CSYM = 'I' %OR CSYM = 'N' %OR CSYM = 'P'
   %RETURN %IF CSYM = 'S' %OR CSYM = 'T' %OR CSYM = 'L'
   %RETURN %IF CSYM = 'F'
!
   PRINT SOURCE LINE
   %PRINTTEXT '*NOT A CONTROL SYMBOL ' ; PRINT SYMBOL(CSYM) ; NEWLINE
   CSYM = -1
%END
!
!
%ROUTINE  DO SPECIAL1
%SWITCH  SW('A':'Z')
%INTEGER  I,K,X
!
%ROUTINE  READN(%INTEGERNAME  N)
   N = 0
1: %IF APTR <= AMAX %AND '0' <= A(APTR) <= '9' %THENSTART
      N = 10*N + A(APTR)-'0' ; APTR = APTR+1
      -> 1
   %FINISH
%END
!
!
   CHECK CSYM
   %IF CSYM > 0 %THEN -> SW(CSYM) %ELSE %RETURN
!
SW('D'):  CLOSE B %IF B1+1 # B2     ;!  DIAGRAM
   READN(SPEC COPY SIZE) ; SPEC COPY SIZE = 1 %IF SPEC COPY SIZE = 0
   ACTUAL COPY SIZE = 0
   %IF FORMAT LINE NO + VSKIPB + SPEC COPY SIZE >   %C
       LINES ON PAGE %THENSTART
      VSKIPB = -1 ; CLOSE B  ;! OFFICIAL WAY TO GET A NEWPAGE
   %FINISH
   %IF SPEC COPY SIZE > LINES ON PAGE %THENSTART
      PRINT SOURCE LINE
      %PRINTTEXT '*TOO BIG FOR PAGE SIZE ='
      WRITE(LINES ON PAGE,1) ; NEWLINE
   %FINISH
   COPY START LINE = INPUT LINE NO
   COPY = ON
   -> END
!
SW('E'): CLOSE B                    ;!   END OF INPUT FILE
   %IF FORMAT LINE NO # 0 %THENSTART    ;!   IF FIRST LINE IS FILLED....
```

```
961         VSKIPB = -1 ; CLOSE B                ;! ....THROW ANOTHER PAGE
962       %FINISH
963       RUN FLAG = OFF                  ;! STOP AT END OF CURRENT CYCLE
964       SELECT OUTPUT(MONITOR) ; NEWLINES(2)
965       %PRINTTEXT '<>E AT'
966       WRITE(SECTION,1) ; WRITE(-(PAGE-1),1) ; NEWLINE
967       -> END
968   !
969   SW('F'): PROCESS FLAG LINE ; -> END       ;!  RESET OPTIONS
970   !
971   SW('I'): READN(I) ; I = 1 %IF I = 0 %AND REL # ON ;!  SET INDENT MARKER
972       -> 10 %IF I > 20          ;!  CAN'T BE NEGATIVE
973       -> TAB NOT SET %IF TABTAB(I) < 0
974       INDENT = I
975       -> END
976   !
977   SW('L'): READN(I)                        ;!  NEWLINES
978       I = 1 %IF I = 0
979       CLOSE B
980       %IF FORMAT LINE NO # 0 %THENSTART
981          I = I-1 %IF VSKIPB = NLFACTOR-1
982          VSKIPB = VSKIPB + I*NLFACTOR
983       %FINISH
984       ! SUPPRESS INDENT ON NEXT LINE
985       HSKIPB = TABTAB(0); TOTAL EXTRA B = HSKIPB
986       -> END %IF INDENT < 2 %OR REL # ON ;! STRAIGHT NEW LINE IF REL # ON
987       I = -1 ; -> 1             ;! OTHERWISE TAB TO N-1 TH. STOP
988   !
989   SW('N'):  READN(I) ; I = 1 %IF I = 0  ;!  NEWPAGE
990       INDENT = 0 %IF REL # ON
991       CLOSE B
992       I = I-1 %IF FORMAT LINE NO = 0  ;! ALREADY AT BOTTOM
993       %IF I > 0 %THENSTART
994          %CYCLE  I = 1,1,I
995             VSKIPB = -1 ; CLOSE B     ;!  DO THIS NUMBER OF NEWPAGES
996          %REPEAT
997       %FINISH
998       -> END
999   !
1000  SW('P'):  INDENT = 0 %IF REL # ON          ;!  NEW PARAGRAPH
1001      CLOSE B %IF B2 # B1+1     ;!   NEW LINE IF NOT ALREADY SO
1002  ! NEW PARAGRAPH MAY BE MULTIPLE-SPACED:- THROW 'PGAP' LOGICAL NEWLINES
1003      %IF FORMAT LINE NO # 0 %THENSTART
1004         I = PGAP; I = I-1 %IF VSKIPB >= 0
1005         VSKIPB = VSKIPB + I*NLFACTOR
1006      %FINISH
1007      I=0 ; I=NLFACTOR %IF OVER # ON  ;! TERMINATE EARLY IF NO CARRY OVER
1008      %IF FORMAT LINE NO + VSKIPB + I >= LINES ON PAGE %THENSTART
1009         VSKIPB = -1 ; CLOSE B                  ;!  FORCE A NEW PAGE
1010      %FINISH
1011      I = 1 ; -> 1     ;!  GO DO A TAB 1
1012  !
1013  SW('S'):  INDENT = 0 %IF REL # ON ; CLOSE B       ;!   NEW SECTION
1014      VSKIPB = -1 ; CLOSE B       ;!    FORCE NEW PAGE
1015      %IF SECTION > 0 %THENSTART
1016         SECTION = SECTION+1 ; PAGE = 1
1017      %FINISH
1018      -> END
1019  !
1020  SW('T'):      READN(I)            ;!     TAB
```

```
1021    1: K = B2-B1-1 + TOTAL EXTRA B      ;!  TOTAL CHARS ON LINE SO FAR
1022       %IF I = 0 %THENSTART   ;!  I # 0 WHENEVER CSYM = 'L'
1023          %CYCLE  I = 1,1,20
1024             X = TABTAB(I)
1025             -> TAB NOT SET %IF X < 0
1026             -> 3 %IF X > K
1027          %REPEAT
1028          -> TAB NOT SET
1029       %FINISHELSESTART
1030          I = I+INDENT %IF REL = ON ;! TAB RELATIVE TO CURRENT INDENTATION?
1031          %UNLESS 1 <= I <= 20 %THENSTART
1032    10:      PRINT SOURCE LINE
1033             %PRINTTEXT '*TAB NO. OUT OF RANGE' ; WRITE(I,1) ; NEWLINE
1034             -> END
1035          %FINISH
1036       %FINISH
1037    2: X = TABTAB(I)
1038       %IF X < 0 %THENSTART
1039    TAB NOT SET:  PRINT SOURCE LINE
1040          %PRINTTEXT '*TAB' ; WRITE(I,1) ; %PRINTTEXT ' NOT SET' ; NEWLINE
1041          -> END
1042       %FINISH
1043       %IF K > X %THENSTART    ;!  IF ALREADY PAST THE TAB POSN.
1044          PRINT SOURCE LINE
1045          %PRINTTEXT '*TAB BACKWARDS?' ; NEWLINE
1046          -> END
1047       %FINISH
1048    3: LAST TAB = B2+1
1049       %IF B2 = (B1+1) %THENSTART   ;!  %IF INITIAL INDENT
1050          HSKIPB = X ; TOTAL EXTRA B = X
1051       %FINISHELSESTART
1052          B(B2) = B(B2) + (X-K)*SP ; TOTAL EXTRA B = TOTAL EXTRA B + (X-K)
1053       %FINISH
1054       -> END
1055    !
1056    SW('V'): READN(I)  ;! VERIFY THAT I LOGICAL LINES REMAIN ON CURRENT PAGE
1057       I = I-1 ; I = 0 %IF I < 0  ;!   REQUESTED I  INCLUDES CURRENT LINE
1058       CLOSE B %IF B2 # B1+1
1059       %IF FORMAT LINE NO + VSKIPB + I*NLFACTOR+1 > LINES ON PAGE %THENSTART
1060          VSKIPB = -1 ; CLOSE B          ;!  FORCE A NEW PAGE
1061       %FINISH
1062       -> END
1063    !
1064    END: LAST CSYM = CSYM
1065    %END
1066    !
1067    !
1068    !
1069    %ROUTINE  DO SPECIAL2
1070    %SWITCH  SW('A':'Z')
1071    !
1072    %ROUTINE  COPYN
1073    1: %IF '0' <= A(Y) <= '9' %THENSTART
1074          C2 = C2+1 ; C(C2) = A(Y)
1075          Y = Y+1 ; -> 1
1076       %FINISHELSESTART
1077          C2 = C2+1 ; C(C2) = ' '
1078       %FINISH
1079    %END
1080    !
```

```
1081    !
1082    %ROUTINE  CSMARK        ;!   OUTPUTS '<>'[CSYM]
1083       CLOSE C %IF C2-C1-1 + Y-APTR+1 + TOTAL EXTRA C    %C
1084            + LVL + 2*INDENT + 4 > 72
1085       C2 = C2+1 ; C(C2) = '<' ; C2 = C2+1 ; C(C2) = '>'
1086       C2 = C2+1 ; C(C2) = CSYM
1087    %END
1088    !
1089    !
1090       CHECK CSYM
1091       %IF CSYM > 0 %THEN -> SW(CSYM) %ELSE %RETURN
1092    !
1093    SW('D'):SW('N'):SW('V'): CLOSE C %IF C2 # C1+1
1094       CSMARK ; COPYN ; CLOSE C
1095       VSKIPC = 5 %IF CSYM = 'N'     ;!   FOR  <>N
1096       -> END
1097    !
1098    SW('F'):  COPY TO C(Y-3,AMAX) ; -> END     ;!  FLAG LINE
1099    !
1100    SW('L'): CSMARK ; COPYN ; CLOSE C      ;!   NEW LINE(S) [N]
1101          -> END
1102    !
1103    SW('P'): CLOSE C %IF C2 # C1+1           ;!  PARAGRAPH
1104          CSMARK        ;! AT START OF NEXT LINE
1105          C2 = C2+1 ; C(C2) = ' '    ;!  FOLLOW <>P WITH A SPACE
1106          VSKIPC = 1 %IF VSKIPC = 0       ;!    INSERT BLANK LINE MAYBE
1107          -> END
1108    !
1109    SW('S'):SW('E'): CLOSE C ;!      NEW SECTION,END OF INPUT
1110          CSMARK ; CLOSE C
1111          VSKIPC = 5 %IF CSYM # 'E'   ;!  %IF NOT <>E
1112          -> END
1113    !
1114    SW('T'):SW('I'): CSMARK ; COPYN ; -> END    ;!  TAB [N], INDENT [N]
1115    !
1116    END: %END      ;!  DO SPECIAL2
1117    !
1118    !
1119    !
1120    !
1121    !
1122    !
1123    !   ** IF SPEED IS CRITICAL, REPLACING THE PARAMETERS OF
1124    !   'DEFINE ATOM' WITH DIRECT REFERENCES TO THE APPROPRIATE VARIABLES
1125    !   SHOULD BE A FIRST EASY STEP.  ALTERNATIVELY, AS THERE IS ONLY ONE
1126    !   CALL ON THE ROUTINE, IT COULD BE DEFINED AS A MACRO AND INSERTED
1127    !   IN-LINE.  **
1128    !
1129    %ROUTINE  DEFINE ATOM(%INTEGERNAME  FROM,TO,SPACES,TYPE)
1130    ! PICKS OFF NEXT ITEM IN INPUT BUFFER(A)
1131    ! AN ITEM IS EITHER A CONTROL SEQUENCE( <>? ) OR
1132    ! NON-SPACE(S) FOLLOWED BY SPACE(S)
1133    !
1134    ! TYPE IS SET TO:
1135    ! < 0 : END OF LINE
1136    ! = 0 : NON-SPACE ..... SPACE STRING
1137    ! > 0 : CONTROL SYMBOL, TYPE = SYMBOL
1138    !
1139    %INTEGER  K
1140       TYPE = -1
```

```
1141    1: %IF FROM < AMAX %THENSTART
1142          K = A(FROM)
1143          %IF K = '<' %AND A(FROM+1) = '>' %THENSTART
1144             TYPE = A(FROM+2) ; FROM = FROM+3 ; TO = FROM ; %RETURN
1145          %FINISH
1146          %IF K # ' ' %THENSTART
1147             TYPE = 0 ; TO = FROM
1148    2:       TO = TO+1 ; K = A(TO)
1149             -> 3 %IF K = '<' %AND A(TO+1) = '>'
1150             -> 2 %IF K # ' '
1151    3:       TO = TO-1           ;!   POINT AT LAST CHAR NOT SPACE
1152             SPACES = 1
1153             SPACES = DOTSP %IF J # ON %AND A(TO) = '.'
1154             %RETURN
1155          %FINISH
1156          FROM = FROM + 1 ; -> 1
1157       %FINISH
1158    %END
1159    !
1160    !
1161    !
1162    %ROUTINE  STRETCH
1163    %INTEGER  I,J,N,P
1164    %INTEGER  X,Y,Z
1165    !
1166       LAST TAB = (B1+2) %IF LAST TAB < 0
1167       -> FAIL %IF LAST TAB = B2
1168    !
1169    0: P = B(B2)
1170       %IF P&127 = ' ' %THENSTART
1171          B2 = B2-1
1172          TOTAL EXTRA B = TOTAL EXTRA B - (P>>9)
1173          -> 0
1174       %FINISH
1175    !
1176
1177       N = CHARS ON LINE - ( B2-B1-1 + TOTAL EXTRA B )
1178       %RETURN %IF N = 0
1179       -> FAIL %IF N < 0 %OR LAST TAB >= B2-1
1180       %IF N<=2 %OR ((FORMAT LINE NO+VSKIPB+1)//NL FACTOR)&1 # 0 %THENSTART
1181          X = B2-1; Y = -1; Z = LAST TAB
1182       %FINISHELSESTART
1183          X = LAST TAB; Y = +1; Z = B2-1
1184       %FINISH
1185    !
1186    1: J = N          ;!   SET TRAP FOR LOOPS(LINES CONTAINING NO SPACES)
1187       %CYCLE  I = B2-1,-1,LAST TAB
1188          P = B(I)
1189          %IF B(I+1)&127 = ' ' %AND (P&127 = '.' %OR P&127 = '?' %C
1190             %OR P&127 = '!') %THENSTART
1191             B(I) = P+SP          ;!  INSERT EXTRA SPACES AFTER '.'
1192             N = N-1 ; %RETURN %IF N = 0
1193          %FINISH
1194       %REPEAT
1195       %IF N-2 <= 0 %THENSTART    ;! IF TWO OR LESS SPACES TO DISPOSE OF
1196          %CYCLE  I = B2-1,-1,LAST TAB
1197             P = B(I)&127
1198             %IF B(I+1)&127=' ' %AND (P='.' %OR P=';'%C
1199                            %OR P=':' %OR P='?' %OR P='!') %THENSTART
1200                B(I+1) = B(I+1) + SP
```

```
1201                N = N-1; %RETURN %IF N = 0
1202             %FINISH
1203          %REPEAT
1204          -> 1 %IF N # J     ;!  ANY PROGRESS SO FAR? -  YES ->
1205       %FINISH
1206       %CYCLE  I = X,Y,Z
1207          P = B(I)
1208          %IF P&127 = ' ' %THENSTART
1209             B(I) = P+SP
1210             N = N-1 ; %RETURN %IF N = 0
1211          %FINISH
1212       %REPEAT
1213       -> 1 %IF N # J    ;! .... IF WE'VE GOT RID OF SOME SPACES ON THIS PASS
1214    !
1215    FAIL:  PRINT SOURCE LINE
1216       %PRINTTEXT '*CAN''T STRETCH IT' ; NEWLINE
1217    !
1218    %END
1219    !
1220    !
1221    !
1222    !
1223    !
1224    !
1225    !
1226    !
1227       Y = 2 ; LVL = 0 ; MOD = 0
1228       %IF COPY = ON %THENSTART
1229          %IF A(Y) = '<' %AND A(Y+1) = '>' %THENSTART
1230             %IF ACTUAL COPY SIZE # SPEC COPY SIZE %THENSTART
1231               SELECT OUTPUT(MONITOR)
1232                %PRINTTEXT '*ACTUAL SIZE OF DIAGRAM AT LINE'
1233                WRITE(COPY START LINE,1) ; %PRINTTEXT ' IS'
1234                WRITE(ACTUAL COPY SIZE,1) ; NEWLINE
1235             %FINISH
1236             COPY = OFF
1237             COPY TO C(2,Y+1)        ;! INSERT '<>' ALONE AS <>D TERMINATOR
1238             Y = Y+2
1239             %RETURN %IF Y = AMAX      ;!  '<>'  ALONE IN INPUT LINE
1240             -> 0                  ;!  ....OTHERWISE, PROCESS REST OF LINE
1241          %FINISH
1242          ACTUAL COPY SIZE = ACTUAL COPY SIZE + 1
1243          COPY TO C(2,AMAX) %IF S = ON          ;!  IF LISTING IS REQUIRED
1244          SPACES = 1 ; APTR = Y ; Y = AMAX   ;! 'ATOM' IS WHOLE LINE ...
1245          -> 1                                ;! ... GO COPY IT
1246       %FINISH
1247    !
1248    0: APTR = Y ; REALCH = OFF    ;!  SET TO 'ON' IF ATOM PRINTS
1249       DEFINE ATOM(APTR,Y,SPACES,CSYM) ; OLDLVL = LVL
1250       -> 4 %IF CSYM < 0      ;! END OF INPUT LINE
1251       %IF CSYM > 0 %THEN DO SPECIAL1 %ELSESTART
1252          %IF B2-B1-1 + Y-APTR+1 + TOTAL EXTRA B > CHARS ON LINE %THENSTART
1253             !    *** DESIGN FAILING ***
1254             ! SHIFT CHARACTERS ARE COUNTED IN LENGTH OF INPUT ATOM
1255             ! AND MAY ERRONEOUSLY PREVENT IT BEING INCLUDED IN OUTPUT
1256             ! LINE. E.G. _[IS] HAS LENGTH 5, NOT 2
1257             STRETCH %IF J&F = ON                ;! RIGHT JUSTIFY ?
1258             CLOSE B
1259          %FINISH
1260    1:    %CYCLE  X = APTR,1,Y
```

```
1261            I = A(X)
1262            %IF I = UCS %OR I = ULS %THENSTART  ;!INVERT I IF COPY=ON
1263               MOD = UC ; MOD = UL %IF I = ULS
1264               I = A(X+1)
1265               %IF I = '(' %OR I = '<' %OR I = '{' %THENSTART
1266                  MOD = MOD!SHIFT(LVL) ; LVL = LVL+1
1267                  RBR(LVL) = OPPOSITE(I) ; SHIFT(LVL) = MOD
1268                  MOD = 0 ; X = X+1
1269               %FINISH
1270               -> 2
1271            %FINISH
1272            %IF I = RBR(LVL) %THENSTART
1273               %IF LVL = 0 %THENSTART
1274                  PRINT SOURCE LINE
1275                  %PRINTTEXT '*TOO MANY R.H. BRACKETS' ; NEWLINE
1276               %FINISHELSESTART
1277                  LVL = LVL-1
1278               %FINISH
1279               -> 2
1280            %FINISH
1281            B2 = (B2+1) ; B(B2) = ( I&127 ) ! MOD ! SHIFT(LVL)
1282            MOD = 0 ; REALCH = ON  ;! WE'VE GOT A PRINTING CHAR
1283    2:   %REPEAT
1284         %IF REALCH = ON %THENSTART
1285            B2 = (B2+1) ; B(B2) = ((SPACES-1)*SP + ' ') ! SHIFT(LVL)
1286            TOTAL EXTRA B = TOTAL EXTRA B + SPACES-1
1287         %FINISH
1288      %FINISH
1289    !
1290      -> 3 %IF COPY = ON %AND ACTUAL COPY SIZE # 0   ;! NOT ON <>D LINE
1291    !
1292      %IF S = ON %THENSTART              ;! NEW SOURCE FILE
1293         %IF CSYM > 0 %THEN DO SPECIAL2 %ELSESTART
1294            CLOSE C %IF C2-C1-1 + Y-APTR+1 + TOTAL EXTRA C %C
1295                + LVL + 2*INDENT > 72
1296            %CYCLE X = APTR,1,Y
1297               C2 = (C2+1) ; C(C2) = A(X)
1298            %REPEAT
1299            C2 = (C2+1) ; C(C2) = (SPACES-1)*SP + ' '
1300            TOTAL EXTRA C = TOTAL EXTRA C + SPACES - 1
1301         %FINISH
1302      %FINISH
1303      %IF CSYM = 0 %THEN Y = Y+1 %ELSE Y = APTR  ;! CONTROL SYMBOL?
1304      -> 0
1305    3: VSKIPB = VSKIPB + EMPTY LINES
1306      HSKIPB = TABTAB(0) %IF REL # ON  ;! INDENT SUPPRESSED IF NON-RELATIVE
1307      CLOSE B                    ;!   COPY THE LINE
1308    4: %IF LVL # 0 %THENSTART
1309         PRINT SOURCE LINE
1310         %PRINTTEXT '*TOO FEW R.H. BRACKETS' ; NEWLINE
1311         LVL = 0
1312      %FINISH
1313    %END    ;!  FORMAT
1314    !
1315    !
1316    !
1317    !
1318    !
1319    %ROUTINE  GET(%INTEGERNAME  N)
1320       SELECT INPUT(MONITOR) ; N = -1
```

```
1321    1: PROMPT(':')
1322       %IF '0' <= NEXT SYMBOL <= '9' %THENSTART
1323          N = 0
1324    2:    N = 10*N + NEXT SYMBOL-'0' ; SKIP SYMBOL
1325          -> 2 %IF '0' <= NEXT SYMBOL <= '9'
1326       %FINISHELSESTART
1327          %IF NEXT SYMBOL = '*' %THENSTART    ;! N = -1 FROM ABOVE
1328             SKIP SYMBOL ; %RETURN
1329          %FINISH
1330          SKIP SYMBOL ; -> 1    ;! CONTINUE SEARCH FOR FIRST DIGIT
1331       %FINISH
1332    %END
1333    !
1334    !
1335    !
1336    ! ******** START OF PROGRAM PROPER ********
1337    !
1338    0: SELECT OUTPUT(MONITOR) ; NEWLINE
1339       %PRINTTEXT 'FORMAT   JULY ''72' ; NEWLINE ; NEWLINE
1340    !
1341    !
1342    1: INITIALISE
1343    2: READ INPUT LINE
1344       %IF W = ON %THENSTART
1345          FORMAT
1346          -> 2 %IF RUN FLAG = ON
1347       %FINISHELSESTART
1348          %IF WSTAT = ON %THENSTART
1349    3:       WSTAT = OFF
1350             SELECT OUTPUT(MONITOR) ; %PRINTTEXT 'REQD SECTION,PAGE=?'
1351             GET(REQD SECTION) ; -> END %IF REQD SECTION < 0
1352             GET(REQD PAGE) ; -> 3 %IF REQD SECTION = 0 = REQD PAGE
1353             %IF SECTION > REQD SECTION %OR (SECTION = REQD SECTION %C
1354                %AND PAGE >= REQD PAGE %AND PAGE # 0) %THENSTART
1355    4:          SELECT INPUT(INPUT) ; CLOSE INPUT ;!  REWIND INPUT FILE
1356                SELECT OUTPUT(MONITOR)
1357                %IF RUN FLAG # ON %THENSTART
1358                   %PRINTTEXT '*IMPOSSIBLE PAGE/SECTION*'
1359                   WSTAT = ON              ;!  SET UP TO ASK FOR PAGE/SECTION
1360                %FINISH
1361                NEWLINE ; %PRINTTEXT '*REWOUND*' ; NEWLINE
1362                -> 1
1363             %FINISH
1364             F = ON %IF SECTION = REQD SECTION %AND PAGE = REQD PAGE
1365          %FINISH
1366          FORMAT ; -> 4 %IF RUN FLAG # ON    ;!    REWIND AFTER <>E
1367          -> 2
1368       %FINISH
1369    !
1370    END:  %ENDOFPROGRAM
```

```
%BEGIN          ;!          UCCONV    APRIL '72
%INTEGERARRAY  A(0:100)
%ROUTINESPEC READ LINE; %ROUTINESPEC  TRANSLATE
!
    SELECT OUTPUT(0)
    NEWLINE; %PRINTTEXT 'UCCONV      APRIL ''72'; NEWLINE
    SELECT OUTPUT(1)
!
    %FAULT9 -> 2
1: READ LINE; TRANSLATE; -> 1
2: A(A(0)) = NL %AND TRANSLATE %IF A(0) # 0
!
%ROUTINE  READ LINE
    A(0) = 0
    %UNTIL A(A(0)) = NL %DO:
        A(0) = A(0) + 1
        READ SYMBOL( A(A(0)) )
    %REPEAT
%END
!
%ROUTINE  TRANSLATE
%OWNINTEGER  ON = -1, OFF = 0
%INTEGER  I,S,UC
!
    %PREDICATE  UCL(%INTEGER I)
        %TRUE %IF 'A' <= I <= 'Z'
        %FALSE
    %END
    !
    %PREDICATE  LCL(%INTEGER  I)
        %TRUE %IF UCL(I-32)
        %FALSE
    %END
!
    UC = OFF; S = 0
    %IF A(0) >= 4 %AND A(1) = ' ' = A(2) %AND A(3) = ' ' %THEN:
        %PRINTTEXT ' <>P'; SPACE %IF A(4) # ' '; I = 3
    %ELSE I = 0
!
    %WHILE A(I) # NL %DO:
        I = I+1
        %PRINTTEXT ']' %AND UC = OFF %IF UC = ON %AND ( A(I) = NL %C
            %OR LCL(A(I)) )
        %IF A(I) = ' ' %THEN S = S+1 %AND -> 1 %ELSE SPACES(S) %AND S = 0
        %IF A(I) = '<' %AND A(I+1) = '>' %AND ( UCL(A(I+2)) %OR %C
            LCL( A(I+2)) ) %THEN:
            %PRINTTEXT '<>'; PRINT SYMBOL(A(I+2)); I = I+2
        %ELSE:
            %IF UCL(A(I)) %AND UC = OFF %THEN:
                %PRINTTEXT '@'
                %PRINTTEXT '[' %AND UC = ON %IF UCL( A(I+1) )
            %FINISH
            PRINT SYMBOL(A(I))
        %FINISH
1: %REPEAT
%END
!
%ENDOFPROGRAM
```

```
%BEGIN          ;!      FMTIMP MAY '72
%INTEGERARRAY  A,B(0:200)
%INTEGER  LINE NO  ; LINE NO = 0
%INTEGER  LINES ON PAGE, CHARS ON LINE, DIAGRAM SIZE


%ROUTINE  READ LINE
%INTEGER  S
   A(0) = 0
   %UNTIL S = NL %DO:
      READ SYMBOL(S)
      A(0) = A(0) + 1; A(A(0)) = S
   %REPEAT
%END


%ROUTINE  WRITE(%INTEGER  N,M)
%OWNINTEGERARRAY  TENTO(0:6)= 1,10,100,1000,10000,100000
%INTEGER  I,K,NEG,MAX
   NEG = 0
   N = -N %AND NEG = 1 %IF N < 0
   MAX = 0
   MAX = MAX+1 %WHILE TENTO(MAX) < N
   MAX = MAX-1 %IF TENTO(MAX) # N
   M = M-(MAX+1+NEG); M = 0 %IF M < 0
   SPACES(M); PRINT SYMBOL('-') %IF NEG # 0
   M = N
   %FOR I = MAX,-1,0 %DO:
      K = 0
      K = K+1 %AND N = N-TENTO(I) %WHILE N >= TENTO(I)
      %IF M = N %THEN SPACE %ELSE PRINT SYMBOL('0'+K)
   %REPEAT
%END

%ROUTINE  PRINT LINE
%INTEGER  I
   LINE NO = LINE NO + 1
   WRITE(LINE NO,4); SPACES(2)
   PRINT SYMBOL(B(I)) %FOR I = 1,1,B(0)
   %IF (LINE NO//DIAGRAM SIZE)*DIAGRAM SIZE = LINE NO %THEN:
      %PRINTTEXT '<>'; NEWLINE
      %IF DIAGRAM SIZE+2 <= LINES ON PAGE %THEN:
         %PRINTTEXT '<>N <>T <>L'
         WRITE((LINES ON PAGE-DIAGRAM SIZE)//2,0)
         NEWLINE
      %FINISH
      %PRINTTEXT '<>D'; WRITE(DIAGRAM SIZE,0);  NEWLINE
   %FINISH
%END


%ROUTINE  MODIFY
%INTEGER  I,J,K,II,JJ,COMMENT,PC
%OWNINTEGER  ON = -1, OFF = 0
   J = 1; COMMENT = OFF; PC = 0
   %FOR I = 1,1,A(0) %DO:
      %IF A(I) = '''' %AND COMMENT # ON  %THEN:
      II = I; JJ = J
      K = 0; B(J) = '@'; B(J+1) = '['; J = J+2
      %UNTIL K = 0 %AND A(I-1) = '''' %AND A(I) # '''' %DO:
```

```
61              %IF A(I) = NL %THEN: ;! QUOTES NOT BALANCED - MUST BE A COMMENT
62                 I = II; J = JJ; COMMENT = ON; -> O
63              %FINISH
64              K = \K %IF A(I) = ''''
65              B(J) = A(I); I = I+1; J = J+1
66              %IF B(J-1) = ']' %AND A(I) # NL %THEN:
67                 B(J) = ']'; B(J+1) = '@'; B(J+2) = '['
68                 J = J+3
69              %FINISH
70           %REPEAT
71           B(J) = ']'; J = J+1
72           I = I-1; -> 1
73        %FINISH
74        %IF A(I) = '%' %THEN:
75           PC = PC+1     ;!  COUNT NUMBER OF '%'
76           B(J) = '%'; B(J+1) = '['; J = J+2
77           I = I+1
78           %UNTIL %NOT 'A' <= A(I) <= 'Z' %DO:
79              B(J) = A(I); I = I+1; J = J+1
80           %REPEAT
81           B(J) = ']'; J = J+1
82           I = I-1; -> 1
83        %FINISH
84     O: %IF A(I) = ';' %AND A(I+1) = '!' %THEN:
85           %WHILE PC # O %DO:
86              B(J) = ' '; J = J+1
87              PC = PC-1
88           %REPEAT
89        %FINISH
90        B(J) = A(I); J = J+1
91     1: %REPEAT
92        B(O) = J-1
93     %END
94
95
96
97        SELECT INPUT(O); SELECT OUTPUT(O)
98        %FAULT9 -> RESTART
99     RESTART: %PRINTTEXT 'FMTIMP   MAY ''72'; NEWLINE
100
101       %PRINTTEXT 'LINES ON PAGE, CHARS ON LINE=?'
102       PROMPT(':'); READ(LINES ON PAGE)
103       PROMPT(':'); READ(CHARS ON LINE)
104       DIAGRAM SIZE = (LINES ON PAGE//10)*10
105
106
107       SKIP SYMBOL %WHILE INPUT # O
108       %PRINTTEXT 'TYPE ANY OTHER OPTIONS REQ''D
109    <>F    '
110       SELECT OUTPUT(1)
111       %PRINTTEXT '<>F   '
112       PRINT SYMBOL(NEXT SYMBOL) %AND SKIP SYMBOL %WHILE NEXT SYMBOL # NL
113       NEWLINE
114
115       %PRINTTEXT '<>F ?I U @,%; D '
116       WRITE(LINES ON PAGE,O); %PRINTTEXT ','
117       WRITE(CHARS ON LINE,O); %PRINTTEXT ';'; NEWLINES(5)
118
119       %PRINTTEXT '<>N'
120       %IF DIAGRAM SIZE+2 <= LINES ON PAGE %THEN:
```

```
121           %PRINTTEXT ' <>T <>L'; WRITE((LINES ON PAGE-DIAGRAM SIZE)//2,0)
122       %FINISH
123       NEWLINE
124       %PRINTTEXT '<>D'; WRITE(DIAGRAM SIZE,0); NEWLINE
125
126
127
128       SELECT INPUT(1)
129       %FAULT9 -> STOP
130   1: READ LINE; MODIFY; PRINT LINE; -> 1
131   STOP: %PRINTTEXT '<>
132   <>E';  NEWLINE
133
134
135   %ENDOFPROGRAM
```

ATLAS2 4/4/72

```
#32=UL16
#10=UL2
#=L53,U31
!=L39          VERTICAL BAR
;=L31,L63
+=U61
-=U62
.=U63
@=U58
#20=UL28              STOPCODE
#8=UL21                 BACKSPACE
$=L39,U99         VERTICAL BAR, UPPERCASE S
&=L61
,=L63
(=L56
)=L57
*=L62
'=L48
:=L31
<=L51
==L53
>=L52
?=L60
[=L49
]=L50
_=L54
/=U31
0=U32      DIGITS 0-9
A=U81         UPPERCASE ALPHABET
#97=L81     LOWER CASE ALPHABET
```

ATLAS 1 4/4/72

```
%BEGIN
%INTEGER  I,J,X
%INTEGERARRAY  A(0:200), T(0:100)
%OWNINTEGER  UC=128, LC=16    ;!  UPPER/LOWER CASE MARKERS
%INTEGER  SIGNBIT; SIGNBIT = 1<<17

    %FAULT9 -> F9

    SELECT OUTPUT(0)
    A(0) = 0; T(0) = 0

1:  SKIP SYMBOL %WHILE NEXT SYMBOL = ' ' %OR NEXT SYMBOL = NL
    READ SYMBOL(I)
    READ(I) %IF I = '#' %AND '0' <= NEXT SYMBOL <= '9'
    %IF NEXT SYMBOL = '=' %THEN  SKIP SYMBOL  %ELSESTART
2:      %PRINTTEXT '?'; PRINT SYMBOL(I); %PRINTTEXT '/'; WRITE(I,1)
        NEWLINE; -> 4
    %FINISH

    SKIP SYMBOL %WHILE NEXT SYMBOL = ' '
    X = 0
3:  %UNTIL '0' <= NEXT SYMBOL <= '9' %THENSTART
        READ SYMBOL(J)
        -> 2 %IF 'U' # J # 'L'
        %IF J = 'U' %THEN X = X!UC %ELSE X = X!LC
    %FINISH
    READ(J); X = X!( J&(127-16) )    ;!  PARITY IN TRACK 5

    SKIP SYMBOL %WHILE NEXT SYMBOL = ' '
    %IF NEXT SYMBOL = ',' %THENSTART
        -> 2 %IF X&(-1<<8) # 0
        X = X<<8; SKIP SYMBOL; -> 3
    %FINISH
    %IF X&(-1<<8) = 0 %THENSTART
        A(0) = A(0)+1; A(A(0)) = (X<<7) + (I&127)
    %FINISHELSESTART
        T(0) = T(0)+1; T(T(0)) = I&127
        T(0) = T(0)+1; T(T(0)) = X
    %FINISH

4:  SKIP SYMBOL %WHILE NEXT SYMBOL # NL
    -> 1


F9: %IF T(0) # 0 %THENSTART
        %CYCLE  I = 1,2,T(0)-1
            A(0) = A(0)+1
            A(A(0)) = T(I) + ( (A(0)+T(0)//2)<<7 ) + SIGNBIT
            A( A(0) + T(0)//2 ) = T(I+1)
        %REPEAT
    %FINISH
    SELECT OUTPUT(1)
%PRINTTEXT '
%ROUTINE  PRINT ATLAS(%INTEGER  P)
! ATLAS FLEXOWRITERS USE THE SAME TAPE CODE FOR RUNOUT AS FOR
! UPPERCASE SHIFT.  IN ORDER TO BE SURE THAT THE PRINTER
! ACTUALLY IS IN UPPERCASE BEFORE STARTING PRINTING FOR SERIOUS
! IT IS DESIRABLE TO SEND AT LEAST ONE RUNOUT CHARACTER( = UPPER CASE
! SHIFT )
'
```

```
 61
 62       %PRINTTEXT '%OWNINTEGERARRAY  A(0:' *
 63       WRITE( A(0)+T(0)//2,1 ); %PRINTTEXT ')='
 64       WRITE( A(0),1 )
 65
 66       %CYCLE  I = 1,1,A(0)+T(0)//2
 67          PRINT SYMBOL(','); NEWLINE %IF ((I-1)//10)*10 = I-1
 68          WRITE(A(I),1)
 69       %REPEAT
 70
 71     %PRINTTEXT '
 72    %OWNINTEGER  CASE=128
 73    %INTEGER  I,K,X
 74    !
 75    %ROUTINE  PRINT(%INTEGER  P)
 76    %INTEGER  Q,N
 77       N = 0 ; Q = P
 78    1: %IF Q # 0 %THENSTART
 79          N = (Q&1)+N ; Q = Q >>1 ; -> 1
 80       %FINISH
 81       P = P!!16 %IF N&1 = 0     ;!  SET ODD PARITY IN TRACK 5
 82       PRINT SYMBOL(P)
 83    %END
 84    !
 85    %ROUTINE  SHIFT UP
 86       PRINT(7) ; CASE = 128
 87    %END
 88    !
 89    %ROUTINE  SHIFT DOWN
 90       PRINT(6) ; CASE = 16
 91    %END
 92    !
 93    !'; %PRINTTEXT'
 94       P = P&127
 95    !     UPPERCASE SHIFT CODE IS USED AS RUNOUT
 96       %IF P = 0 %THENSTART ; SHIFT UP ; %RETURN ; %FINISH
 97    !
 98       K = P
 99       K = ''A'' %IF ''A'' <= P <= ''Z''
100       K = ''A''+32 %IF ''A'' <= P-32 <= ''Z''
101       K = ''0'' %IF ''0'' <= P <= ''9''
102       %CYCLE  I = A(0),-1,1
103          -> 1 %IF A(I)&127 = K
104       %REPEAT
105    !  ''SPACE'' IS THE FIRST ENTRY IN ''A'' SO IF THE CHARACTER IN
106    !  ''K'' IS NOT MATCHED, A SPACE IS AUTOMATICALLY SUBSTITUTED
107    !'; %PRINTTEXT'
108    1: I = A(I)
109       X = ((I<<1)>>8)     ;!  >>7  AND GET RID OF SIGN BIT
110       X = A(X) %IF I < 0  ;!  IS IT A POINTER? :  < 0 : YES
111    !
112    2: %IF X&CASE = 0 %THENSTART      ;!  MUST SHIFT CASE
113          %IF CASE = 128 %THEN SHIFT DOWN %ELSE SHIFT UP
114       %FINISH
115    !  NOTE THAT BIT 5 MUST BE SET SO THAT ANY CARRIES SKIP TO POSN. 6
116       PRINT( ((X!16)&127)+P-K )    ;!   DEALS WITH ''A''-''Z'' ETC.
117       X = X>>8 ; %RETURN %IF X = 0
118       PRINT(21) ; -> 2     ;!  BACKSPACE TO OVERPRINT
119    %END;!ATLAS
120    '
```

121 %ENDOFPROGRAM