TECHNICAL MEMORANDUM 1871

EDITOR

A COMPUTER PROGRAM
FOR
DOCUMENTATION PURPOSES

WILLIAM H. BOLTE

DECEMBER 1968

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Best Available Copy
PICATINNY ARSENAL
DOVER, NEW JERSEY

Reproduced by the CLEARINGHOUSE for Federal Scientific & Technical Information
TECHNICAL MEMORANDUM 1871

EDITOR - A COMPUTER PROGRAM
FOR DOCUMENTATION PURPOSES

BY

WILLIAM H. ELLIS

DECEMBER 1968

DATA PROCESSING SYSTEMS OFFICE
PICA TINNY ARSENAL
OCEON, N.J.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>2. INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>3. GENERAL DESCRIPTION</td>
<td>3</td>
</tr>
<tr>
<td>4. CODE DESCRIPTION &amp; APPLICATION</td>
<td>4</td>
</tr>
<tr>
<td>A. EDITOR PROGRAM CODES</td>
<td>4</td>
</tr>
<tr>
<td>4. INDIVIDUAL SAMPLE CODE ILLUSTRATIONS</td>
<td>7</td>
</tr>
<tr>
<td>C. ILLUSTRATIVE EXAMPLES</td>
<td>11</td>
</tr>
<tr>
<td>5. APPENDICES</td>
<td>15</td>
</tr>
<tr>
<td>A. FLOW CHART OF BASIC PROGRAM LOGIC</td>
<td>16</td>
</tr>
<tr>
<td>B. LISTING OF FORTRAN IV SOURCE PROGRAM</td>
<td>21</td>
</tr>
<tr>
<td>6. TABLE OF DISTRIBUTION</td>
<td>26</td>
</tr>
</tbody>
</table>

PAGE 1
1. ABSTRACT

This report describes the application of an editing program called 'EDITOR' for documentation purposes. This program was written in FORTRAN IV language for the IBM-360/65 digital computer by Mr. Sidney Kravitz (Mechanical Engineer) of the Data Processing Systems Office, Picatinny Arsenal. It requires a total memory length of 20,592 bytes. This report enjoys the unique distinction of having been physically prepared by the very process it describes.

This program is presently being used on an experimental basis to evaluate its feasibility for Arsenal-wide distribution.
2. INTRODUCTION

Program documentation can be and often is a tedious, albeit necessary, task undertaken at the completion of developing a computer program. 'EDITOR' was written to expedite this task.

'EDITOR' offers a solution to the shortage of available typists and secretaries. It takes advantage of the IBM-1403 high speed printer located at the Pilatinnv Computer Center, thus enabling the user to have his reports 'typed' by the computer.

All text editing instructions (line spacing, underlining, centering, page numbering; etc...) are passed to the 'EDITOR' program by means of alpha numerical codes punched in column 1 of the input data cards. The data cards contain the text of the report, having been punched from FORTRAN coding sheets upon which the author has 'written' his report.

This program is rather simple and easy to use. However, it is felt that better understanding of the individual program codes will be achieved by studying actual samples and variations of each. Therefore, the latter part of this report will contain samples of most of the program codes and their effects as well as an illustrative example of part of the actual coded deck used for the documentation of this report.
3. GENERAL DESCRIPTION

This program was designed to allow the printed output pages to fall within the common government standard (8 x 10-1/2 inches) page size. Page numbering is accomplished automatically starting after the first page, since this is often the title page. Roman numeral numbering may be initiated at the option of the user up to a maximum of XVII. Arabic numeral numbering then begins and continues until the end of this report. If desired, additional reports can immediately follow, separated by the appropriate logic control cards.

Normally 24 lines of printed information are produced per page of output (double spacing). If page overflow occurs, the information is continued on the next page with the proper page number sequence.

This program produces output on a one to one basis, that is, one punched card of information produces one line of output information. All punched cards do not necessarily require program codes (in column 1) all the time. Only when a certain operation categorized by one of the program codes is desired, should it appear, being supplied by the user.
### 4. Code Description & Application

#### 4.A. Editor Program Codes

<table>
<thead>
<tr>
<th>CODE</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CENTER &amp; UNDERLINE THE LINE</td>
</tr>
<tr>
<td>C</td>
<td>CENTER THE LINE</td>
</tr>
<tr>
<td>F</td>
<td>LAST LINE ON THE LAST PAGE OF THE REPORT (A NEW REPORT MAY BE PLACED IMMEDIATELY AFTER THIS CARD)</td>
</tr>
<tr>
<td>L</td>
<td>LAST LINE ON A PAGE</td>
</tr>
<tr>
<td>S</td>
<td>SELECTIVE UNDERLINING - UNDERScores ONLY (BUT ALL OF) THE CHARACTERS FOUND ON THE CARD.</td>
</tr>
<tr>
<td>U</td>
<td>UNDERLINES EVERYTHING BETWEEN THE LEFTMOST CHARACTER AND THE RIGHTMOST CHARACTER OF THE LINE</td>
</tr>
<tr>
<td>V</td>
<td>THE FIRST 'V' PLACES THE PROGRAM IN THE 'V-MODE'. THE SECOND 'V' REMOVES IT FROM THE 'V-MODE'. THE THIRD 'V' PLACES THE PROGRAM BACK INTO THE 'V-MODE'. AGAIN, ETC... WHEN IN THE 'V-MODE', THE PROGRAM WILL LIST ALL 30 COLUMNS OF CHARACTERS AND SINGLE SPACE ALL LINES. ALL CODES FOUND IN COLUMN 1 ARE DISREGARDED AS COMMANDS BUT ARE NEVERTHELESS LISTED. THE ONLY EXCEPTION TO THE RULE IS THE 'V'-CODE ITSELF. A 'V' WILL NEVER BE LISTED IN COLUMN 1 OF THE LISTING WHETHER IN THE 'V-MODE' OR NOT. HOWEVER, PAGE NUMBERING STILL CONTINUES.</td>
</tr>
</tbody>
</table>
**MEANING**

**DIGIT**

Any odd digit between 1 and 9 will cause the program to skip (Digit+1) lines before printing the information found on that card.

**DIGIT**

Any even digit between 2 and 10 will cause the program to skip (Digit+1) lines after printing the information found on that card.

**N**

Allows single spacing (even though double spacing normally occurs - it is overridden in this case).

**R**

Roman numeral numbering is normally (automatically) initiated by the program starting with the second page of the report. 'R' denotes the last line of the last page for roman numerals. If no 'R' appears, the program will begin page 1 after page xvii. A second 'R' in the same report will be disregarded by the program. When the input data is preceded by an 'R' code, program control skips roman numeral numbering and employs numerical numbering only, again starting with the second page of the report.
THE FOLLOWING ARE SOME POINTERS TO REMEMBER CONCERNING THE USE OF 'EDITORS'.

1) NOT ALL CARDS REQUIRE A PROGRAM CODE.
2) REQUIRED CODES ARE TO BE PUNCTED IN COLUMN 1 ONLY.
3) A 'V' IN COLUMN 1 IS NEVER PRINTED ON THE OUTPUT LISTING REGARDLESS OF WHETHER THE PROGRAM IS IN THE 'V-MODE' OR NOT.
4) ONE PUNCHED CARD PRODUCES ONE LINE OF OUTPUT.
5) ROMAN NUMERAL NUMBERING CANNOT EXCEED XVII.
6) WITH PROPER USE OF THE 'DIGIT' & 'IN' CODES ANY NUMBER OF LINES CAN FOLLOW OR PRECEDE A LINE OF OUTPUT.
7) IF THE USER PUNCHED LINE INFORMATION BETWEEN COLUMNS 7 TO 72 INCLUSIVELY, PROPER PAGE ALIGNMENT WILL OCCUR. WRITING BEYOND COLUMN 72 COULD NECESSITATE THE NEED FOR PHOTOGRAPHIC REDUCTION IN ORDER FOR IT TO FALL WITHIN THE 4 x 10-1/2 INCH LIMITS.
8) UNLESS SPECIFIED BY A PROGRAM CODE, DOUBLE SPACING NORMALLY OCCURS.

PAGE 9
4.A. INDIVIDUAL SAMPLE CODE ILLUSTRATIONS

1. CODE - "A" - CENTER & UNDERLINE

CARD:

RESULT: CENTER & UNDERLINE EXAMPLE

2. CODE - "C" - CENTERS THE LINE

CARD:

RESULT: CENTER EXAMPLE

3. CODE - "S" - SELECTIVE UNDERLINING

CARD:

RESULT: SELECTIVE UNDERLINING EXAMPLE

4. CODE - "L" - UNDERLINE

CARD:

RESULT: UNDERLINE EXAMPLE
A. INDIAN SAMPLE CODE ILLUSTRATED INDEX

5. CODE - "DIGIT" - MULTIPLE SPACING

A) ODD NUMBER

CARDS:

FIRST CARD

SECOND CARD

RESULT:

FIRST CARD

SECOND CARD

FIRST CARD

(3+1) OR 4 SPACES

SECOND CARD

FIRST CARD

B) EVEN NUMBER

CARDS:

FIRST CARD

SECOND CARD

RESULT:

FIRST CARD

SECOND CARD

FIRST CARD

(4+1) OR 5 SPACES

FIRST CARD

PAGE 3
A. CODE - "V" - VERBATIM PCCE

Cards:

```
5  EAC 'V' PCCE
4  B17CCLC //*
3  2LZ 3C8FL TXY
2  REGIN 'V' PCCE
```

Result:

```
REGIN 'V' PCCE
2LZ 3C8FL TXY
B17CCLC //*
END 'V' PCCE
```

---

B. CODE - "N" - NC SKIP - SINGLE SPACE

Cards:

```
NC SKIP EXAMPLE
SINGLE SPACE
```

Result:

```
SINGLE SPACE
NC SKIP EXAMPLE
```

---

C. CODE - "F" - LAST LINE OF LAST PAGE OF REPORT

Card:

```
F (LINE DATA PRINTED IF PRESENT)
```

Result:

```
REINITIALIZES PAGE NUMBERING SEQUENCE
```
9. CODE - "R" - LAST PAGE OF ROMAN NUMERALS

CARD:

R (LINE DATA PRINTED IF PRESENT)

RESULT:
THE PAGE FOLLOWING BEGINS PAGE NUMBERING WITH 1

10. CODE - "L" - AN EXAMPLE OF THE "L" CODE (LAST LINE ON PAGE) CAN BE FOUND IN SECTION 4.C.
I. THE FOLLOWING IS THE LISTING OF THE TABLE OF CONTENTS OF THE ACTUAL PROGRAM CODED DECK USED FOR THE DOCUMENTATION OF THIS REPORT. IT WAS LISTED UNDER THE "V-MODE".

### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>2. INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>3. GENERAL DESCRIPTION</td>
<td>3</td>
</tr>
<tr>
<td>4. CODE DESCRIPTION &amp; APPLICATION</td>
<td>4</td>
</tr>
<tr>
<td>a. EDITOR PROGRAM CODES</td>
<td>5</td>
</tr>
<tr>
<td>b. INDIVIDUAL SAMPLE CODE ILLUSTRATIONS</td>
<td>7</td>
</tr>
<tr>
<td>5. ILLUSTRATIVE EXAMPLES</td>
<td>11</td>
</tr>
<tr>
<td>6. APPENDICES</td>
<td>15</td>
</tr>
<tr>
<td>7. FLOW CHART OF BASIC PROGRAM LOGIC</td>
<td>16</td>
</tr>
<tr>
<td>8. LISTING OF FORTRAN IV SOURCE PROGRAM</td>
<td>21</td>
</tr>
<tr>
<td>9. TABLE OF DISTRIBUTION</td>
<td>26</td>
</tr>
</tbody>
</table>
2. THE FOLLOWING IS THE LISTING OF SECTION 4.8. OF THE ACTUAL PROGRAM CODED OCK USED FOR THE DOCUMENTATION OF THIS REPORT. IT WAS LISTED UNDER THE 'V-NODE'.

### 4.8. INDIVIDUAL SAMPLE CODE ILLUSTRATIONS

1. **CODE**: "**A**" - CENTER & UNDERLINE
   
   **CARD**:  
   
   **RESULT**: CENTER & UNDERLINE EXAMPLE

2. **CODE**: "**C**" - CENTERS THE LINE
   
   **CARD**: C CENTER EXAMPLE

3. **CODE**: "**S**" - SELECTIVE UNDERLINING
   
   **CARD**: S SELECTIVE UNDERLINING EXAMPLE

4. **CODE**: "**U**" - UNDERLINE
   
   **CARD**: U UNDERLINE EXAMPLE

---

Page 17
4.6. ILLUSTRATIVE EXAMPLES (CONT.)

BEGIN "V" MODE
2LZ 308FPL TXY
BITCOLD /*
END "V" MODE

7. CODE - **NN** - NO SKIP - SINGLE SPACE
CARD:
NO SKIP EXAMPLE
RESULT:
SINGLE SPACE
NO SKIP EXAMPLE

9. CODE - **FF** - LAST LINE OF LAST PAGE OF REPORT
CARD:
F (LINE DATA PRINTED IF PRESENT)
RESULT:
REINITIALIZES PAGE NUMBERING SEQUENCE

C: INITIATE PAGE NUMBERING CODE ILLUS. (CONT.)

2. CODE - **FF** - LAST PAGE OF HUMAN MINERALS
CARD:
F (LINE DATA PRINTED IF PRESENT)
RESULT:
The page following begins page numbering with 1

1. CODE - **EE** - AN EXAMPLE OF THE "EE" CODE - LAST LINE ON
PAGE 1 CAN BE FOUND IN SECTION 2.6.
5. APPENDICES
5.4. FLOW CHART OF BASIC PROGRAM LOGIC
FLOW CHART OF BASIC LOGIC

PAGE 17
SUBROUTINE SELECT

NOCHARACTER

TEST CHARACTERS

BETA(I) = ' '
(BLANKS)

CHARACTER

BETA(I) = ' -'
(UNDERLINE)

RETURN

SUBROUTINE LIMIT

DETERMINE FIRST CHARACTER POSITION

DETERMINE LAST CHARACTER POSITION

RETURN
SUBROUTINE ULINE

- Calculate left most character position on card
- Calculate right most character position on card
- Return

SUBROUTINE CENTER

- Center line of information for output listing
- Return

SUBROUTINE SPLINE

- Print blank line
- Increment line count
- Return
SUBROUTINE PAGEINO

TEST
PAGE NUMBER

WHITE
PAGE NUMBER

WRITE
PAGE NUMBER

ARABIC NUMERALS

PAGE INCREMENT

CAUSE PRINTER TO MOVE TO TOP OF NEW PAGE

RETURN

PAGE 20
C EDITOR PROGRAM, SECOND VERSION; SEPTEMBER 1966

DIMENSION ALPHA(80), BETA(80), ROMAN(20)
DATA C/"C1", U/"U1", A/"A1", XN/"XN", R/"R", XL/"XL", P/"P",/
1 PAGE/"PAGE", X1/"X1", X2/"X2", X3/"X3", X4/"X4", X5/"X5",/
2 AO/"AO", X7/"X7", X8/"X8", X9/"X9", V/"V", SPACE/" SPACE",
3/

20 LMN=0
IA=0
LINE=1
LRoman=0
WRITE(6,3)
3 FORMAT(11/://///)
1 READ(5,2,END=50)XC,1 ALPHA(I),I=1,79
2 FORMAT(9041)
3 FORMAT(7X,79A1)
4 IF(XA.EQ.V) IA=IA+1
IA=IA/2
IX=IA-2*IA
IF(IX.NE.1) GO TO 15
IF(XA.EQ.V) XA=SPACE
BETA(I)=XC
DD 22 [1,79
BETA(I+1)=ALPHA(I)
22 CONTINUE
25 WRITE(6,23) (BETA(I),I=1,80)
23 FORMAT(17X,80A1)
26 LINE=LINES+1
GO TO 17
15 LSkip=0
IF(XA.EQ.X1) LSkip=1
IF(XA.EQ.X2) LSkip=2
IF(XA.EQ.X3) LSkip=3
IF(XA.EQ.X4) LSkip=4
IF(XA.EQ.X5) LSkip=5
IF(XA.EQ.X6) LSkip=6
IF(XA.EQ.X7) LSkip=7
IF(XA.EQ.X8) LSkip=8
IF(XA.EQ.X9) LSkip=9
LSkip=LSkip/2
LSkip=LSkip-2-LSkip
IF(LSkip=9,9,10
10 DD 11 [1,LSkip
CALL SPLINE(LINE)
IF(LIN.EQ.50) CALL PAGEND(LRomain,LINE)

PAGE 21
11 CONTINUE
9 IF(XC.EQ.C) CALL LIMIT(ALPHA, NR, NL)
   IF(XC.EQ.U) CALL LIMIT(ALPHA, NR, NL)
   IF(XC.EQ.A) CALL LIMIT(ALPHA, NR, NL)
   IF(XC.EQ.C) CALL CENTER(ALPHA, NR, NL)
   IF(XC.EQ.A) CALL CENTER(ALPHA, NR, NL)
   IF(XC.EQ.A) CALL LIMIT(ALPHA, NR, NL)
   IF(XC.EQ.U) CALL ULINE(BETA, NR, NL)
   IF(XC.EQ.1) GO TO 6
   IF(XC.EQ.A) GO TO 6
   IF(XC.EQ.XN) GO TO 6
   IF(XC.EQ.5) CALL SELECT(ALPHA, BETA)
   IF(XC.EQ.5) GO TO 6
5 WRITE(6,3)(ALPHA(I), I=1,79)
   LINE=LINE+2
   GO TO 12
6 WRITE(6,4)(ALPHA(I), I=1,79)
   LINE=LINE+1
   IF(XC.EQ.U) WRITE(6,7)(BETA(I), I=1,79)
   IF(XC.EQ.A) WRITE(6,7)(BETA(I), I=1,79)
   IF(XC.EQ.5) LINE=LINE+1
   IF(XC.EQ.5) LINE=LINE+1
   IF(XC.EQ.5) LINE=LINE+1
7 FORMAT(1H*,16X,79A1,7/)
12 IF(LSKIP.EQ.0) GO TO 13
   IF(LSKIP) 13,21,13
21 DO 14 I=1,LSKIP
   CALL SPLIT(LINE)
   IF(LINE.EQ.50) CALL PAGENO(ROMAN, LINE)
14 CONTINUE
13 IF(XC.EQ.R) GO TO 16
   IF(XC.EQ.F) GO TO 16
   IF(XC.EQ.XL) GO TO 16
17 IF(LINE.EQ.49) CALL SPLIT(LINE)
   IF(LINE.EQ.50) CALL PAGENO(ROMAN, LINE)
   GO TO 19
16 IF(LINE.EQ.50) GO TO 17
   LINE=LINE+1
   DO 18 I=LINE,49
   CALL SPLIT(LINE)
18 CONTINUE
   GO TO 17
19 IF(XC.EQ.F) GO TO 20
   IF(LMAN.EQ.1) GO TO 1
   IF(XC.EQ.R) LMAN=1
   IF(XC.EQ.R) LROMAN=18

PAGE 22
GO TO 1
50 STOP
END

SUBROUTINE SELECT(ALPHA,BETA)
DIMENSION ALPHA(80),BETA(80)
DATA RANK/* ','HANK/* '/*
DO 1 I = 1,79
1 IF(ALPHA(I).EQ.RANK) BETA(I)=RANK
IF(ALPHA(I).NE,RANK) BETA(I)=HANK
CONTINUE
RETURN
END

SUBROUTINE LIMIT(ALPHA,NR,NL)
DIMENSION ALPHA(80)
DATA BLANK/* '
NL=0
NR=0
DO 1 I=1,79
1 IF(BLANK.NE,ALPHA(I))GO TO 2
CONTINUE
3 RETURN
2 NL=1
DO 4 I=1,79
J=80-I
4 IF(BLANK.NE,ALPHA(J)) GO TO 5
CONTINUE
GO TO 3
5 NR=J
GO TO 3
END
SUBROUTINE ULINE(BETA, NR, NL)
DIMENSION BETA(80)
DATA BANK/* */
DATA CRANK/* */
NL=NL-1
NR=NR+1
IF(NL.EQ.0) GO TO 2
DO 1 I=1, NL
BETA(I)=BANK
1 CONTINUE
DO 3 I=NL, NR
BETA(I)=CRANK
3 CONTINUE
IF(NR.EQ.80) GO TO 4
DO 5 I=NR, 79
BETA(I)=BANK
5 CONTINUE
4 RETURN
END

SUBROUTINE CENTER(ALPHA, NR, NL)
DIMENSION ALPHA(80), GAMMA(80)
DATA TANK/* */
DO 8 I=1, 79
GAMMA(I)=ALPHA(I)
8 CONTINUE
NA=NR-NL+1
NB=(79-NA)/2
NC=NB-NL
IF(NC)1, 2 3
2 RETURN
3 J=79-NC
DO 6 I=1, NC
ALPHA(I)=TANK
6 CONTINUE
DO 5 I=1, J
NE=NC+I
ALPHA(NE)=GAMMA(I)
5 CONTINUE
GO TO 2
1 NC=NC
J=79-NC
DO 9 I=1, J
9 CONTINUE
NE = NE + 1
ALPHA(I) = GAMMA(NE)
9 CONTINUE
J = J + 1
DO 10 I = J, 79
ALPHA(I) = TANK
10 CONTINUE
GO TO 2
END

SUBROUTINE SPLINE(LINE)
WRITE(6,1)
1 FORMAT(LHO)
LINE = LINE + 1
RETURN
END

SUBROUTINE PAGENO(LROMAN, LINE)
DIMENSION ROMAN(20)
DATA PAGE/'PAGE'/
DATA ROMAN/'I', 'II', 'III', 'IV', 'V', 'VI', 'VII',
        'VIII', 'IX', 'X', 'XI', 'XII', 'XIII', 'XIV', 'XV', 'XVI',
        'XVII'/
MAN = LROMAN - 17
1 FORMAT(LHO)
   IF(LROMAN.EQ.0) GO TO 5
   WRITE(6,1)
   IF(LROMAN.LT.18) WRITE(6,2) PAGE, ROMAN(LROMAN)
2 FORMAT('SOX', A4, 'X', A4)
   IF(LROMAN.GE.18) WRITE(6,3) PAGE, MAN
3 FORMAT('SOX', A4, 'X', A4)
5 LROMAN = LROMAN + 1
LINE = 1
WRITE(6,4)
4 FORMAT('I'/////) RETURN
END
The report describes the application of an editing program called "EDITOR" for documentation purposes.

This program was written in FORTRAN IV Language for the IBM 360/65 Digital Computer and requires a total memory length of 27,582 Bytes.

The "EDITOR" program permits the user to have his reports "typed" on an IBM 1403 high-speed printer. All text editing instructions (line spacing, underlining, centering, page numbering, etc...) are passed to the "EDITOR" program by means of Alpha Numerical codes punched in Column 1 of the input data cards. The data cards contain the text of the report, having been punched from FORTRAN coding sheets upon which the author has "written" his report.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDITOR</td>
<td>ROLE</td>
<td>ROLE</td>
<td>ROLE</td>
</tr>
<tr>
<td>Documentation</td>
<td>ROLE</td>
<td>ROLE</td>
<td>ROLE</td>
</tr>
<tr>
<td>Computer Generated Text</td>
<td>ROLE</td>
<td>ROLE</td>
<td>ROLE</td>
</tr>
</tbody>
</table>
EDITOR MODIFICATION
Technical Memorandum #1871
Modified: 31 March 1969

CONTENTS: (1) "P" Code Description - Add to Section 4.
"Code Description and Application"

(2) Program Listing Replacement -
Replaces old TM 1871 Listing of Section 5. B.
"Listing of FORTRAN Source Program"

WILLIAM H. BOLTE
SEAD/DPSO Bldg. 351
Picatinny Arsenal
Ext. 3663
PROGRAM: "EDITOR" - A compiler program for documentation purposes (T.M. 1871)

MODIFIED: 31 March 1969

MODIFICATION: Added "P" Code

PURPOSE OF CODE: The "P" code allows one to skip a certain specified number of pages. This number is a user option.

INSTRUCTIONS FOR "P" CODE USAGE:

Punch the letter "P" in Column 1 of the card to be used, followed by an integer number. This number specifies the number of pages to be skipped and has an allowable field width of two (2); therefore, each data item must be right justified in its field, since leading and trailing blanks are treated as zeros (0's).

i.e. ___________
     Col 1:2:3

NOTE: Any information punched after column three (3) is ignored and hence not printed during output.

ILLUSTRATIVE USE:

CODE - "P" - Multiple Page Skip

CARDS:

Printing on Card #3
P 2
L End of this page

RESULT:

END OF THIS PAGE
(Skips two (2) pages)

PRINTING ON CARD #3
(In this case it is printed on top of third (3rd) page)

COMMENT:

This modification sheet is to be used in accordance with the Technical Memorandum #1871. Any problems, questions, inquiries or suggestions regarding the use of this code or any other subject matter from the T.M. 1871 should be directed to Mr. William H. Holte, SEAD/DFSO, Building 351, Ext. 3663.
*EDIT* PROGRAM LISTING REVISION 31 MAR 69

C REPLACES TM=1871 PROGRAM LISTING

C EDITN PROGRAM, FOURTH VERSION, MARCH 1969
*P* CODE ADDED

DIMENSION ALPH(A), BETA(R), ROMAN(20)

DATA C('C')/U('U')/A('A')/XN('X')/R('R')/XL('L')/P('P')/PAGE

10/X1/1'/X2/2'/X3/3'/X4/4'/X5/5'/X6/6'/X7/7'/X8/8'/

2/X9/9'/V/V'/SPACE

0/1/F/1/F/S/9'

1 LINE=0

I=0

LINE=1

LROMAN=0

WRITE (6, 22)

1 READ (5, 23, END=28) XC, (ALPHA(I), I=1, 79)

1 IF (XC.EQ.V) IA=IA+1

1 W=IA/2

1 X=IA-2*W

1 IF (X.EQ.1) GO TO 4

1 IF (X.EQ.2) XC=SPACE

1 BETA(I)=XC

1 DD 3 I=1, 79

1 BETA(I+1)=ALPHA(I)

1 CONTINUE

1 WRITE (6, 26) (BETA(I), I=1, 80)

1 LINE=LINE+1

1 IF I.OF (6, 26) I=I+3

1 CONTINUE

1 LSKIP=0

1 IF (XC.EQ.X1) LSKIP=1

1 IF (XC.EQ.X2) LSKIP=2

1 IF (XC.EQ.X3) LSKIP=3

1 IF (XC.EQ.X4) LSKIP=4

1 IF (XC.EQ.X5) LSKIP=5

1 IF (XC.EQ.X6) LSKIP=6

1 IF (XC.EQ.X7) LSKIP=7

1 IF (XC.EQ.X8) LSKIP=8

1 IF (XC.EQ.X9) LSKIP=9

1 LSKIP=LSKIP/2

1 LSKIP=LSKIP/2

1 IF (LSKIP) 7, 7, 5

1 DO 6 I=1, 5, LSKIP

6 PAGE 21
CALL SPLINE (LINE)
IF (LINE.EQ.50) CALL PAGE(1) (LROMAN,LINE)
   CONTINUE
7 IF (XC.EQ.) CALL LIMIT (ALPHA,NR,NL)
   IF (XC.EQ.U) CALL LIMIT (ALPHA,NR,NL)
   IF (XC.EQ.A) CALL CENTER (ALPHA,NR,NL)
   IF (XC.EQ.A) CALL LIMIT (ALPHA,NR,NL)
   CALL PAFNL.(LROMAN,LINE)
   IF (LINE.EQ.50) CALL PAGFNL.(LROMAN,LINE)
11 CONTINUE
12 IF (XC.EQ.) GO TO 14
   IF (XC.EQ.U) GO TO 14
   IF (XC.EQ.A) GO TO 14
13 IF (LINE.EQ.49) CALL SPLINE (LINE)
   IF (LINE.EQ.50) CALL PAGE(1) (LROMAN,LINE)
   GO TO 16
4 IF (LINE.EQ.50) GO TO 11
   IF (LINE.EQ.50) GO TO 11
15 CONTINUE
   GO TO 13
16 IF (XC.EQ.) GO TO 1
   IF (LROMAN.EQ.) GO TO 1
IF (XC.EQ.R) LMAN=1
GO TO 2
17 AA=ALPHA(1)
   AB=ALPHA(2)
   MA=0
   IF (AA.EQ.X1) MA=1
   IF (AA.EQ.X2) MA=2
   IF (AA.EQ.X3) MA=3
   IF (AA.EQ.X4) MA=4
   IF (AA.EQ.X5) MA=5
   IF (AA.EQ.X6) MA=6
   IF (AA.EQ.X7) MA=7
   IF (AA.EQ.X8) MA=8
   IF (AA.EQ.X9) MA=9
   MR=0
   IF (AB.EQ.X1) MB=1
   IF (AB.EQ.X2) MB=2
   IF (AB.EQ.X3) MB=3
   IF (AB.EQ.X4) MB=4
   IF (AB.EQ.X5) MB=5
   IF (AB.EQ.X6) MB=6
   IF (AB.EQ.X7) MB=7
   IF (AB.EQ.X8) MB=8
   IF (AB.EQ.X9) MB=9
   MC=10*MA+MB
   IF (LINE.EQ.1) GO TO 19
   IF (LINE.NE.50) CALL SPLINE LINE
   IF (LINE.NE.50) GO TO 13
   IF (LINE.NE.50) CALL PAGENO (LRMAN,LINE)
19   D) 21 I=1,MC
   D) 20 J=1,49
20   CALL SPLINE LINE
   CALL PAGENO (LRMAN,LINE)
21   CONTINUE
   GO TO 2
28   STOP
C
22   FORMAT (111/) / 
23   FORMAT (80A1)
24   FORMAT (17X,79A1,/)  
25   FORMAT (17X,79A1)
26   FORMAT (17X,80A1)
27   FORMAT (1H+,16X,79A1,/)  
CND

PAGE 23
SUBROUTINE SELECT (ALPHA, BETA)
DIMENSION ALPHA(80), BETA(80)
DATA RANK(''), HANK('')
DO 1 I=1,79
IF (ALPHA(I).EQ.RANK) BETA(I)=RANK
IF (ALPHA(I).NE.RANK) BETA(I)=HANK
1 CONTINUE
RETURN
END

SUBROUTINE LIMIT (ALPHA, NR, NL)
DIMENSION ALPHA(80)
DATA BLANK('')
NL=0
DO 3 I=1,79
IF (BLANK.NE.ALPHA(I)) GO TO 3
3 CONTINUE
RETURN
END

SUBROUTINE JLINE (META, N, NL)
DIMENSION META(80)
DATA RANK(''), CRANK('')
NL=NL-1
META(N)+=1
IF (NLM.EQ.0) GO TO 2
DO 5 I=1,NL
META(I)=RANK
5 CONTINUE
END
C N T I N U E
I F ( N R P . E Q . 8 0 ) G O T O 5
D ) 4 I = N R P - 7 9
B E T A ( I ) = B A N K
C N T I N U E
R E T U R N
E N D

D I M E N S I O N A L P H A ( 9 0 ) , G A M M A ( 8 0 )
D A T A T A N K / /
D I 1 1 I = 1 , 7 9
G A M M A ( 1 ) = A L P H A ( 1 )
C N T I N U E
N A = N K - N L + 1
N M = ( N L - N A ) / 2
N C = N H - N L
I F ( N C ) 5 , 2 , 3
R E T U R N
J = 7 9 - N C
D O 4 I = 1 , N C
A L P H A ( I ) = T A N K
C N T I N U E
D O 5 1 = 1 , J
N C = N C + 1
A L P H A ( N C ) = G A M M A ( 1 )
C N T I N U E
G O T O 2
N C = N C
J = 7 9 - N C
D O 7 1 = 1 , J
N C = N C + 1
A L P H A ( N C ) = G A M M A ( N L )
C N T I N U E
J = J + 1
D O 1 1 1 = 1 , 7 9
A L P H A ( 1 ) = T A N K
C N T I N U E
G O T O 2
E N D

S U B R O U T I N E S P L I N E ( L I N E )
SUBROUTINE PAGEINJ (LRoman, Line)

DIMENSION ROMAN(20)
DATA PAGE, PAGE/ 
"DATA ROMAN'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI'
1, 'XII', 'XIII', 'XIV', 'XV', 'XVI', 'XVII', 'XVIII'
" 
MAN=ROMAN-17
IF (LRoman.LE.9) GO TO 1
WRITE (6,1) 
IF (LRoman.LT.14) WRITE (6,3) PAGE, ROMAN(LRoman)
IF (LRoman.LE.14) WRITE (6,4) PAGE, MAN
1 LROMAN=ROMAN+1
LINE=1
WRITE (6,5)
RETURN
C
2 FORMAT (1HU)
3 FORMAT (50X, 34, 1X, 34)
4 FORMAT (50X, 44, 14)
5 FORMAT (**/****/
END