

# SHARE PROGRAM LIBRARY AGENCY



PROGRAM NUMBER

**060008**

---

## University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA  
(305) - 284-6257

SHARE PROGRAM LIBRARY SUBMITTAL FORM

SHARE PROGRAM LIBRARY AGENCY  
Triangle Universities Computation Center  
Post Office Box 12076  
Research Triangle Park, North Carolina  
27709 USA  
Attention: Mr. Joe Ragland

SPLACONTROL NUMBER: 142

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the "SHARE Program Library Standards Manual".

- (1) Program Number (to be filled in by SPLA) ..... 360D-06.0.008
- (2) System Type (machine) ..... S/360
- (3) Search Key ..... TEXT-FORMATTING
- \_\_\_\_\_
- \_\_\_\_\_
- (4) Programming Language ..... PL/I (IBM 360 F level)
- (5) Author's Name and Address .....  
Dr. James E. George  
Los Alamos Scientific Laboratory  
P. O. Box 1663, MS 272  
Los Alamos, NM 87545
- (6) Direct Inquiries to Name and Address .....  
(if different than Author) \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- (7) Title of Program ..... PRINT - A Text Formatting Program
- \_\_\_\_\_
- \_\_\_\_\_
- (8) Submitter's Installation Membership Code..... SIA
- (9) Submitter's Own Program Identification and Suffix(Optional)... PRINT
- (10) Primary Subject Code..... 06 0
- (11) Operating or Monitor System Required OS
- (12) New or Revision Code (if revision, show prior Program Number in Item 1).. N
- (13) Year Completed..... 1971
- (14) Date of Submittal..... May 1973
- (15) Documentation (number of original pages submitted)..... 32
- (16) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

# SHARE PROGRAM LIBRARY SUBMITTAL FORM

## DISCLAIMER

Triangle Universities Computation Center (TUCC) serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, INC., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

### Subject Guide:

- Purpose
- Programming Language used
- Version and modification level or release number
- Field of application
- Type of routine (main program, subroutine, etc.)
- Specific description of machine requirements

ABSTRACT PRINT is a text formatting program written in PL/I to produce documents using the printer. The input to PRINT contains the text to be printed interspersed with the necessary control information to generate the desired format of the text. The features supported by PRINT are:

Automatic or manual paging

Page numbering with or without titling

Paragraphing with or without indentation (left or right), numbering and/or titling

Underlining

Tables with or without numbering

Print direct image (i.e. print text as is)

Right justification of the text

Tabs

Also, the margin, line length and spacing between lines are variable. The aim was to provide a modular system which would be easy to change and would allow the value of any control variable to be changed by input control.

(Please attach additional pages if necessary).....Total pages attached \_\_\_\_\_

### Permission to Publish

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program."

(17) Signature of Submitter and Date

(18) Signature of Installation Addressee

*James E. George* 5/31/73  
*Guy T. Schaaf*

August 1971

PRINT - A Text Formatting Program

James E. George  
Stanford Linear Accelerator Center  
Computation Group  
Stanford, California

1. ABSTRACT

PRINT is a text formatting program written in PL/I to produce documents using the printer. The input to Print contains the text to be printed interspersed with the necessary control information to generate the desired format of the text. The features supported by PRINT are:

- Automatic or manual paging
- Page numbering with or without titling
- Paragraphing with or without indentation (left or right), numbering and/or titling
- Underlining
- Centering
- Tables with or without numbering
- Print direct image (i.e. print text as is )
- Right justification of the text
- Tabs

Also, the margin, line length and spacings between lines are variable. The aim was to provide a modular system which would be easy to change and would allow the value of any control variable to be changed by input control.

2. INTRODUCTION

The use of computer programs to format text to a final form for publication has been discussed in several documents (Meadow 1970; van Dam and Rice 1970; Berns 1968 and 1969). Basically, the problem is to take a linear string input and convert it to a multi-sheet two-dimensional document. Further, the final document is aesthetically good or bad by some undefinable human judgement. A really good program does not currently exist. PRINT represents a useable basic system for the preparation of a limited class of documents of interest to the author.

Initially, I became interested in text formatting

because I wanted to maintain a bibliography using a computer text editing system; I was willing to maintain the entries in alphabetic order, but wanted to be able to number the entries and vary the formatting on output (e.g. bibliography vs. working form for constructing an annotated bibliography). Although TEXT 360 (----1968) existed at the time, it was not locally available; FORMAT (Berns 1968 and 1969) was available\* but did not have the numbering feature. Hence, PRINT was originally developed to provide an automatic numbering for successive entries for the preparation of bibliographies; I also decided to develop it as a personal document formatting system which I could easily experiment with and modify for my own purposes.

Several documents have been produced using PRINT (George 1969a and b; George and Hoffman 1969), the library at SLAC has used it to produce a serials list (Section 5.2), a two-dimensional expression capability has been illustrated with PRINT (George 1971) and the cost of using it has been studied (George and Hoffman 1969).

### 3. GENERAL DESCRIPTION

An output document from PRINT consists of chapters, pages (with/without page title and /or page numbering), paragraphs (with/without paragraph titles and/or paragraph numbering), footnotes (with/without footnote numbering), tables (with/without numbering), centered text, unaltered text, underlined text, text whose position is controlled by tabs and text which may be right justified\*\*. In order to form these output units, the input must be identified as these units.

An input unit is denoted by a control character in column 1; a blank in column 1 generally means a continuation of the previous classification. Using these input units, PRINT maintains two buffers from which it composes the output document one line at a time. One buffer contains the text input, excluding footnotes, which has not been used (this buffer will be called the input buffer) and the other

-----

\*The version of FORMAT locally available has been extensively modified by John Ehrman (Ehrman-Berns 1971) and was used to prepare Dave Gries's book (Gries 1971).

\*\*A good discussion of justification is given in Meadow (Meadow 1970, pgs. 253-256).

buffer contains the unused footnotes (this buffer will be called the footnote buffer).

PRINT's normal cycle is to select and print full lines from the input buffer; when the input buffer contains less than one full line, more input is requested from an input section. Footnotes interrupt the normal cycle whenever just enough room is left at the bottom of the current page to print the current contents of the footnote buffer. Full lines are selected by examining the buffer (input or footnote) starting at the current line length to be printed and scanning to the left for the first occurrence of a break character.

Footnotes are a problem in this and many other systems (e.g. see the discussion in Meadow 1970, pgs. 259-260). The problem is to at least start the footnote on the page where it is referenced. In PRINT, footnotes and references to footnotes are unrelated; a footnote is printed at the bottom of a page which usually has just enough room for the footnote. Generally, the footnotes occur on the same page as the reference, however it is possible for the footnote to appear before or after this page; this can be altered by the manual paging control or the drain buffer control.

The input function of PRINT separates the single input stream into footnotes, text and format control. The input continues processing until a non-footnote or non-format control is obtained.

The beginning of a footnote unit is indicated by the footnote control character occurring in column 1; all succeeding cards which have a blank in column 1 are part of the footnote and are saved in the footnote buffer. A footnote is terminated by the occurrence of a non-blank control character in column 1, other than the footnote control character.

The beginning of a format control change is indicated by the occurrence of the read new parameter control character in column 1. Upon recognition of this control character, the input function of PRINT executes a GET DATA statement in PL/I thus allowing the value of any variable defined to be altered; usually these are the parameters controlling the formatting of the text.

In most cases, PRINT has two copies of each variable; one copy is used for format control and the value of the second copy is assigned to the first copy upon recognition

of a text unit. Hence, format control is specified before the text item appears and is assigned when the beginning of this text unit is recognized.

#### 4. CONTROL AND VARIABLE DESCRIPTIONS

The text control tokens must occur in column 1 with the beginning of the text on the same card. These tokens are:

<u>NAMES *</u>	<u>INITIAL VALUE</u>	<u>EXPLANATION</u>
PARC(PAR)	P	Indicates start of a new paragraph with numbering.
IDC(ID)	I	Indicates start of a new paragraph without numbering.
TABLE(TABLEG)	T	Indicates start of a table without numbering.
TABLEN(TABLNG)	S	Indicates start of a table with numbering.
CENC(CEN)	C	Indicates that the text on this card is to be centered.
ASIS(ASI)	A	Indicates that the text on this card is to be printed as is.

All the text controls above may involve indentations except centering; the indentation is in terms of a multiplier of the basic indentation unit. For example,

12Start of a paragraph

would start a new paragraph whose first line would be indented 2 times the basic unit to the right, and

1-2Start of a paragraph

would start a new paragraph whose first line would be indented 2 times the basic unit to the left.

The table option functions like the paragraph for the

-----

The name of the second copy of the variable is given followed by the name of the first copy in parentheses throughout this section; all control tokens are of type CHAR(1).

first line and then prints subsequent lines starting at the fifth tab position.

The footnote control character is (also type CHAR(1)):

<u>NAMES</u>	<u>INITIAL VALUE</u>	<u>EXPLANATION</u>
FTC(FT)	\$	Indicates the beginning of a footnote.

Like the paragraph control, a single digit or a minus sign and a digit may immediately follow the footnote control to indicate the desired indentation.

The controls which may occur anywhere in any text are (also of type CHAR(1)):

<u>NAMES</u>	<u>INITIAL VALUE</u>	<u>EXPLANATION</u>
UNDERC(UNDER)	-	Indicates the beginning or end of the underlining; all text bracketed by underlines will be underlined. Should an error occur, the underline will be turned off by the next occurrence of a non-blank control character in column 1.
CTABC(CTAB)	@	Indicates a tab position.

The format control tokens must also occur in column 1 and any text contained on the card is lost. The controls are of type CHAR(1) and are:

<u>NAMES</u>	<u>INITIAL VALUE</u>	<u>EXPLANATION</u>
TERMC(TERM)	F	Terminate PRINT.
PAGC(PAGE)	X	Drain the input buffer and start a new page.
DR(DRA)	R	Drain the input buffer.
PARMC(PARM)	D	The following cards until an unquoted ';' contain variable names and values to read with PL/I's GET DATA statement.



CHAP(CHAPC) # Drain all buffers and initialize all variables.

The variables used during the printing are:

<u>NAMES</u>	<u>TYPE</u>	<u>INITIAL VALUE</u>	<u>EXPLANATION</u>
COLNUM(NUMCOL)	FIXED BIN	80	Number of columns of the input card to be used.
BMAR(MARB)	FIXED BIN	10	Lines of margin at the bottom of a page.
LEN(L)	FIXED BIN	50	Length of line to be printed.
LMAR(COL)	FIXED BIN	50	Width of left margin in characters.
LSPACE(SPACE1)	FIXED BIN	1	Inner paragraph, footnote and table spacing.
PSPACE(SPACE2)	FIXED BIN	2	Spacing at the beginning of a paragraph or footnote.
TSPACE(SPACE3)	FIXED BIN	2	Spacing at the beginning of a table.
IDENT(DBL)	FIXED BIN	5	Basic unit for indentation.
TMAR(MART)	FIXED BIN	10	Lines of margin at the top of a page.
TABA(TAB(1))	FIXED BIN	5	Character position from left margin for first tab position.

TABB(TAB(2))	FIXED BIN	10	" " " for second tab position.
TABC(TAB(3))	FIXED BIN	15	" " " for third tab position.
TABD(TAB(4))	FIXED BIN	20	" " " for fourth tab position.
TABD(TAB(5))	FIXED BIN	25	" " " for fifth tab position.
----- (PGNUM)	FIXED BIN	1	Current page number; for new chapter '0' implies previous numbering.
----- (PARNUM)	FIXED BIN	1	Current paragraph number; for new chapter '0' implies previous numbering.
----- (FTNUM)	FIXED BIN	1	Current footnote number; for new chapter '0' implies previous numbering.
----- (FTCOL)	FIXED BIN	50	Left margin for footnotes.
----- (FTL)	FIXED BIN	50	Line length for footnotes.
SA(S1)	BIT(1)	TRUE	Number pages if true.
SB(S2)	BIT(1)	TRUE	Number paragraphs if true.
SC(S3)	BIT(1)	TRUE	Right justify if true.
SD(S4)	BIT(1)	TRUE	Number footnotes if true.

PGTITLE(PTITLE) CHAR(5) VAR NULL Paragraph title.  
 CHTITLE(CTITLE) CHAR(100) VAR NULL Chapter title.

The break characters which determine how the input is broken into lines are (all of type CHAR(1)):

<u>NAMES</u>	<u>INITIAL VALUE</u>
B1	(blank)
B2	. (period)
B3	, (comma)
B4	; (semicolon)
B5	: (colon)
B6	?
B7	! (exclamation point)
B8	- (dash)
B9	_ (underline)

Also, a character (variable MARKER) is printed in character position 1 of a line containing underlining or page faults; this allows the local version of a text editor (WYLBUR) to output the text to a typewriter.

A complete listing of PRINT is given in Appendix A.

## 5. EXAMPLES

Some examples were indicated in the Introduction. Some short examples will be given here.

### 5.1. Bibliography example

The input

P0George, James E. (19691). The System Specifcation of GLAF: A Linear String Graphical Language Facility. Stanford Linear Accelerator Center, Computation Group, GSG 61.

P0Meadow, Charles T. (1970). Editing Text, Chapter 9 in Man-Machine Communication, Wiley, 244-278.

produces the output

1. George, James E. (19691). The System Specifcation of GLAF: A Linear String Graphical Language Facility. Stanford Linear Accelerator Center, Computation Group, GSG 61.
2. Meadow, Charles T. (1970). Editing Text, Chapter 9 in Man-Machine Communication , Wiley, 244-278.

## 5.2. Serials example

The input

```
D
PSPACE=2 TSPACE=1 SC='0'B TABE=1 CTABC='@'
TAB=5 TABB=30 TABC=45 TABD=53 LEN=70 LMAR=50 BMAR=5 TMAR=5 LSPACE=1;
C_S L A C LIBRARY SERIALS LIST_
T0@@@FEBRUARY 1971
I-1A C M
see--Association for Computing Machinery
I-1A C M ARTIFICIAL INTELLIGENCE (SIGART)
T0@on order @@Central Lab Library
I-1A C M COMMUNICATIONS (SIGCOMM)
T0@v.1- @1970- @Central Lab Library
I-1A C M COMPUTER GRAPHICS (SIGGRAPH)
T0@v.5- @1971- @Central Lab Library
I-1A C M LANGUAGE ANALYSIS AND STUDIES IN THE HUMANITIES (SICLASH)
T0@v.2- @1969- @Central Lab Library
I-1A C M MICROPROGRAMMING (SIGMICRO)
T0@on order @@Central Lab Library
```

produces the output

S L A C LIBRARY SERIALS LIST

FEBRUARY 1971

A C M

see--Association for Computing Machinery

A C M ARTIFICIAL INTELLIGENCE (SIGART)

on order

Central Lab Library

A C M COMMUNICATIONS (SIGCOMM)

v.1-

1970-

Central Lab Library

A C M COMPUTER GRAPHICS (SIGGRAPH)

v.5-

1971-

Central Lab Library

A C M LANGUAGE ANALYSIS AND STUDIES IN THE HUMANITIES (SICLASH)

v.2-

1969-

Central Lab Library

A C M MICROPROGRAMMING (SIGMICRO)

on order

Central Lab Library

## 5.3. This paper

The input which produces this paper is given in  
Appendix B.

BIBLIOGRAPHY

1. Berns, G.M. (1968). Some Design Considerations for a Fast, General Purpose Text Processor, and the FORMAT Program. IBM Data Processing Division, Washington Scientific Center, 320-3503.
2. Berns, Gerald M. (1969). Description of FORMAT, a Text-Processing Program. CACM, 12, 3(March), 141-146.
3. Ehrman, John R. and Berns, Gerald M. (1971). FORMAT, A Text Processing Program. Stanford Linear Accelerator Center, SLAC-135.
4. George, James E. (1969a). The System Specification of GLAF: A Linear String Graphical Language Facility. Stanford Linear Accelerator Center, Computation Group, GSG 61.
5. George, J.E. (1969b). Rules for Transforming a Grammar to a Simple Precedence Grammar Utilizing Artificial Productions. Stanford Linear Accelerator Center, Computation Group, GSG 62.
6. George, James E. (1971). GEMS-A Graphical Experimental Meta System. Stanford Linear Accelerator Center, SLAC-134.
7. George, James E. and Hoffman, Lance J. (1969). A Comparison of Manual and Computer-Based Text Editing. Stanford Linear Accelerator Center, Computation Group, CGTM 69.
8. Gries, David (1971). Compiler Construction for Digital Computers , Wiley.
9. Meadow, Charles T. (1970). Editing Text, Chapter 9 in Man-Machine Communication , Wiley, 244-278.
10. van Dam, Andries and Rice, David E. (1970). Computers and Publishing: Writing, Editing and Printing. in Advances in Computers , 10, 145-174.
11. -----(1968). TEXT 360 Reference Manual and Operating Guide. IBM Corporation, New York.

## APPENDIX A

## Listing of PRINT

```

PRINT: PROC OPTIONS(MAIN);
/*THESE VARIABLES ARE USED FOR THE ACTUAL CONTROL*/
DCL (CHAP      CHAR(1)      , /*CONTROL CHAR FOR CHAPTER      */
     FT        CHAR(1)      , /*CONTROL CHAR FOR FOOTNOTE     */
     UNDER     CHAR(1)      , /*CONTROL CHAR FOR UNDERLINE    */
     TERM      CHAR(1)      , /*CONTROL CHAR FOR TERMINATION   */
     CEN        CHAR(1)      , /*CONTROL CHAR FOR CENTERING     */
     ID         CHAR(1)      , /*CONTROL CHAR FOR INDENTATION   */
     PAR        CHAR(1)      , /*CONTROL CHAR FOR PARAGRAPH     */
     PAGE       CHAR(1)      , /*CONTROL CHAR FOR MANUAL PAGE   */
     TABLEG    CHAR(1)      , /*CONTROL CHAR FOR TABLES      */
     TABLNG     CHAR(1)      , /*CONTROL FOR TABLE W NUMBER   */
     PARM       CHAR(1)      , /*READ NEW PARAMETER CONTROL    */
     CTAB       CHAR(1)      , /*TAB CONTROL CHAR              */
     DRA        CHAR(1)      , /*DRAIN CONTROL CHAR            */
     ASI        CHAR(1)      , /*PRINT AS IS CONTROL CHAR      */
     B1         CHAR(1)      , /*BREAK CHARACTERS              */
     B2         CHAR(1)      , /*BREAK CHARACTERS              */
     B3         CHAR(1)      , /*BREAK CHARACTERS              */
     B4         CHAR(1)      , /*BREAK CHARACTERS              */
     B5         CHAR(1)      , /*BREAK CHARACTERS              */
     B6         CHAR(1)      , /*BREAK CHARACTERS              */
     B7         CHAR(1)      , /*BREAK CHARACTERS              */
     B8         CHAR(1)      , /*BREAK CHARACTERS              */
     B9         CHAR(1)      , /*BREAK CHARACTERS              */
     MARKER     CHAR(1)      , /*UNDERLINE AND PAGE FAULT      */
     NUMCOL     FIXED BIN    , /*NUMBER OF COLS TO BE USED     */
     PGNUM      FIXED BIN    , /*PAGE NUMBER 0=PREV NUM        */
     PARNUM     FIXED BIN    , /*PARAGRAPH NUMBER 0=PREVIOUS   */
     FTNUM      FIXED BIN    , /*FOOTNOTE NUMBER 0=PREVIOUS    */
     MART       FIXED BIN    , /*MARGIN AT TOP IN LINES        */
     COL        FIXED BIN    , /*LEFT HAND MARGIN IN CHAR      */
     FTCOL      FIXED BIN    , /*LEFT MARGIN FOR FOOTNOTES     */
     MARB       FIXED BIN    , /*LINES OF MARGIN AT BOTTOM     */
     L          FIXED BIN    , /*LENGTH OF LINES TO BE PRINTED*/
     FTL        FIXED BIN    , /*FOOTNOTE LENGTH              */
     TAB(5)     FIXED BIN    , /*TAB VALUES                   */
     SPACE1     FIXED BIN    , /*INNER PARAGRAPH/FOOT SPACING  */
     SPACE2     FIXED BIN    , /*BETWEEN PARA/FOOT SPACING     */
     SPACE3     FIXED BIN    , /*SPACING AT START OF TABLE    */
     DBL        FIXED BIN    , /*BASIC UNIT FOR INDENTATION    */
     PTITLE     CHAR(5) VAR   , /*PARAGRAPH TITLE               */
     CTITLE     CHAR(100) VAR , /*PAGE TITLE                    */
     S1         BIT(1)       , /*TRUE=PAGE NUMBERING           */
     S2         BIT(1)       , /*TRUE=PARAGRAPH NUMBERING      */
     S3         BIT(1)       , /*TRUE=RIGHT JUSTIFY            */
     S4         BIT(1)       , /*TRUE=FOOTNOTE NUMBERING       */
     FOOT       CHAR(2000) VAR , /*UNPRINTED FOOTNOTES          */
     IN         CHAR(2000) VAR , /*UNPRINTED TEXT                */
     SW         BIT(1)       , /*UNDERLINE SWITCH FOR TEXT     */
     SF         BIT(1)       , /*UNDERLINE SWITCH FOR FOOT     */
     LINSTOP    FIXED BIN    , /*LINE NUM TO STOP PRINT CUR INP*/
     BUF        CHAR(2000) VAR , /*READ BUPPER                   */
     EXT,
     X CHAR(1) , (SX,SZ,SY) BIT(1) , (I,J) FIXED BIN;

```

```

DCL K FIXED BIN;
DCL NPG BIT(1);
/*THESE VARIABLES ARE USED WHEN THE NEXT ID PAR OR CEN OCCURS*/
DCL (CHAPC CHAR(1) , /*CONTROL FOR CHAPTER */
FTC CHAR(1) , /*FOOTNOTE CONTROL CHARACTER */
UNDERC CHAR(1) , /*UNDERLINE CONTROL CHAR */
TERMC CHAR(1) , /*TERMINATE CONTROL CHAR */
CENC CHAR(1) , /*CENTER CONTROL CHAR */
IDC CHAR(1) , /*INDENT CONTROL CHAR */
PARC CHAR(1) , /*PARAGRAPH CONTROL CHAR */
TABLE CHAR(1) , /*CONTROL CHAR FOR TABLES */
TABLN CHAR(1) , /*CONTROL FOR TABLE W NUMBER */
PAGC CHAR(1) , /*PAGE CONTROL CHAR */
PARMC CHAR(1) , /*READ PARAMETER CONTROL CHAR */
DR CHAR(1) , /* DRAIN CONTROL CHAR */
CTABC CHAR(1) , /*TAB CONTROL CHAR */
ASIS CHAR(1) , /*PRINT AS IS CONTROL CHAR */
COLNUM FIXED BIN , /*NUMBER OF COLS TO BE USED */
BMAR FIXED BIN , /*BOTTOM MARGIN */
LEN FIXED BIN , /*LENGTH OF LINE TO BE PRINTED */
LMAR FIXED BIN , /*LEFT HAND MARGIN */
LSPACE FIXED BIN , /*INNER PARA/FOOT SPACING */
PSPACE FIXED BIN , /*BETWEEN PARA/FOOT SPACING */
TSPACE FIXED BIN , /*SPACING AT START OF TABLE */
INDENT FIXED BIN , /*BASIC UNIT FOR INDENTATION */
TMAR FIXED BIN , /*TOP MARGIN */
PGTITLE CHAR(5) VAR , /*PARAGRAPH TITLE */
CHTITLE CHAR(100) VAR , /*CHAPTER TITLE */
TABA FIXED BIN , /*FIRST TAB LOCATION */
TABB FIXED BIN , /*SECOND TAB */
TABC FIXED BIN , /*THIRD TAB */
TABD FIXED BIN , /*FOURTH TAB */
TABE FIXED BIN , /*FIFTH TAB */
SA BIT(1) , /*TRUE=PAGE NUMBERING */
SB BIT(1) , /*TRUE=PARAGRAPH NUMBERING */
SC BIT(1) , /*TRUE=RIGHT JUSTIFY */
SD BIT(1) ) ; /*TRUE=FOOTNOTE NUMBERING */

OUTPUT1: PROC (A1,K1);
/*SIMPLE OUTPUT---OUTPUTS A1 AFTER SINGLE SPACE
AND STARTING IN COLUMN K1 */
DCL (A1,A2,A3,A4) CHAR(*) VAR,(K1,K2,K3,K4,I1,J1)
FIXED BIN;
PUT EDIT(A1) (SKIP,COL(K1),A);
RETURN;

OUTPUT2: ENTRY (A1,A2,K1);
/*UNDERLIN OUTPUT---NO SKIPS OUTPUTS A1 IN COL(1)
AND A2 IN COL(K1) */
PUT EDIT(A1,A2) (SKIP(0),A,SKIP(0),COL(K1),A);
RETURN;

OUTPUT3: ENTRY (A1,K1,A2,A3,K2,K3,K4);
/*PARAGRAPH TITLE AND NUMBERING OUTPUT--SINGLE SKIP
OUTPUTS A1 IN COL(K2), K1 IN NEXT K3 COLS A2 NEXT
THEN A3 AFTER SKIPPING K4 COLS*/
PUT EDIT(A1,K1,A2,A3) (SKIP,COL(K2),A,F(K3),A,
X(K4),A);
RETURN;

OUTPUT4: ENTRY (A1,A2,K1,K2);
/*UNDERLINE PARAGRAPH TITLE AND NUMBERING---
NO SKIPS A1 IN COL1 A2 IN COL(K1) +K2 SPACES */
PUT EDIT(A1,A2) (SKIP(0),A,SKIP(0),COL(K1),X(K2),A);

```



```

      RETURN;
OUTPUT5: ENTRY(A1,K1,K2);
/* PG TITLE AND NUMBER--NO SKIPS PRINTS A1 IN COL(K2) AND
   K1 IN NEXT 4 COLS */
      PUT EDIT(A1,K1) (SKIP(0),COL(K2),A,P(4));
      RETURN;
OUTPUT6: ENTRY(A1,A2);
/* DIAGNOSTIC OUTPUT---SINGLE SKIP OUTPUTS A1,A2 */
      PUT EDIT(A1,A2) (SKIP,A,A);
      RETURN;
EJECT:  ENTRY(A1);
/*NEW PAGE--EJECTS PAGE PRINTS A1 IN COL(1) */
      PUT EDIT(A1) (PAGE,A);
      RETURN;
DNSPACE: ENTRY(K1,K2);
/*SPACES DOWN K2-K1 SPACES */
      I1=K2-K1;
      IF I1<=0 THEN RETURN;
      DO J1=I1 TO 1 BY -1;
          PUT EDIT(' ') (SKIP,A);
      END;
      RETURN;
INPUT:  ENTRY(A1,A2,K1,A3,A4);
      ON ENDFILE BEGIN;
          A2=TERM; GO TO EXIT; END;
START:  GET EDIT(A2) (COL(1),A(80));
          A2=SUBSTR(A2,1,K1);
          IF SUBSTR(A2,1,1)=A3 THEN DO;
LOOP:    DO WHILE(SUBSTR(A2,LENGTH(A2),1)=' ');
              A2=SUBSTR(A2,1,LENGTH(A2)-1);
          END;
          A1=A1||A2;
          GET EDIT(A2) (COL(1),A(80));
          A2=SUBSTR(A2,1,K1);
          IF SUBSTR(A2,1,1)=' ' | SUBSTR(A2,1,1)=A3 THEN
              GO TO LOOP;
          END;
          IF SUBSTR(A2,1,1)=A4 THEN BEGIN;
              ON ERROR BEGIN;
                  PUT EDIT('CONVERSION ERROR ',ONSOURCE) (SKIP,A,A);
                  ONSOURCE='0';
              END;
              ON NAME PUT EDIT('NAME ERROR ',DATAFIELD) (SKIP,A,A);
              GET DATA;
              GO TO START;
          END;
EXIT:    RETURN;
      END OUTPUT1;
LINENUM: PROC RETURNS(FIXED BIN);
          DCL SYSPRINT FILE PRINT;
          RETURN(LINENO(SYSPRINT));
      END LINENUM;
INIT:    PROC;
/*INITIALIZES VARIABLES*/
          CHAP,CHAPC='*';
          PT,PTC='$';
          UNDER,UNDERC='_';
          TABLE,TABLEG='T';
          TABLN,TABLNG='S';
          TERM,TERMC='F';

```

```

CEN,CENC='C';
ID,IDC='I';
PAR,PARC='P';
PAGE,PAGC='X';
PARM,PARMC='D';
CTAB,CTABC='@';
DRA,DR='R';
ASI,ASIS='A';
B1=' ';
B2='.';
B3=',';
B4=':';
B5=': ';
B6='?';
B7='!'; /*EXCLAMATION POINT */
B8='-';
B9='_';
MARKER='.';
NUMCOL,COLNUM=80;
PGNUM=1;
PARNUM=1;
FTNUM=1;
MARB,BMAR=10;
PTL,L,LEN=50;
COL,LMAR=50;
SPACE1,LSPACE=1;
SPACE2,PSPACE=2;
SPACE3,TSPACE=2;
DBL,INDENT=5;
MART,THAR=10;
PTITLE,PGTITLE='';
CTITLE,CHTITLE='';
FTCOL=50;
TAB(1),TABA=5;
TAB(2),TABB=10;
TAB(3),TABC=15;
TAB(4),TABD=20;
TAB(5),TABE=25;
S1,SA='1'B;
S2,SB='1'B;
S3,SC='1'B;
S4,SD='1'B;
END IMIT;

```

ASSIGN: PROC;

/\*ASSIGNS NEW CONTROL VARIABLES\*/

```

CHAP=CHAPC;
FT=FTC;
UNDER=UNDERC;
TABLEG=TABLE;
TABLNG=TABLW;
TERM=TERMC;
CEN=CENC;
ID=IDC;
PAR=PARC;
PAGE=PAGC;
PARM=PARMC;
CTAB=CTABC;
ASI=ASIS;
DRA=DR;
NUMCOL=COLNUM;

```

```

MARB=BMAR;
L=LEN;
COL=LMAR;
SPACE1=LSPACE;
SPACE2=PSPACE;
SPACE3=TSPACE;
DBL=INDENT;
MART=THAR;
PTITLE=PGTITLE;
CTITLE=CHTITLE;
TAB(1)=TABA;
TAB(2)=TABB;
TAB(3)=TABC;
TAB(4)=TABD;
TAB(5)=TABE;
S1=SA;
S2=SB;
S3=SC;
S4=SD;
END ASSIGN;
INSERT_TAB: PROC (BUF);
    /*INSERTS AN UNUSED CHAR TO FORCE TAB CONTROL--CHANGED TO BLANK
    BEFORE PRINTING 12-0-9-2 */
    DCL BUF CHAR(2000) VAR, A CHAR(200) VAR, (I,J) FIXED BIN;
    DCL K FIXED BIN;
    DO I=1 TO 5;
        J=INDEX(BUF,CTAB);
        IF J=0 THEN RETURN;
        A='';
        DO K=1 TO TAB(I)-J;
            A=A||' '; /*12-0-9-2 */
        END;
        BUF=SUBSTR(BUF,1,J-1)||A||SUBSTR(BUF,J+1);
    END;
END INSERT_TAB;
NORM_HEAD: PROC (A) RETURNS (CHAR(2000) VAR);
    /*NORMALIZES HEAD OF A BY DELETING LEADING BLANKS */
    DCL (A,B) CHAR(2000) VAR;
    IF A='' | A=' ' THEN RETURN('');
    B=A;
    DO WHILE (SUBSTR(B,1,1)=' ' & LENGTH(B)>1);
        B=SUBSTR(B,2);
    END;
    RETURN(B);
END NORM_HEAD;
NORM_TAIL: PROC (A) RETURNS (CHAR(2000) VAR);
    /*NORMALIZES TAIL OF A BY DELETING TRAILING BLANKS LEAVING
    2 BLANKS AFTER .?! REMOVES - ELSE LEAVES 1 BLANK */
    DCL (A,B) CHAR(2000) VAR, X CHAR(1);
    IF A='' | A=' ' THEN RETURN('');
    B=A; B=DELETE(B);
    X=SUBSTR(B,LENGTH(B),1);
    IF X='.' | X='?' | X='!' THEN B=B||' ';
    ELSE IF X='-' THEN B=SUBSTR(B,1,LENGTH(B)-1);
    ELSE B=B||' ';
    RETURN(B);
END NORM_TAIL;
OUTPUT: PROC (A,L,COL,SPACE,LINSTOP,UNDER,SW,SW3) RETURNS (BIT(1));
    /*OUTPUTS A UNTIL LINSTOP OR A TO SHORT FALSE IF LINSTOP */
    /*SW IS UNDERLINE FROM PREVIOUS LINE, SW3 IS JUSTIFY */

```

```

DCL A CHAR(2000) VAR, (L,COL,SPACE,LINSTOP) FIXED BIN,
    UNDER CHAR(1), (SW,SW3) BIT(1);
DCL B CHAR(2000) VAR, S BIT(1);
LOOP:   IF LINENUM+SPACE>LINSTOP THEN RETURN('0'B);
        S=SELECT(A,B,L);
        IF ~S THEN RETURN('1'B);
        IF SW3 THEN B=JUSTIFY(B,L);
        CALL DRAIN(B,COL,SPACE,L,UNDER,SW);
        GO TO LOOP;
        END OUTPUT;
NEWLINE: PROC (FOOT,MARB,L,SPACE1,SPACE2);
/*CALCULATES NEW LINSTOP*/
DCL FOOT CHAR(2000) VAR, (MARB,I) FIXED BIN;
DCL (SPACE1,SPACE2) FIXED BIN;
DCL L FIXED BIN;
DCL JJ FIXED BIN;
    IF FOOT='' THEN JJ=0; ELSE
        IF INDEX(SUBSTR(FOOT,2),FT)=0 THEN JJ=1; ELSE JJ=2;
    I=59-MARB-(LENGTH(FOOT)/L)*SPACE1-JJ*SPACE2;
    RETURN(I);
    END NEWLINE;
DRAIN:  PROC (A,COL,SPACE,L,UNDER,SW);
/*PRINTS L CHAR OF A AND SETS A TO NULL */
DCL (A,B) CHAR(2000) VAR, (COL,SPACE,L,J) FIXED BIN,
    UNDER CHAR(1),SW BIT(1);
DCL I FIXED BIN;
    IF SUBSTR(A,1,1)=UNDER THEN DO;
        SW='1'B; A=SUBSTR(A,2);
        END;
    IF LENGTH(A)>L THEN A=SUBSTR(A,1,L);
/*REMOVE SPEC TAB CHAR 12-0-9-2 */
DO I=1 TO LENGTH(A);
    IF SUBSTR(A,I,1)=' ' THEN SUBSTR(A,I,1)=' ';
    END;
    CALL UNDERLINE(A,B,UNDER,SW);
    CALL DSPACE (1,SPACE);
    CALL OUTPUT1(A,COL);
    A='';
    IF B='' | B=' ' THEN RETURN;
    CALL OUTPUT2(MARKER,B,COL);
    RETURN;
    END DRAIN;
JUSTIFY: PROC (A,L) RETURNS(CHAR(2000) VAR);
/*JUSTIFIES A TO LENGTH L BY ADDING INTERNAL BLANKS */
/* OR L+1 IF FIRST CHAR IS UNDER */
DCL (A,B) CHAR(2000) VAR, (L,I,J) FIXED BIN;
DCL S BIT(1);
    B=DELETE(A);
    IF INDEX(B,' ')=0 THEN RETURN(A);
    IF SUBSTR(B,1,1)=UNDER THEN I=L+1-LENGTH(B);
    ELSE I=L-LENGTH(B);
    S='1'B;
LOOP:   DO J=LENGTH(B) TO 1 BY -1 WHILE (I>0);
        IF SUBSTR(B,J,1)=' ' & S THEN DO;
            S='0'B;
            I=I-1; B=SUBSTR(B,1,J)||' '||SUBSTR(B,J+1);
            END;
        IF SUBSTR(B,J,1)~=' ' THEN S='1'B;
        END;
    IF I>0 THEN GO TO LOOP;

```

```

        RETURN(B);
        END JUSTIFY;
SELECT: PROC(A,B,L) RETURNS(BIT(1));
/*SELECTS LINE FROM A DELETES THIS FROM A FALSE IF LEN(A)<L*/
/*BREAK CHARACTERS ARE B1 THRU B9 */
/*IF FIRST CHAR IS UNDER THEN SELECTS L+1 */
DCL (A,B) CHAR(2000) VAR,(L,I,K) FIXED BIN;
DCL X CHAR(1);
DCL LL FIXED BIN;
        IF LENGTH(A)=L THEN DO; B=A; A=''; RETURN('1'B); END;
        IF LENGTH(A)<L THEN RETURN('0'B);
        IF SUBSTR(A,1,1)=UNDER THEN LL=L+1; ELSE LL=L;
        K=LL+1;
LOOP:      DO I=K TO 1 BY -1;
        X=SUBSTR(A,I,1);
        IF X=B1 | X=B2 |X=B3 |X=B4 |X=B5 |X=B6 |X=B7 |X=B8 |X=B9
            THEN GO TO EXIT;
        END;
EXIT:      IF I<=L/2 THEN DO;
        B=SUBSTR(A,1,L-1)||'-';
        A=NORM_HEAD(SUBSTR(A,L));
        RETURN('1'B);
        END;
        IF SUBSTR(A,I,1)=' ' THEN B=SUBSTR(A,1,I-1);
        ELSE IF I=LL+1 THEN DO; K=LL; GO TO LOOP; END;
        ELSE B=SUBSTR(A,1,I);
        A=NORM_HEAD(SUBSTR(A,I+1));
        B=DELETE(B);
        RETURN('1'B);
        END SELECT;
NEW_PAGE: PROC(TITLE,MART,COL,SPACE,L,PGNUM,S1);
/*SETS UP NEW PAGE */
DCL TITLE CHAR(100) VAR, (MART,PGNUM,COL,SPACE,L,I) FIXED
BIN,S1 BIT(1);
PGNUM=PGNUM+1;
IF ~S1 THEN DO;
        CALL EJECT(MARKER);
        CALL DNSPACE(SPACE,MART);
        END;
ELSE DO;
        I=LENGTH(TITLE); I=COL+L-I-4;
        CALL EJECT(MARKER); CALL DNSPACE (SPACE,MART);
        CALL OUTPUT5(TITLE,PGNUM,I);
        END;
END NEW_PAGE;
UNDERLINE: PROC(A,B,UNDER,SW);
/*CALCULATES IN B THE UNDERLINE CONTROL WITH SW PREV LINE */
DCL (A,B) CHAR(2000) VAR,(I,J,K) FIXED BIN,UNDER CHAR(1),
SW BIT(1);
B='';
DO I=1 TO LENGTH(A); B=B||' '; END;
IF SW THEN DO; SW='0'B; I=0; GO TO LOOPA; END;
LOOP:      I=INDEX(A,UNDER);
IF I=0 THEN RETURN; ELSE SUBSTR(A,I,1)=' ';
LOOPA:     J=INDEX(A,UNDER);
IF J=0 THEN DO;
        J=LENGTH(A)+1; SW='1'B; END;
        ELSE SUBSTR(A,J,1)=' ';
DO K=I+1 TO J-1; SUBSTR(B,K,1)='_'; END;
GO TO LOOP;

```

```

      END UNDERLINE;
DELETE: PROC(A) RETURNS(CHAR(2000) VAR);
/*DELETES TRAILING BLANKS */
DCL (A,B) CHAR(2000) VAR;
  IF A=''|A=' ' THEN RETURN('');
  B=A;
  DO WHILE (SUBSTR(B,LENGTH(B),1)=' ' & LENGTH(B)>1);
    B=SUBSTR(B,1,LENGTH(B)-1);
  END;
  RETURN(B);
END DELETE;

CENTER: PROC(A,L,COL,SPACE1,SPACE2,LINSTOP,UNDER,SW) RETURNS(BIT(1));
/*PRINTS A CENTERED LINE FALSE IF LINSTOP */
DCL (A,B) CHAR(2000) VAR, (L,COL,SPACE1,SPACE2,LINSTOP,I,J)
  FIXED BIN, UNDER CHAR(1), (SW,SY,SX) BIT(1);
  SW='0'B; A=DELETE(A);
  IF LINENUM+(LENGTH(A)/L)*SPACE1+SPACE2>LINSTOP
    THEN RETURN('0'B);
  SY='0'B;
  A=SUBSTR(A,2);
  CALL DNSPACE (SPACE1,SPACE2);
LOOP:  IF A=''|A=' ' THEN RETURN('1'B);
  A=NORM_HEAD(DELETE(A));
  SX=SELECT(A,B,L);
  IF ~SX THEN DO; B=A; A=''; END;
  B=DELETE(B);
  I=COL+(L-LENGTH(B))/2;
  CALL DRAIN(B,I,SPACE1,L,UNDER,SY);
  GO TO LOOP;
END CENTER;

FOOT_OUT: PROC(FOOT,PTNUM,L,COL,SPACE1,SPACE2,LINSTOP,DBL,UNDER,
  FT,SF,S3,S4) RETURNS(BIT(1));
/*OUTPUTS FOOTNOTES TRUE IF LINSTOP ELSE FALSE */
DCL (FOOT,B,C) CHAR(2000) VAR, (L,COL,SPACE1,SPACE2,LINSTOP,
  DBL,I,PTNUM) FIXED BIN, (SX, SF,S3,S4) BIT(1), (UNDER,FT)
  CHAR(1), X CHAR(5) VAR;
  IF FOOT='' | FOOT=' ' THEN RETURN('0'B);
  IF SPACE1+SPACE2+LINENUM>LINSTOP THEN
    RETURN('1'B);
  B='';
  DO I=1 TO L; B=B||'-'; END;
  SX='0'B;
  CALL DRAIN(B,COL,SPACE1,L,UNDER,SX);
LOOP:  IF FOOT='' | FOOT=' ' THEN RETURN('0'B);
  IF LINENUM+SPACE1>LINSTOP THEN RETURN('1'B);
  IF SUBSTR(FOOT,1,1)=FT THEN DO;
  IF LINENUM+SPACE2>LINSTOP THEN RETURN('1'B);
    SF='0'B; SUBSTR(FOOT,1,1)=' '; SX='1'B; END;
    ELSE SX='0'B;
  I=INDEX(FOOT,FT);
  IF I=0 THEN DO; B=FOOT; FOOT=''; END;
    ELSE DO;
      B=SUBSTR(FOOT,1,I-1); FOOT=SUBSTR(FOOT,I);
    END;
  B=DELETE(B);
  IF SX THEN DO;
    X='';
    SX=NEW_PAR(C,B,PTNUM,L,COL,SPACE2,LINSTOP,DBL,X,UNDER,
      S4,S3,SF);
  END;

```

```

ELSE C=B;
SX=OUTPUT(C,L,COL,SPACE1,LINSTOP,UNDER,SP,S3);
IF -SX THEN DO;
    FOOT=C||FOOT; RETURN('1'B);
END;
IF C='' | C=' ' THEN GO TO LOOP;
CALL DRAIN(C,COL,SPACE1,L,UNDER,SP);
GO TO LOOP;
END FOOT_OUT;
NEW_PAR: PROC(IN,A,PARNUM,L,COL,SPACE,LINSTOP,DBL,PTITLE,UNDER,
S2,S3,SW) RETURNS(BIT(1));
/*PRINTS NEW PARAGRAPH FALSE IF LINSTOP LEAVES REMAINDER IN IN
*/
DCL (IN,A,B,C) CHAR(2000) VAR, (COL,SPACE,LINSTOP,DBL,L,
PARNUM,I,K) FIXED BIN, PTITLE CHAR(5) VAR, UNDER CHAR(1),
(S2,S3,SW,SWW,SX) BIT(1);
DCL KK FIXED BIN;
DCL JJ FIXED BIN;
SWW='1'B; GO TO START;
NEW_PARP: ENTRY(IN,A,PARNUM,L,COL,SPACE,LINSTOP,DBL,PTITLE,UNDER,
S2,S3,SW) RETURNS(BIT(1));
SWW='0'B;
START:
IF LINENUM+SPACE>LINSTOP THEN RETURN('0'B);
IF SUBSTR(A,2,1)>='0' & SUBSTR(A,2,1)<='9' THEN DO;
    I=SUBSTR(A,2,1); A=SUBSTR(A,3); END;
ELSE IF SUBSTR(A,2,1)='- ' & SUBSTR(A,3,1)>='0' &
SUBSTR(A,3,1)<='9' THEN DO;
    I=SUBSTR(A,2,2); A=SUBSTR(A,4);
END;
ELSE DO; I=1; A=SUBSTR(A,2); END;
IF SWW THEN A=DELETE(NORM_HEAD(A));
CALL INSERT_TAB(A);
K=L-DBL*I;
SX=SELECT(A,B,K);
IF -SX THEN DO; B=A; A=''; END;
B=DELETE(B);
IF S3 THEN B=JUSTIFY(B,K);
/*REMOVE SPEC TAB CHAR 12-0-9-2 */
DO KK=1 TO LENGTH(B);
IF SUBSTR(B,KK,1)=' ' THEN SUBSTR(B,KK,1)=' ';
END;
IF SUBSTR(B,1,1)=UNDER THEN DO; SW='1'B; B=SUBSTR(B,2);
END; ELSE SW='0'B;
CALL UNDERLINE(B,C,UNDER,SW);
IF S2 THEN DO;
    PARNUM=PARNUM+1;
    JJ=1; KK=10;
    DO WHILE (PARNUM>=KK); JJ=JJ+1; KK=KK*10; END;
    K=COL-(JJ+1+1+LENGTH(PTITLE));
    IF I<0 THEN K=K+I*DBL;
    IF K<=0 THEN K=1;
    CALL DMSPACE(1,SPACE);
    CALL OUTPUT3(PTITLE,PARNUM,'.',B,K,JJ,1+I*DBL);
    IF C-='' & C-=' ' THEN
        CALL OUTPUT4(MARKER,C,K,LENGTH(PTITLE)+JJ+1+1+
I*DBL);
    END;
ELSE DO;
    K=COL+I*DBL;

```

```

        IF K<=0 THEN K=1;
        CALL DNSPACE(1,SPACE);
        CALL OUTPUT1(B,K);
        IF C~='' & C~=' ' THEN
            CALL OUTPUT2(MARKER,C,K);
        END;

    IN=A;
    RETURN('1'B);
    END NEW_PAR;
DCL NORM_HEAD INTERNAL ENTRY (CHAR(2000) VAR) RETURNS (CHAR(2000)
VAR);
DCL NORM_TAIL INTERNAL ENTRY (CHAR(2000) VAR) RETURNS (CHAR(2000)
VAR);
DCL DELETE      INTERNAL ENTRY (CHAR(2000) VAR) RETURNS (CHAR(2000)
VAR);
DCL OUTPUT INTERNAL ENTRY (CHAR(2000) VAR, FIXED BIN, FIXED BIN,
FIXED BIN, FIXED BIN, CHAR(1), BIT(1), BIT(1)) RETURNS (BIT(1));
DCL NEWLINE INTERNAL ENTRY (CHAR(2000) VAR, FIXED BIN, FIXED BIN,
FIXED BIN, FIXED BIN) RETURNS (FIXED BIN);
DCL JUSTIFY INTERNAL ENTRY (CHAR(2000) VAR, FIXED BIN) RETURNS
(CHAR(2000) VAR);
DCL SELECT INTERNAL ENTRY (CHAR(2000) VAR, CHAR(2000) VAR, FIXED
BIN) RETURNS (BIT(1));
DCL CENTER INTERNAL ENTRY (CHAR(2000) VAR, FIXED BIN, FIXED
BIN, FIXED BIN, FIXED BIN, FIXED BIN, CHAR(1), BIT(1)) RETURNS
(BIT(1));
    DCL (NEW_PAR, NEW_PARP)
        INTERNAL ENTRY (CHAR(2000) VAR, CHAR(2000) VAR, FIXED
BIN, FIXED BIN, FIXED BIN, FIXED BIN, FIXED BIN,
FIXED BIN, CHAR(5) VAR, CHAR(1), BIT(1), BIT(1), BIT(1)) RETURNS
(BIT(1));
DCL FOOT_OUT INTERNAL ENTRY (CHAR(2000) VAR, FIXED BIN, FIXED BIN
, FIXED BIN, FIXED BIN, FIXED BIN, FIXED BIN,
FIXED BIN,
CHAR(1), CHAR(1), BIT(1), BIT(1), BIT(1)) RETURNS (BIT(1));
NEWCHAP: CALL INIT;
        FOOT=''; IN=''; SW='0'B; SP='0'B; I=PGNUM; J=PARNUM;
        K=FTNUM;
        CALL INPUT(FOOT, BUF, NUMCOL, FT, PARM); CALL ASSIGN;
        IF MART+SPACE1+SPACE2>59-MARB | MART+SPACE2+(NUMCOL/L)*
            SPACE1>59-MARB THEN GO TO SCAN;
        CALL EJECT(MARKER); CALL DNSPACE(SPACE2, MART);
        NPG='1'B;
        IF PGNUM=0 THEN PGNUM=I+1;
        IF PARNUM=0 THEN PARNUM=J; ELSE PARNUM=PARNUM-1;
DCL OUTPUT1 ENTRY (CHAR(*) VAR, FIXED BIN),
OUTPUT2 ENTRY (CHAR(*) VAR, CHAR(*) VAR, FIXED BIN),
OUTPUT3 ENTRY (CHAR(*) VAR, FIXED BIN, CHAR(*) VAR,
CHAR(*) VAR, FIXED BIN, FIXED BIN, FIXED BIN),
OUTPUT4 ENTRY (CHAR(*) VAR, CHAR(*) VAR, FIXED BIN,
FIXED BIN),
OUTPUT5 ENTRY (CHAR(*) VAR, FIXED BIN, FIXED BIN),
OUTPUT6 ENTRY (CHAR(*) VAR, CHAR(*) VAR),
EJECT ENTRY (CHAR(*) VAR),
DNSPACE ENTRY (FIXED BIN, FIXED BIN),
INPUT ENTRY (CHAR(*) VAR, CHAR(*) VAR, FIXED BIN,
CHAR(*) VAR, CHAR(*) VAR),
LINENUM ENTRY RETURNS (FIXED BIN);
        IF FTNUM=0 THEN FTNUM=K; ELSE FTNUM=FTNUM-1;
        GO TO LOOPX;

```



```

LOOP:      CALL INPUT(FOOT,BUF,NUMCOL,FT,PARM);
LOOPX:     X=SUBSTR(BUF,1,1);
           LINSTOP=NEWLINE(FOOT,HARB,FTL,SPACE1,SPACE2);
           SX=OUTPUT(IN,L,COL,SPACE1,LINSTOP,UNDER,SW,S3);
DUMP:      IF ~SX THEN DO;
           LINSTOP=59-HARB;
           SX=FOOT_OUT(FOOT,PTNUM,FTL,FTCOL,SPACE1,SPACE2,LINSTOP,
DBL,UNDER,FT,SP,S3,S4);
           CALL NEW_PAGE(CTITLE,HART,COL,SPACE2,L,PGNUM,S1);
           NPG='1'B;
           LINSTOP=NEWLINE(FOOT,HARB,FTL,SPACE1,SPACE2);
           END;
           IF X=' ' THEN DO;
           IN=NORM_TAIL(IN)||NORM_HEAD(DELETE(BUF));
           IF NPG THEN CALL DNSPACE(SPACE1,SPACE2);
           NPG='0'B;
           SX=OUTPUT(IN,L,COL,SPACE1,LINSTOP,UNDER,SW,S3);
           GO TO LOOP;
           END;
           IF X=CHAP | X=TERM THEN DO;
           IF NPG THEN CALL DNSPACE(SPACE1,SPACE2);
           NPG='0'B;
           IF IN~='' & IN~=' ' THEN CALL DRAIN(IN,COL,SPACE1,L,
UNDER,SW);
           LINSTOP=59-HARB;
           SY=FOOT_OUT(FOOT,PTNUM,FTL,FTCOL,SPACE1,SPACE2,LINSTOP,
DBL,UNDER,FT,SP,S3,S4);
           IF SY THEN DO;
           CALL NEW_PAGE(CTITLE,HART,COL,SPACE1,L,PGNUM,S1);
           GO TO LOOPA;
           END;
           CALL ASSIGN;
           IF X=CHAP THEN GO TO NEWCHAP; ELSE IF X=TERM THEN GO TO
TERMINATE;
           END;
           IF X = PAGE THEN DO;
           IF NPG THEN CALL DNSPACE(SPACE1,SPACE2);
           NPG='0'B;
           IF IN~='' & IN~=' ' THEN CALL DRAIN(IN,COL,SPACE1,L,
UNDER,SW);
           CALL ASSIGN;
           LINSTOP=59-HARB;
           SY=FOOT_OUT(FOOT,PTNUM,FTL,FTCOL,SPACE1,SPACE2,LINSTOP,
DBL,UNDER,FT,SP,S3,S4);
           CALL NEW_PAGE(CTITLE,HART,COL,SPACE1,L,PGNUM,S1);
           NPG='1'B;          GO TO LOOP;
           END;
           IF X=CEN THEN DO;
           IF IN~='' & IN~=' ' THEN DO;
           IF NPG THEN CALL DNSPACE(SPACE1,SPACE2);
           CALL DRAIN(IN,COL,SPACE1,L,UNDER,SW);
           END;
           CALL ASSIGN;
           NPG='0'B;
           SX=CENTER(BUF,L,COL,SPACE1,SPACE2,LINSTOP,UNDER,SW);
           IF ~SX THEN GO TO DUMP; ELSE GO TO LOOP;
           END;
           IF X=ID THEN DO;
           IF IN~='' & IN~=' ' THEN DO;
           IF NPG THEN CALL DNSPACE(SPACE1,SPACE2);

```

```

        CALL DRAIN(IN, COL, SPACE1, L, UNDER, SW);
        END;
    CALL ASSIGN;
    NPG='0'B;
    SY='0'B;
    SX=NEW_PAR(IN, BUF, PARNUM, L, COL, SPACE2, LINSTOP, DBL,
        PTITLE, UNDER, SY, S3, SW);
    IF ~SX THEN GO TO DUMP; ELSE GO TO LOOP;
    END;
IF X=PAR THEN DO;
    IF IN~='' & IN~=' ' THEN DO;
        IF NPG THEN CALL DNSPACE(SPACE1, SPACE2);
        CALL DRAIN(IN, COL, SPACE1, L, UNDER, SW);
        END;
    CALL ASSIGN;
    NPG='0'B;
    SX=NEW_PAR(IN, BUF, PARNUM, L, COL, SPACE2, LINSTOP, DBL,
        PTITLE, UNDER, S2, S3, SW);
    IF ~SX THEN GO TO DUMP; ELSE GO TO LOOP;
    END;
IF X=TABLEG THEN DO;
    IF IN~='' & IN~=' ' THEN DO;
        IF NPG THEN CALL DNSPACE(SPACE1, SPACE2);
        CALL DRAIN(IN, COL, SPACE1, L, UNDER, SW);
        END;
    CALL ASSIGN;
    NPG='0'B;
    SY='0'B;
    SX=NEW_PAR(IN, BUF, PARNUM, L, COL, SPACE3, LINSTOP, DBL,
        PTITLE, UNDER, SY, SY, SW);
    L=L-TAB(5)+1; COL=TAB(5)+COL-1;
    IF ~SX THEN GO TO DUMP; ELSE GO TO LOOP;
    END;
IF X=TABLEG THEN DO;
    IF IN~='' & IN~=' ' THEN DO;
        IF NPG THEN CALL DNSPACE(SPACE1, SPACE2);
        CALL DRAIN(IN, COL, SPACE1, L, UNDER, SW);
        END;
    CALL ASSIGN;
    NPG='0'B;
    SY='0'B;
    SX=NEW_PAR(IN, BUF, PARNUM, L, COL, SPACE3, LINSTOP, DBL,
        PTITLE, UNDER, S2, SY, SW);
    L=L-TAB(5)+1; COL=TAB(5)+COL-1;
    IF ~SX THEN GO TO DUMP; ELSE GO TO LOOP;
    END;
IF X=ASI THEN DO;
    IF IN~='' & IN~=' ' THEN DO;
        IF NPG THEN CALL DNSPACE(SPACE1, SPACE2);
        CALL DRAIN(IN, COL, SPACE1, L, UNDER, SW);
        END;
    NPG='0'B;
    SY='0'B;
    SZ='0'B;
    SX=NEW_PARP(IN, BUF, PARNUM, L, COL, SPACE1, LINSTOP, DBL,
        PTITLE, UNDER, SY, SZ, SW);
    IF ~SX THEN GO TO DUMP;
    IN='';
    GO TO LOOP;
    END;

```

```

IF X=DRA THEN DO;
  IF IN~=' ' & IN~=' ' THEN DO;
    IF WPG THEN CALL DNSPACE(SPACE1,SPACE2);
    CALL DRAIN(IN,COL,SPACE1,L,UNDER,SW);
    END;
  GO TO LOOP;
END;

```

```

CALL OUTPUT6('INCORRECT CONTROL WAS ',X);
GO TO LOOP;

```

SCAN:

```

X=' ';
CALL EJECT(MARKER);
CALL OUTPUT6('IMPROPER CONTROL EITHER NART+SPACE2',
  ' OR NART+(NUMCOL/L)*SPACE1+SPACE2 > 59-NARB ');
DO WHILE (X~=CHAP);
  GET EDIT(BUF) (A(80)); X=SUBSTR(BUF,1,1);
  END;
GO TO NEWCHAP;

```

TERMINATE:

```

CALL EJECT(MARKER);
CALL OUTPUT6('NORMAL TERMINATION','');
END PRINT;

```

## APPENDIX B

## Sample Input for PRINT

```

D
BMAR=5  THAR=5  B8=';'  LEN=60  PTL=60  SC='0'B  SD='0'B
CHTITLE='PRINT'  TABA=20  TABB=25  TABE=25  TABC=45 ;
IO@@@CGTM 128
IO@@@August 1971
C _PRINT - A Text Formatting Program_
C James E. George
D
PSPACE=1 ;
C Stanford Linear Accelerator Center
C Computation Group
C Stanford, California
D
PSPACE=2 ;
POABSTRACT
I PRINT is a text formatting program written in PL/I to produce
documents using the printer. The input to Print contains the text to
be printed interspersed with the necessary control information to
generate the desired format of the text. The features supported by
PRINT are:
T2Automatic or manual paging
D
TSPACE=1 ;
T2Page numbering with or without titling
T2Paragraphing with or without indentation (left or right), numbering
and/or titling
T2Underlining
T2Centering
T2Tables with or without numbering
T2Print direct image (i.e. print text as is )
T2Right justification of the text
T2Tabs
D
TSPACE=2 ;
IOAlso, the margin, line length and spacings between lines are variable.
The aim was to provide a modular system which would be easy to change and
would allow the value of any control variable to be changed by input
control.
POINTRODUCTION
I The use of computer programs to format text to a final form for
publication has been discussed in several documents (Meadow 1970;
van Dam and Rice 1970; Berns 1968 and 1969). Basically, the problem is
to take a linear string input and convert it to a multi-sheet two-dimensional
document. Further, the final document is aesthetically good or bad by some
undefinable human judgement. A really good program does not currently
exist. PRINT represents a useable basic system for the preparation
of a limited class of documents of interest to the author.
I Initially, I became interested in text formatting because I wanted
to maintain a bibliography using a computer text editing system; I
was willing to maintain the entries in alphabetic order, but wanted
to be able to number the entries and vary the formatting on output
(e.g. bibliography vs. working form for constructing an annotated
bibliography). Although TEXT 360 (----1968) existed at the time,
it was not locally available; FORMAT (Berns 1968 and 1969) was
available* but did not have the numbering feature. Hence, PRINT

```

was originally developed to provide an automatic numbering for successive entries for the preparation of bibliographies; I also decided to develop it as a personal document formatting system which I could easily experiment with and modify for my own purposes.

\$0\*The version of FORMAT locally available has been extensively modified by John Ehrman (Ehrman-Berns 1971) and was used to prepare Dave Gries's book (Gries 1971).

I Several documents have been produced using PRINT (George 1969a and b; George and Hoffman 1969), the library at SLAC has used it to produce a serials list (Section 5.2), a two-dimensional expression capability has been illustrated with PRINT (George 1971) and the cost of using it has been studied (George and Hoffman 1969).

#### POGENERAL DESCRIPTION

I An output document from PRINT consists of chapters, pages (with/without page title and /or page numbering), paragraphs (with/without paragraph titles and/or paragraph numbering), footnotes (with/without footnote numbering), tables (with/without numbering), centered text, unaltered text, underlined text, text whose position is controlled by tabs and text which may be right justified\*\*. In order to form these output units, the input must be identified as these units.

\$0\*\*A good discussion of justification is given in Meadow (Meadow 1970, pgs. 253-256).

I An input unit is denoted by a control character in column 1; a blank in column 1 generally means a continuation of the previous classification. Using these input units, PRINT maintains two buffers from which it composes the output document one line at a time. One buffer contains the text input, excluding footnotes, which has not been used (this buffer will be called the input buffer) and the other buffer contains the unused footnotes (this buffer will be called the footnote buffer).

I PRINT's normal cycle is to select and print full lines from the input buffer; when the input buffer contains less than one full line, more input is requested from an input section. Footnotes interrupt the normal cycle whenever just enough room is left at the bottom of the current page to print the current contents of the footnote buffer. Full lines are selected by examining the buffer (input or footnote) starting at the current line length to be printed and scanning to the left for the first occurrence of a break character.

I Footnotes are a problem in this and many other systems (e.g. see the discussion in Meadow 1970, pgs. 259-260). The problem is to at least start the footnote on the page where it is referenced. In PRINT, footnotes and references to footnotes are unrelated; a footnote is printed at the bottom of a page which usually has just enough room for the footnote. Generally, the footnotes occur on the same page as the reference, however it is possible for the footnote to appear before or after this page; this can be altered by the manual paging control or the drain buffer control.

I The input function of PRINT separates the single input stream into footnotes, text and format control. The input continues processing until a non-footnote or non-format control is obtained.

I The beginning of a footnote unit is indicated by the footnote control character occurring in column 1; all succeeding cards which have a blank in column 1 are part of the footnote and are saved in the footnote buffer. A footnote is terminated by the occurrence of a non-blank control character in column 1, other than the footnote control character.

I The beginning of a format control change is indicated by the occurrence of the read new parameter control character in column 1. Upon recognition of this control character, the input function of PRINT executes a GET DATA statement in PL/I thus allowing the value of any variable defined to be

altered; usually these are the parameters controlling the formatting of the text.

I In most cases, PRINT has two copies of each variable; one copy is used for format control and the value of the second copy is assigned to the first copy upon recognition of a text unit. Hence, format control is specified before the text item appears and is assigned when the beginning of this text unit is recognized.

#### POCONTROL AND VARIABLE DESCRIPTIONS

I The text control tokens must occur in column 1 with the beginning of the text on the same card. These tokens are:

TO_NAMES_*	INITIAL VALUE	EXPLANATION
------------	---------------	-------------

\$0 The name of the second copy of the variable is given followed by the name of the first copy in parenthesis throughout this section; all control tokens are of type CHAR(1).

TOPARC(PAR)@P@Indicates start of a new paragraph with numbering.

TOIDC(ID)@I@Indicates start of a new paragraph without numbering.

TOTABLE(TABLEG)@T@Indicates start of a table without numbering.

TOTABLEN(TABLENG)@S@Indicates start of a table with numbering.

TOCENC(CEN)@C@Indicates that the text on this card is to be centered.

TOASIS(ASI)@A@Indicates that the text on this card is to be printed as is.

I All the text controls above may involve indentations except centering; the indentation is in terms of a multiplier of the basic indentation unit. For example,

I2I2Start of a paragraph

I0would start a new paragraph whose first line would be indented 2 times the basic unit to the right, and

I2I-2Start of a paragraph

I0would start a new paragraph whose first line would be indented 2 times the basic unit to the left.

I The table option functions like the paragraph for the first line and then prints subsequent lines starting at the fifth tab position.

I The footnote control character is (also type CHAR(1)):

TO_NAMES_	INITIAL VALUE	EXPLANATION
-----------	---------------	-------------

TOFTC(FT)@\$@Indicates the beginning of a footnote.

I0Like the paragraph control, a single digit or a minus sign and a digit may immediately follow the footnote control to indicate the desired indentation.

I The controls which may occur anywhere in any text are (also of type CHAR(1)):

TO_NAMES_	INITIAL VALUE	EXPLANATION
-----------	---------------	-------------

TOUNDERC(UNDER)@\_@Indicates the beginning or end of the underlining; all text bracketed by underlines will be underlined.

Should an error occur, the underline will be turned off by the next occurrence of a non-blank control character in column 1.

D

CTABC='/' ;

TOCTABC(CTAB)/@/Indicates a tab position.

D

CTABC='@' ;

I The format control tokens must also occur in column 1 and any text contained on the card is lost. The controls are of type CHAR(1) and are:

TO_NAMES_	INITIAL VALUE	EXPLANATION
-----------	---------------	-------------

TOTERM(TERM)@P@Terminate PRINT.

TOPAGC(PAGE)@X@Drain the input buffer and start a new page.

TODR(DRA)@R@Drain the input buffer.

TOPARMC(PARM)@D@The following cards until an unquoted ';' contain variable names and values to read with PL/I's GET DATA statement.

TOCHAP(CHAPC)@#@Drain all buffers and initialize all variables.  
 I The variables used during the printing are:

D

TABA=18 TABB=35 TABC=45 TABE=45;

TO\_NAMES\_@\_TYPE\_ \_INITIAL VALUE\_@ \_EXPLANATION\_

TOCOLNUM(NUMCOL)@FIXED BIN@80@Number of columns of the input card to be used.

TOTMAR(MARB)@FIXED BIN@10@Lines of margin at the bottom of a page.

TOLEN(L)@FIXED BIN@50@Length of line to be printed.

TOLMAR(COL)@FIXED BIN@50@Width of left margin in characters.

TOLSPACE(SPACE1)@FIXED BIN@1@Inner paragraph, footnote and table spacing.

TOSPACE(SPACE2)@FIXED BIN@2@Spacing at the beginning of a paragraph or footnote.

TOTSPACE(SPACE3)@FIXED BIN@2@Spacing at the beginning of a table.

TODENT(DBL)@FIXED BIN@5@Basic unit for indentation.

TOTMAR(MART)@FIXED BIN@10@Lines of margin at the top of a page.

TOTABA(TAB(1))@FIXED BIN@5@Character position from left margin for first tab position.

TOTABB(TAB(2))@FIXED BIN@10@ " " " for second tab position.

TOTABC(TAB(3))@FIXED BIN@15@ " " " for third tab position.

TOTABD(TAB(4))@FIXED BIN@20@ " " " for fourth tab position.

TOTABD(TAB(5))@FIXED BIN@25@ " " " for fifth tab position.

TO----- (PGNUM)@FIXED BIN@1@Current page number; for new chapter  
 '0' implies previous numbering.

TO----- (PARNUM)@FIXED BIN@1@Current paragraph number; for new chapter  
 '0' implies previous numbering.

TO----- (PTNUM)@FIXED BIN@1@Current footnote number; for new chapter  
 '0' implies previous numbering.

TO----- (FTCOL)@FIXED BIN@50@Left margin for footnotes.

TO----- (FTL)@FIXED BIN@50@Line length for footnotes.

TOSA(S1)@BIT(1)@TRUE@Number pages if true.

TOSB(S2)@BIT(1)@TRUE@Number paragraphs if true.

TOSC(S3)@BIT(1)@TRUE@Right justify if true.

TOSD(S4)@BIT(1)@TRUE@Number footnotes if true.

TOPGTITLE(PTITLE)@CHAR(5) VAR@NULL@Paragraph title.

TOCHTITLE(CTITLE)@CHAR(100) VAR@NULL@Chapter title.

I The break characters which determine how the input is broken into lines are (all of type CHAR(1)):

T \_NAMES\_@\_INITIAL VALUE\_

T B1@ (blank)

T B2@. (period)

T B3@, (comma)

T B4@; (semicolon)

T B5@: (colon)

T B6@?

T B7@! (exclamation point)

T B8@- (dash)

T B9@\_ \_ (underline)

I Also, a character (variable MARKER) is printed in character position 1 of a line containing underlining or page faults; this allows the local version of a text editor (WYLBUR) to output the text to a typewriter.

I A complete listing of PRINT is given in Appendix A.

X

#### POEXAMPLES

I Some examples were indicated in the Introduction. Some short examples will be given here.

D

PARNUM=0 PGTITLE='5.' LEN=80 UNDERC='/';

POBibliography example

I The input

AOP0George, James E. (19691). The System Specification of GLAF: A Linear  
A0 String Graphical Language Facility. Stanford Linear Accelerator  
A0 Center, Computation Group, GSG 61.

AOP0Meadow, Charles T. (1970). Editing Text, Chapter 9 in \_Man-Machine  
A0 Communication\_, Wiley, 244-278.

I0produces the output

D

PARNUM=0 PGTITLE='' LEN=60 UNDERC='\_';

POGeorge, James E. (19691). The System Specification of GLAF: A Linear  
String Graphical Language Facility. Stanford Linear Accelerator  
Center, Computation Group, GSG 61.

POMeadow, Charles T. (1970). Editing Text, Chapter 9 in \_Man-Machine  
Communication\_, Wiley, 244-278.

X

D

PARNUM=1 PGTITLE='5.' LEN=80 CTABC='/' UNDERC='/';

POSerials example

I The input

A0D

A0 PSPACE=2 TSPACE=1 SC='0'B TABE=1 CTABC='@'

A0TABA=5 TABB=30 TABC=45 TABD=53 LEN=70 LMAR=50 BMAR=5 THAR=5 LSPACE=1;

A0C\_S L A C LIBRARY SERIALS LIST\_

A0T0@@@FEBRUARY 1971

A0I-1A C M

A0 see--Association for Computing Machinery

A0I-1A C M ARTIFICIAL INTELLIGENCE (SIGART)

A0T0@on order @@Central Lab Library

A0I-1A C M COMMUNICATIONS (SIGCOMM)

A0T0@v.1- @1970- @Central Lab Library

A0I-1A C M COMPUTER GRAPHICS (SIGGRAPH)

A0T0@v.5- @1971- @Central Lab Library

A0I-1A C M LANGUAGE ANALYSIS AND STUDIES IN THE HUMANITIES (SICLASH)

A0T0@v.2- @1969- @Central Lab Library

A0I-1A C M MICROPROGRAMMING (SIGNICRO)

A0T0@on order @@Central Lab Library

I0produces the output

D

PSPACE=2 TSPACE=1 SC='0'B TABE=1 CTABC='@' UNDERC='\_';

TABA=5 TABB=30 TABC=45 TABD=53 LEN=70 LMAR=50 BMAR=5 THAR=5 LSPACE=1;

C\_S L A C LIBRARY SERIALS LIST\_

T0@@@FEBRUARY 1971

I-1A C M

see--Association for Computing Machinery

I-1A C M ARTIFICIAL INTELLIGENCE (SIGART)

T0@on order @@Central Lab Library

I-1A C M COMMUNICATIONS (SIGCOMM)

T0@v.1- @1970- @Central Lab Library

I-1A C M COMPUTER GRAPHICS (SIGGRAPH)

T0@v.5- @1971- @Central Lab Library

I-1A C M LANGUAGE ANALYSIS AND STUDIES IN THE HUMANITIES (SICLASH)

T0@v.2- @1969- @Central Lab Library

I-1A C M MICROPROGRAMMING (SIGNICRO)

T0@on order @@Central Lab Library

D

PARNUM=2 PGTITLE='5.' LEN=60;

POThis paper

I The input which produces this paper is given in Appendix B.

D

PARNUM=0 PGTITLE='';

X



## C BIBLIOGRAPHY

- POBerns, G.M. (1968). Some Design Considerations for a Fast, General Purpose Text Processor, and the FORMAT Program. IBM Data Processing Division, Washington Scientific Center, 320-3503.
- POBerns, Gerald M. (1969). Description of FORMAT, a Text-Processing Program. CACM, 12, 3(March), 141-146.
- POEhrman, John R. and Berns, Gerald M. (1971). FORMAT, A Text Processing Program. Stanford Linear Accelerator Center, SLAC-135.
- POGeorge, James E. (1969a). The System Specification of GLAF: A Linear String Graphical Language Facility. Stanford Linear Accelerator Center, Computation Group, GSG 61.
- POGeorge, J.E. (1969b). Rules for Transforming a Grammar to a Simple Precedence Grammar Utilizing Artificial Productions. Stanford Linear Accelerator Center, Computation Group, GSG 62.
- POGeorge, James E. (1971). GEMS-A Graphical Experimental Meta System. Stanford Linear Accelerator Center, SLAC-134.
- POGeorge, James E. and Hoffman, Lance J. (1969). A Comparison of Manual and Computer-Based Text Editing. Stanford Linear Accelerator Center, Computation Group, CGTM 69.
- POGries, David (1971). \_Compiler Construction for Digital Computers\_, Wiley.
- POMeadow, Charles T. (1970). Editing Text, Chapter 9 in \_Man-Machine Communication\_, Wiley, 244-278.
- POvan Dam, Andries and Rice, David E. (1970). Computers and Publishing: Writing, Editing and Printing. in \_Advances in Computers\_, 10, 145-174.
- PO----- (1968). TEXT 360 Reference Manual and Operating Guide. IBM Corporation, New York.

F