219
9-27

DR-1926

SLAC-135

MASTER

FORMAT, A Text Processing Program

John R. Ehrman

Gerald M. Berns

SLAC Report No. 135

July 1971

AEC Contract AT(04-3)-515

STANFORD LINEAR ACCELERATOR CENTER

Stanford University • Stanford, California

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SLAC-135
UC-32
(MISC)

FORMAT, A TEXT PROCESSING PROGRAM

JOHN R. EHRMAN
COMPUTATION GROUP
STANFORD LINEAR ACCELERATOR CENTER
STANFORD UNIVERSITY
Stanford, California

GERALD M. BERNS*

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

PREPARED FOR THE U. S. ATOMIC ENERGY
COMMISSION UNDER CONTRACT NO. AT(04-3)-515

July 1971

Printed in the United States of America. Available from National Technical Information Service, U. S. Department of Commerce, 5285 Port Royal Road, Springfield, Virginia 22151. Price: Printed copy \$3.00; Microfiche \$0.95.

* Formerly Staff Member of IBM Washington Scientific Center.

The FORMAT text-processing program was written by G. M. Berns, an employee of the IBM Washington Scientific Center. It was made available to IBM System/360 users as a Type III program (number 360D-06.0.003) available from IBM's Program Information Department.

The program was modified and elaborated at SLAC by J. R. Ehrman, with the help and advice of Mr. Berns. The current version of the program (Release 5) is available from

COSMIC
Barrow Hall
University of Georgia
Athens, Georgia 30601

The FORMAT Manual	i
FORMAT Manual Input	69
FORMAT Source Program	99

The FORMAT Manual

Gerald M. Berns

Release 5

Modifications and Additions by

**John R. Ehrman
Computation Group
Stanford Linear Accelerator Center
Stanford, California 94305**

July 1971

FORMAT --- A Text Processing Program

CONTENTS

I.	Summary of Facilities	1
II.	Introduction	3
III.	Control Cards	9
	Examples of Correct Control Cards	23
	Examples of Faulty Control Cards	23
IV.	Command Operands and Command Words	24
	Summary of Command Operands	28
	Examples of Command Words	29
V.	Special Operands for Capitalization and Special Characters, and the Non-Trivial Blank	31
	Examples of the Usage of Special Operands	34
VI.	The Editor Facility	35
	Examples of Editor Control Card Groups	43
VII.	Rules for Using FORMAT	44
VIII.	Summary of FORMAT Control Cards and Command Operands ..	46
IX.	Datasets Used by FORMAT	50
X.	Description of FORMAT for OS/360, and Suggested Control Cards	53
XI.	Hints and Suggestions	55
XII.	Error Handling and Diagnostic Messages	57
XIII.	Appendix	61
	Index	
	Control Cards	
	Diagnostics	

I. Summary of Facilities

FORMAT is a program for System/360 and System/370 designed to meet the need for a rapid method of editing and producing papers, reports, and other finished and reproducible documents directly on the system printer, using upper and lower case and special characters. It has facilities which simplify the task of index construction. Input to the program is free-form card-image text. The document is formatted and controlled according to control cards and Command Words interspersed throughout the input. FORMAT is a single program requiring no auxiliary programs for its operation.

Via entirely free-form control cards the user may specify:

- Automatic capitalization of all sentences
- Number of text columns per page
- Width of text columns
- Number of lines per page
- Number of print positions between text columns
- Page numbering and first page number (or no numbering)
- Location of page number on the right, the left, or alternating
- Number of print positions for paragraph indentation
- Number of print positions for column indentation
- Line spacing (single spacing, double spacing, etc.)
- Number of lines between paragraphs
- Right-justification of text (or not)
- Tab settings
- Extent of card field from which input is to be read
- Printing of title on every page (or not)
- Position of the title
- Position of the text
- Position of the footer
- Sentences separated by a minimum of 1 or 2 blanks
- Kind of keypunch used
- Upper and lower case output (or all upper case)
- Number of copies of document
- Creation of condensed input tape from card deck
- Editing of input master tape
- Listing and/or punching of input dataset
- Tape input
- Tape output
- Printing of output master tape
- Merging and/or joining of input tapes
- Production of an alphabetized list of all significant words in the document, with a count of each
- That certain words, phrases, or strings be located

- That specific characters are to be left in the spaces skipped over when tabulating to new column positions
- That a particular special character should be recognized as requesting overprinting
- That a page should be made darker by printing each line more than once, on top of itself
- That a particular special character should be recognized as a non-eliminatable blank
- That underlining should or should not begin and end under punctuation characters
- That non-eliminatable blanks should or should not be considered when centering and underlining text

FORMAT does not provide facilities for automatic hyphenation, for automatic production of a table of contents, or for footnotes; page numbers appear only at the top of the page.

Commands embedded within the text (called Command Words) provide the capability to start a new line, paragraph, column, and page; to tabulate leaving blanks, dots, or any other character in the spaces skipped over; to underline (and to stop); to read groups of control cards; to center text within a column-line (and to stop); to print text "as is" (and to stop); to print text in upper case (and to stop); to print text with each word capitalized (and to stop); to indent (immediate or delayed) either or both column margins (and to restore the column format); to keep the next *n* lines in the same text column; and to keep text of unspecified length in the same text column.

FORMAT requires a minimum memory size of 64K in a standard System/360. No additional devices are required beyond those necessary to operate OS/360; however, the availability to the program of magnetic tape drives greatly enhances its usefulness, especially if the Editor facility is to be used with any regularity. FORMAT is written entirely in full Fortran IV and requires the full Fortran library. The System Input dataset (from which FORMAT reads its card input), the System Output dataset (on which FORMAT prints the document and other materials), and the System Punch dataset (which is used for punching a condensed form of the input deck), are defined as Fortran dataset reference numbers 5, 6, and 7, respectively.

The normal output mode is upper and lower case. Means are provided to allow the user to specify upper case only, and special characters. FORMAT produces its normal output for the TN print train, and has facilities for printing all of the 120 possible characters. Note that no subscripts are provided by the TN print train, nor, therefore, by FORMAT.

II. Introduction

Before discussing how FORMAT produces a document, we will define and illustrate some terms and notation. The figure below represents a typical page of text; we will refer to it throughout this introduction.

12	Title
This is the beginning of a paragraph; the size of the indent at the start of the paragraph may be specified on a control card.	
Now,	
this material begins a new column-line: that is, it starts a new line within the current column of text material.	
This text material illustrates the use of an indent: the right margin has been indented an additional 10 spaces.	
This text material illustrates the use of a hanging, or delayed, indent: the text is not indented until the line following the first line of text.	
This text material is centered!	
Footer	

In the above example page, the page number is at the upper left corner; the title (which may occupy more than one line) is at the top of the page; the footer (or footing title, or running foot) is at the bottom of the page; the text material consists of a single column 52 print positions wide; the hanging text was

indented 7 spaces on the left and 8 spaces on the right; the title is separated from the text by 3 blank lines; the footer is separated from the last text line by 3 blank lines; and the indent at the start of a paragraph is 5 spaces.

FORMAT produces a document by reading control cards and text. The text is arranged on the output page in a format determined by the control cards. FORMAT reads its input in one of two phases: an edit phase and a document phase. (There may be either (1) only a document phase, or (2) an edit phase followed by a document phase. We will discuss the edit phase in Section VI.) In each of these phases, FORMAT reads its input in different modes. In the document phase, FORMAT reads its input cards in one of three modes: control card mode, normal text mode, and "as-is" text mode. (As-is text mode will be discussed in Section IV.)

To start the document phase, FORMAT begins by reading its input in control card mode. Because most of the page layout control variables have been preset to "average" values (such as 59 lines per page, 64 characters per line in a single column, etc.), the only control card needed initially is the one that signals the end of a group of control cards, and causes FORMAT to switch to normal text mode: the "GO" control card.

In text mode, FORMAT reads the input text and arranges it in the desired layout on an internal "image" of the page to be printed. As each page image is filled, it is sent to the printer. If any errors are detected, FORMAT makes a note of each, and will print a list of diagnostic messages describing the error at the end of the job. For most errors, FORMAT will assign default values to the erroneous variables, or take default actions for erroneous commands.

While in normal text mode, the user will normally wish to specify actions such as "begin a new paragraph", "skip to a new line", "indent the margins", and so forth. These actions are requested with Command Words, which may appear anywhere in the input text. They are not printed by FORMAT, but cause it to take the specified actions instead (unless they are incorrectly given and therefore cause an error).

FORMAT detects the start of a Command Word by finding an escape character: a right parenthesis which is preceded by one or more blanks (spaces), and followed by one or more non-blank characters. The characters which follow the right parenthesis are called Command Operands, and they specify what actions FORMAT should take. A Command Word is ended by one or more blanks. (Even though it is very unlikely that the input text

will contain a string of characters starting with a right parenthesis, FORMAT provides the Special Operands (described in Section V) which allow us to print such a string if it is desired. Thus the choice of the right parenthesis as the "escape character" is not a limitation on the user.)

To illustrate, the Command Operand which requests the start of a new paragraph is the letter "P". Thus, if the input text contained the Command Word ")P ", the following text material would begin a new paragraph. Another commonly used Command Operand is "L", which has an effect similar to the action caused by striking the "return" key on a typewriter: the end of the current line is signaled, and the carriage is positioned at the start of the following line. Thus, the Command Word ")LL " would cause the line of text in which it appears to terminate, and the following line to be skipped. This example of a command word contains two Command Operands, "L" and "L"; this shows how Command Operands are grouped to form Command Words.

We will now look at a simple example of FORMAT input: suppose we wish to print the first part of the text material shown in the figure. The input text could be prepared as follows:

```
)P THIS IS          THE BEGINNING OF A PARAGRAPH; THE
                        SIZE OF THE INDENT AT THE START      OF THE
PARAGRAPH MAY BE SPECIFIED ON A CONTROL CARD. )L NOW,    )L
THIS
MATERIAL BEGINS A NEW COLUMN-LINE: THAT IS, IT          STARTS
A NEW LINE WITHIN THE CURRENT COLUMN OF TEXT MATERIAL. )LLL
```

Several important points are illustrated in this example. First, the input to FORMAT is entirely free-form: the user may leave as many spaces between input words as he likes, and FORMAT will ignore the excess blanks as it collects words to be placed in the page image. Second, there is no need to start a new input line when a new output line is desired; the "L" Command Operand will start a new line on the output page.

As the input cards are read by FORMAT, it may be necessary to change some of the control variables which determine the arrangement of the text on the page. For example, the user may want to change from one column per page to two (as was done to produce the index for this manual). To go from normal text mode back to control card mode, a Command Word is placed in the input stream which ends with the Command Operand "V". The rest of the card following the "V" is ignored, and FORMAT begins reading

control cards with the next input card. Thus, the user can dynamically modify the layout of the text on the page, and can change the values of the control variables. As before, the end of the control card group is signaled by a "GO" control card.

To illustrate, suppose we wish to set the margin indents to be zero spaces at the left and ten spaces at the right, as in the second portion of the figure above. The necessary input could be prepared as follows:

```

      )V
INDENTATION OF THE COLUMN IS (0,10) POSITIONS
GO
)I #THIS TEXT MATERIAL ILLUSTRATES
THE USE OF AN INDENT: THE RIGHT MARGIN HAS BEEN INDENTED AN
ADDITIONAL 10 SPACES.      )ILLL

```

In this example, the "I" Command Operand was used to control indentation of the margins. Its operation is like that of an "on-off" switch: each appearance of the "I" Command Operand causes indentation to begin (if it was not already in effect) or to end (if it was in effect). It is not like the "L" Command Operand, which causes a new line each time it appears; "I" does not cause additional indentation each time it appears, but turns the indentation on or off. The "#" sign preceding the word "THIS" is called a Special Operand. It causes the immediately following letter (the "T") to be capitalized in the printed output. Special Operands will be discussed in Section V.

We observe that the next segment of text material in the figure also requires an indentation. The control card which allows us to set the amount of indentation (the "INDENTATION OF THE COLUMN" control card) can specify up to seven different indentations. Thus, rather than prepare another control card, we will go back and change the previous input material so that it will control both of the indented segments of text in the figure. The use of the "H" Command Operand will be explained shortly.

```

      )V
INDENTATION OF THE COLUMN IS (0,10), (7,8) POSITIONS
GO
)I #THIS TEXT MATERIAL ILLUSTRATES
THE USE OF AN INDENT: THE RIGHT MARGIN HAS BEEN INDENTED AN
ADDITIONAL 10 SPACES.      )ILLLH2 THIS TEXT MATERIAL
ILLUSTRATES THE USE OF A HANGING, OR DELAYED, INDENT: THE
TEXT IS NOT INDENTED UNTIL THE LINE FOLLOWING THE FIRST
LINE OF TEXT. )H2LLL

```

As noted above for "I", the "H" Command Operand works like an on-off switch. An additional feature illustrated in the above example is the "2" following the "H" Command Operand, which means that the second pair of column indentations is to be used in determining the number of positions to indent. Several other Command Operands may be followed by a number; they are discussed in Section IV.

The last text segment in the figure could be prepared as follows:

```
)M THIS TEXT )L MATERIAL )L IS )L    CENTERED! )M
```

The "M" Command Operand causes centering of the printed text to begin or end. It is like the "I" and "H" Command Operands in being like an on-off switch, but "M" does not depend on a control card to determine the amounts of spacing to be performed.

FORMAT determines that the end of the input has been reached when it detects a Command Word ending with the Command Operand "E". (This means, of course, that the end of the input should occur in normal text mode, not in "as-is" text mode or in control card mode.) FORMAT then prints the final text page, followed by a list of all control cards read, and the diagnostics (if any). At this point, FORMAT will re-initialize itself to read a fresh job, starting to read in control card mode just as it did at the very beginning. Thus, multiple documents may be produced in a single computer run.

To illustrate a complete FORMAT job, we will now give a complete set of input "card images" that could be used to produce the figure at the start of this section. The reader is not expected to understand all of the FORMAT techniques used, although most of them will be familiar.

```
CAPITALIZE AUTOMATICALLY  
WIDTH OF COLUMN IS 52 PRINT POSITIONS  
LINES PER PAGE ARE 32  
TEXT STARTS ON LINE 5, IN PRINT POSITION 1  
COLUMNS PER PAGE = 1  
LEFT TOP POSITION FOR PAGE NUMBER  
PAGE NUMBER STARTING AT 12  
TITLE STARTS ON LINE 1, IN PRINT POSITION 24  
)F TITLE )FE  
FOOTER STARTS ON LINE 32, IN PRINT POSITION 23  
)F FOOTER )FE  
GO
```



```

)P THIS IS          THE BEGINNING OF A PARAGRAPH; THE
                     SIZE OF THE INDENT AT THE START    OF THE
PARAGRAPH MAY BE SPECIFIED ON A CONTROL CARD. )L NOW,    )L
THIS
MATERIAL BEGINS A NEW COLUMN-LINE: THAT IS, IT          STARTS
A NEW LINE WITHIN THE CURRENT COLUMN OF TEXT MATERIAL. )LLV
INDENTATION OF THE COLUMN IS (0,10), (7,8) POSITIONS
GO
)I #THIS TEXT MATERIAL ILLUSTRATES
THE USE OF AN INDENT: THE RIGHT MARGIN HAS BEEN INDENTED AN
ADDITIONAL 10 SPACES.    )ILLH2 THIS TEXT MATERIAL
ILLUSTRATES THE USE OF A HANGING, OR DELAYED, INDENT: THE
TEXT IS NOT INDENTED UNTIL THE LINE FOLLOWING THE FIRST
LINE OF TEXT. )H2LLL
)M THIS TEXT )L MATERIAL )L IS )L    CENTERED! )ME

```

This example shows the three levels of control provided by FORMAT. Control cards provide global controls; Command Words provide controls at the word level; and Special Operands provide controls at the character level.

FORMAT provides a number of other powerful capabilities such as the DICTIONARY, \$LOCATE, and EDITOR facilities. The beginner should experiment with simple text input until some familiarity with FORMAT has been attained. As a start, study the input which produced the examples above. Then punch the above FORMAT job on cards, add the necessary Job Control cards (see Section X; an experienced programmer can help prepare them), and run the job on the computer. Then generate some simple input text, and experiment with other FORMAT features.

A suggested sequence for reading this manual is to skim Sections III through V, and the hints and suggestions given in Section XI. Then, after studying the above input and running a few simple problems, go back and study those sections more carefully. As more experience is gained, the other parts of the manual may be consulted as needed.

III. Control Cards

Every FORMAT job must begin with a control card group, which is defined as a group of control cards ending with the "GO" control card (all other control cards are optional). The position of a control card within a control card group is not significant, unless specified in its description. A control card group may appear at any point in the input text stream (see the "V" Command Operand in Section IV). Default values for each option are assumed if no control card pertaining to that particular option has been supplied in any control card group in the job. The default values are summarized in Section VIII. The values of most of the options can be varied as needed during the reading of the input and the formatting of the document. However, some of the options cannot be reset once they have been set, as noted in their descriptions.

The format of each control card is entirely free-form, as long as the first 3 non-blank characters of each control card are as specified by the suggested control card name, and the control card is on one card image. (FORMAT scans control cards by saving the first three non-blank characters, and then searching for the numbers that give the values of the parameters.) At the conclusion of each job, the control cards used for that job are listed by group on the System Output dataset.

We will now give the specifications for each of the control cards in turn. In some of the descriptions, it is stated that some option may or may not be used in a title; in all such cases, the statement applies to footers also. Examples of correct and faulty control cards will be given at the end of this section. In some of the control cards, numeric operands may be required. These are represented by lower-case letters such as x, y, z, or nn; an operand such as x is not limited to a single digit.

BACKSPACE CHARACTER IS SPECIAL CHARACTER NUMBER nn

To simulate the action of the backspace key on a typewriter, one of the special characters (described in Section V) may be designated as the "backspace" character, except for special characters numbered 43 (§) and 51 (!). The action of the backspace character is as follows: the character to be printed

over and the overprint character are separated by the backspace character, with a few minor exceptions. If the backspace character is followed by a blank, then it is assumed that no overprint was desired, and the backspace character will print normally. Multiple backspaces are ignored, and have no more effect than a single one; they all cause only a single backspace, and the only character which will overprint the character preceding the first backspace will be the character following the last backspace.

The number nn given on the control card must lie between 10 and 50; if it does not, backspacing will be turned off and no character will be recognized as a backspace. Note that the backspace character, when used in the input text, may be in its actual (character) form or in its special (!nn) form. The default action is that no backspaces are recognized.

To give some examples: suppose the backspace character is number 50, the question mark (?). Then the input characters 0?- would produce 0, /?o would produce ø, and lett?_er would produce letter. Note that special characters may be used for overprinting, so that =!50!33 would produce ‡. The figure below makes use of backspacing to print the dividers at the inside edges of the boxes: the characters -?, produce †, and the characters -? produce ‡.

base	displacement
------	--------------

At most 99 backspaces are allowed on a single page. Any backspaces following the 99th will be ignored, and the backspace character will print normally. An error message will flag the location of the 100th backspace on the page.

Backspaces will not work correctly inside a "keep" (a region of text delimited by)K's; see Section IV for a description of a "keep"). The backspace character itself may not be used for overprinting. Backspacing does not apply in titles and footers.

BETWEEN COLUMNS LEAVE x BLANKS

The number of print positions separating text columns is x. The default number is 2.

CAPITALIZE AUTOMATICALLY

When this control card is in effect, FORMAT will automatically capitalize the first word of the document, the first word following Command Operands "P" and "S", and each letter which follows .b !b ?b ."b !"b ?"b .)b !b and ?)b (where b = one or more blanks) in text and titles. The default action is that this option is not used. (See the "NO CAPITALIZATION AUTOMATICALLY" control card.)

CARD FIELD IS x THRU y
or
CARD FIELD EXTENDS THRU y

This control card (in either form) specifies the columns of the input data cards to be used for reading normal text (in text mode), and text for titles and footers (which is read in control card mode). The first column of the card field is x, and the last column of the card field is y. If the second form of the control card is used, the card field extends from column 1 through column y. This control card does not affect control cards (which may be limited by the "CONTROL CARDS END IN" control card; see below), but all other card input to FORMAT, including titles, is read from the field specified. The field must be at least 3 columns wide, and at most 80 columns wide. The default card field is columns 1 through 80.

CENTER TEXT ON LINE x

The first line of the text is printer line x, and the document is centered, if possible, within the print line of 132 characters. The default is line 5 and centering of the document on the printer page. (See the "LINES PER PAGE" and "TEXT STARTS ON" control cards also.)

COLUMNS PER PAGE = x

The number of text columns per document page is x. The maximum allowable number of text columns per page is eight. The default number is 1.

CONTROL CARDS END IN COLUMN x

This control card allows the user to control the position of the right-hand margin of a control card in the same way as

can be done for text input with the "CARD FIELD" control card. If the value of x is less than 7 or greater than 80, it will be set to 80. This control card takes effect starting with the following control card. The default value of x is 80. Note that even though the text for titles and footers is part of a control card group, the card field from which it is taken is set by the "CARD FIELD" control card.

COPIES = x

X specifies the number of copies of the document which are to be produced during the run. The default value is 1. If x is 2 or more, the output dataset from the program is written onto dataset reference number 8 (see Section IX). At the conclusion of the last FORMAT job, dataset reference number 8 is copied onto the System Output dataset x times, where x is the operand field from the last "COPIES = x" control card read. If x is zero it is treated as one, unless the "OUTPUT IS TAPE" control card is specified.

CREATE A TAPE FROM CARD INPUT

If the input dataset is currently the System Input dataset (which is the normal situation), then the entire input dataset following this control card is copied and condensed onto dataset reference number 2. Dataset reference number 2 is then rewound and becomes the input dataset. Printed in the upper far right corner of each document page produced are the first and last card image numbers (from the condensed deck) that were used in producing that page.

FORMAT's condensing function squeezes out unneeded blanks, and responds to but prevents the following three control cards from being copied into the condensed dataset: "029 KEYPUNCH", "026 KEYPUNCH", and "CARD FIELD IS... ". The result is a compact card image dataset (80 characters per record, all of which are used) on which all right parentheses (except those within "as is" regions) are in the 029 (EBCDIC) mode, regardless of their mode in the original card input dataset.

At the conclusion of the run the input dataset on dataset reference number 2 (the condensed input) is listed, with card image numbers and numbered text and title words, onto the System Output dataset. If the listing is printed in upper case only (due either to errors or to the presence of the "SPECIAL PRINT TRAIN" control card), then an asterisk will replace each character for which no graphic is likely to be associated. The

Command Operands contained in each card image are listed again alongside each card image.

The primary use of this control card is to produce a card image input dataset that can be saved for later editing; see Section VI for a description of the Editor facility.

CYCLE THE PAGE NUMBER

If page numbering has been requested (by the LEFT TOP POSITION or RIGHT TOP POSITION control card), then the page number will be alternated between the left and right top corners on successive document pages. The page number appears on line 1 aligned with the appropriate border of the text. The default action is that the first page number is aligned with the right text border. (See the "LEFT TOP POSITION", "RIGHT TOP POSITION", and "PAGE NUMBER" control cards.) Once cycling of the page number has been requested, it stays in effect for the remainder of that job.

DARK PRINT EACH PAGE x TIMES

Normally, each line on the output page will be printed once. If x has a value of 2 or 3, each line will be printed successively on top of itself until it has been printed a total of x times. This allows darker printing of the page, and if the printer is well-adjusted and the printer ribbon is neither too new nor too old, the text is printed without the normal blur and grain from the ribbon. If x is 0, it is set to 1, and if it is greater than 3, it is set to 3. The number of times each line is printed is determined by the value of x in effect at the time the entire page is printed, so it is not possible to print portions of a page in "boldface". The default is single printing.

DICTIONARY OF WORDS USED

An alphabetized list, 6 columns per page, of all significant words in the input stream, with a count of the occurrences of each, is written onto the System Output dataset at the conclusion of the last FORMAT job. This dictionary, in upper case, is formed according to the following rules:

- no word of fewer than 3 letters is listed
- all non-letters are treated as word delimiters, except for "z" which is ignored

- Command Words are ignored
- case of the printed text is ignored, but the input must be in upper case (see the "SPECIAL KEYPUNCH" control card)
- text, titles, and control cards alike are scanned
- words longer than 40 letters are broken up into 40 letter segments
- 94 common words (such as "though", "also", and "where") are suppressed

The dictionary is useful for determining a rough list of candidates for an index, and for a spelling check. The "\$LOCATE" Editor control card can be used (in the edit phase) to find the location of "index candidates" in context.

The DICTIONARY facility uses dataset reference numbers 2 and 3 (see Section IX for details).

DROP CHARACTER FOR 'D' COMMAND IS x

When a tab command is used to skip over blank positions in a column line, the spaces can optionally be filled with a character such as a dot. This character is called the "drop" character, since it may be thought of as being "dropped behind" as the line position moves to the right. Normally, the character dropped by the "D" Command Operand (see Section IV) will be a period. This control card may be used to change that character, as follows: if x is a number between 10 and 51, then the drop character will be the corresponding special character; if x lies between 64 and 255, the drop character will be the EBCDIC character whose representation has that value; if it is zero or omitted, then the drop character will be reset to a period. The default character is a period. As an example, the control card "DROP 30" would drop "bullets" (•) when the 'D' command operand is used.

EDITOR

This control card invokes the FORMAT Editor, which is described in Section VI. If used, this control card must be the first of the job and must be part of the System Input dataset.

FOOTER ON LINE x PRINT POSITION y PRECEDED BY z BLANK LINES

The footer is placed into the print page beginning on line x at print position y, and is separated from the last line of text by at least z blank lines. This control card, if used, must

be followed immediately by the footer text. The text of the footer must be ended by the "E" Command Operand. After the footer text, the only allowable control card is the "TITLE" or the "GO" control card. The footer appears on every document page until it is replaced (through the use of another "FOOTER" control card.) The default value for x is the last line of the document page, the default value for y is the print position of the left text border of the document, and the default value for z is 2.

GO

This is the only control card required by FORMAT. GO signals the end of a control card group, and initiates processing in normal text mode.

INDENTATION OF THE COLUMN IS (x1,y1).....(x7,y7) POSITIONS

This control card, when used with the "H" and "I" Command Operands, enables the user to reduce the width of text columns by x positions on the left and y positions on the right. Seven pairs of column indentations may be specified. The default action is that all x's and y's are zero.

JUSTIFICATION

Text in the document body is right-justified within column-lines, except when a column-line is terminated by a Command Word, or when the line contains tabs. After reading the input and eliminating all extra blanks, FORMAT then performs right-justification by introducing the necessary number of extra blanks, one to each word delimiter, working alternately from the right end of the line leftward and the left end of the line rightward on successive lines. The number of blanks between input text words is ignored. FORMAT does no hyphenation, which means that column-lines containing long words may have large gaps between words. The default action is right-justification. (See the "NO JUSTIFICATION" control card.)

LEFT TOP POSITION FOR PAGE NUMBER

The page number (if any) is placed on line 1 aligned with the left text border. The default action is that it is aligned with the right text border. (See the "CYCLE PAGE NUMBER", "PAGE NUMBER", and "RIGHT TOP POSITION" control cards.)

LINES PER PAGE ARE x

The number of lines of all kinds (including text lines, paragraph separation lines, title lines, and blank lines) which are allowed on a document page is x. The operand may be any number in the range 5 through 1000. The default number is 59.

LIST THE INPUT DATASET

This control card is identical in effect to the "CREATE A TAPE" control card.

NO CAPITALIZATION AUTOMATICALLY

No capitals are automatically produced. This is also the default. (See the "CAPITALIZE AUTOMATICALLY" control card.)

NO JUSTIFICATION

The text is not right-justified (as illustrated in this paragraph, which will have an uneven right margin.) The number of blanks between input text words is ignored. The default action is right-justification of text.

NONTRIVIAL BLANK IS REPRESENTED BY SPECIAL CHARACTER nn

To facilitate the use of the non-trivial (or non-eliminatable) blank from devices (such as IBM 2741 terminals) which do not allow it to be entered in the source stream, the user may make the appearance of one of the special characters be equivalent to the presence of a non-trivial blank. The number nn must be between 10 and 51; otherwise no character will be replaced by the non-trivial blank when it is encountered. Note that the actual special character must be present to be replaced, and not the "special character representation" !nn, which will be treated normally. For example, if the nontrivial blank is represented by special character number 46 (@), then the text "here@@@@there" would be printed as "here there", and the non-trivial blanks are not eliminated as ordinary blanks would be. This equivalence also takes effect in titles and footers. The default is that no such equivalence is made.

NULL CHARACTER SWITCH SET TO x

Non-trivial blanks (or null characters) are normally ignored for centering or underlining purposes when they are at the end of a word. If x has the value 2, they will not be ignored when centering and underlining (under control of the "M" and "U" Command Operands, respectively). If x has any other value, it will be set to 1, which implies that null characters will be treated normally. The default setting is 1. The effect of this card does not apply in Titles or Footers.

OUTPUT MEDIUM IS TAPE

The output from FORMAT is written onto dataset reference number 8 from the point at which this control card is read. At the conclusion of the job(s) the tape is copied onto the System Output dataset the number of times specified on the last read "COPIES = x" control card; or once, if multiple copies are not specified. The tape can then be listed at some other time, using the "PRINT OUTPUT TAPE" control card.

PAGE NUMBER STARTING AT x

The page number starts at x (if non-blank and non-zero) and is placed on line 1 of each document page. If x is zero or blank, page numbering is suppressed. The default page number is 1. (See the "CYCLE PAGE NUMBER", "LEFT TOP POSITION", and "RIGHT TOP POSITION" control cards.) If page numbering is requested (by the "RIGHT TOP POSITION" or "LEFT TOP POSITION" control cards), then enough character positions must be reserved at both the top left and top right corners of the page for the digits of the page number, whether or not the number will actually appear in both positions.

PARAGRAPH INDENT IS x

The number of print positions skipped at the start of a paragraph is x. The default indentation is 5 print positions.

PRINT OUTPUT TAPE

The presence of this control card means that the user has placed a FORMAT-generated output dataset (usually a tape) onto dataset reference number 8, and that he wishes to list it onto the System Output dataset the number of times specified on the

most recent "COPIES = x" control card; or, if none, once. This action is immediate, no document is formed from an input dataset, and no control cards or error diagnostics relating to the current input are written. It is suggested that the tape be file-protected.

PUNCH THE INPUT DATASET

The effect of this control card is identical to that of the "CREATE A TAPE" control card, and in addition, the newly created condensed input deck is punched (i.e., written onto the System Punch dataset) as well as listed at the conclusion of the run.

REPEAT TITLE ON EVERY PAGE

The title (if any) is printed on every page of the document. The default action limits the appearance of the title to the next page produced. (See the "STOP PRINTING TITLE" control card.) Note that the title and footer printed for a given page of text are those in effect when the end of the current page is reached. This means that changing the title or footer when text is being accumulated in mid-page will place the new title or footer on the current page, replacing the old one (possibly before it was expected to).

RIGHT TOP POSITION FOR PAGE NUMBER

The page number (if any) is placed on line 1 aligned with the right text border. The default action is the same as the action of this control card. (See the "CYCLE PAGE NUMBER", "LEFT TOP POSITION", and "PAGE NUMBER" control cards.)

SENTENCES SEPARATED BY AT LEAST x SPACES

Text sentences are separated on the same column-line (when not in an "as is" region; see Section IV) by x blanks with "NO JUSTIFICATION" in effect, and by a minimum of x blanks with "JUSTIFICATION" in effect. The value of x may be 1 or 2; if it is not 2, it will be set to 1. The default value for x is 1. Note that FORMAT will insert a minimum of x blanks, so that extra blanks might appear. If an exact number of spaces is needed, use the non-trivial blank.

SEPARATION LINES BETWEEN PARAGRAPHS ARE x

The number of blank printer lines between paragraphs is x. The default number is 1.

SIDE BY SIDE COPIES

Two copies of the document are produced simultaneously, side by side. The default action is not to print side by side copies. There must be enough space to fit two copies of the printed text, and at least one separating space, into a 132-character print line.

SPACING OF TEXT LINES IS x

The spacing for the document is x (e.g., x = 1 means single spacing, x = 2 means double spacing, etc.). The default assumption is single spacing.

SPECIAL KEYPUNCH

SPECIAL KEYPUNCH IS A 2741

This control card specifies that the text input originated on an upper and lower case keypunch, or on any device producing the specific EBCDIC code for each character desired. Case is not altered by the program, and (for the first form of this control card) the Special Operand "Z" does not produce superscripts. The default action is the usual case and superscript conversion. The "SPECIAL KEYPUNCH" control card allows the user to prepare upper and lower case input from a terminal, but still be able to obtain superscripts through the use of the "Z" Special Operand technique. If there is any other numeric quantity on this control card, the normal "SPECIAL KEYPUNCH" will be assumed. Note that the effect of this card can be changed from 2741 mode to normal SPECIAL KEYPUNCH mode and back, but there is no way to return from either to the normal mode, where upper-case-only input is assumed.

SPECIAL PRINTER TRAIN

This control card implies that the ultimate printer of the document cannot print lower case or superscript characters. Therefore, no translation to lower case or superscripts is made. The default action is that the translation is made.

STOP PRINTING TITLE ON EVERY PAGE

The title (if any) is only printed on the next page produced. The default action is the same as the action of this control card. (See the "REPEAT TITLE" control card.)

TABS ARE SET AT x1.....x14

The operand fields of this control card specify the positions of up to 14 tab stops. These are given as relative character positions within the text column-line (e.g., a tab set at 10 means that the tab field begins in character position 10 in each column-line; the first word following a tab from any position in the column-line before position 10 will be placed in the line starting in character position 10). Tabs must be set in ascending order, and no tab may be set at a position greater than the column width. The action of this control card is analogous to the action of the "tab set" key on a typewriter. The default action is that no tabs are set.

TAPE INPUT DATASET

The presence of this control card means that the user has placed a tape input dataset onto dataset reference number 2. It is suggested that the tape be file-protected. If this control card is used it must be the first control card of the job and must be part of the System Input dataset. This control card calls the FORMAT Editor; therefore, Editor control cards may follow it (see Section VI). It is identical in effect to the "EDITOR" control card, except that no listing of the edited tape is requested.

TEXT STARTS ON LINE x IN PRINT POSITION y
OR
START TEXT ON LINE x IN PRINT POSITION y

The first line of the text is printer line x, and the first print position is y. The default is line 5 and the document is centered on the printer. These control cards are entirely equivalent, and the two forms are provided as a convenience.

TITLE STARTS ON LINE x IN PRINT POSITION y

The first printer line of the title is x, and the first print position of the title is y. This control card, if used,

must be followed immediately by the card images containing the title and the "GO" or "FOOTER" control card. (Remember that the title text must end with the "E" Command Operand.) The title must be positioned above the body of the document. Conflicts of the title with the page number are resolved in favor of the page number, at both of the top corners of the page. The default title line is printer line 2, and the default print position is that of the left text border.

UNDERLINE SWITCH SET TO x

If x is not zero, the underlining algorithm of FORMAT is modified so that the leading and trailing characters of an underlined string will not be underlined if they are any of the following ten punctuation or special characters: period, comma, colon, semicolon, question mark, exclamation point, quotation mark, apostrophe, and left or right parenthesis. If x is zero or blank, all characters in the string are underlined. The default value of x is zero. To give an example, suppose the input text requires that ((X)) be underlined. Then

((X)) and ((X))

would be produced by setting the underline switch to zero or nonzero respectively.

WIDTH OF COLUMNS IS x PRINT POSITIONS

The width in print positions of each text column is x. The default width is 64 print positions. If a single column per page is specified, and the width is chosen to be 132 characters (the maximum), then the maximum number of lines is 59. FORMAT allots 59*132, or 7788, characters for each page of the text. The maximum allowable column width, w, is computed as follows:

$$w = \frac{[\min(7788/L, 132)]/c - s + 1 - b * (n - 1)}{n}$$

where:

- L = lines per page (from "LINES PER PAGE" control card)
- c = 2 if "SIDE BY SIDE" control card is in effect;
= 1 otherwise
- s = starting print position (from "TEXT STARTS ON" control card)
- b = spaces between columns (from "BETWEEN COLUMNS" control card)
- n = number of text columns (from "COLUMNS/PAGE" control card)

026 KEYPUNCH

This control card specifies that the Command Words used in the input cards have been punched on an IBM 026 Keypunch, or any keypunching device which punches 12-8-4 for the character used to begin Command Words [nominally ")"]. The default action assumes the IBM 029 configuration for the ")", which is 11-8-5.

029 KEYPUNCH

This control card specifies that the Command Words used in the input cards have been punched on an IBM 029 Keypunch, or any keypunching device which punches 11-8-5 for the character used to begin Command Words [nominally ")"]. The default action is the same as the action produced by this control card.

Examples of Correct Control Cards

[1]Card Columns.[80]

```
TAB SET      5    10    15    20
  TABS ARE SET AT PRINT POSITIONS 5, 10, 15 AND 20
    TABULATE TO 5 10 15 20
  TAB5 10*                                15                                20

START THE TEXT ON LINE 10, PRINT POSITION 20
START THE TEXT ON LINE 10 PRINT POSITION 20
  START DOCUMENT:  LINE = 10, P. POSITION = 20
STA      10      20
  TEXT STARTS ON LINE 10, PRINT POSITION 20

TEXT STARTS IN DEFAULT POSITION
  S T A R T   T E X T   0   ,   10
    START TEXT   0       10
TEXT STARTS ON LINE 5 IN PRINT POSITION 10

TITLE
TITLE STARTS ON LINE 5 ABOVE LEFT TEXT BORDER

BACKSPACE CHARACTER IS NUMBER 50 (THE QUESTION MARK)
BACK 50 (?)

DAR 2
NO J                MEANS DON'T JUSTIFY
```

Examples of Faulty Control Cards

```
GO NOW
BEGIN TEXT
  (An unrecognizable control card is treated as a "GO" card)
START TEXT:  ,7
  (Text starts on line 7 in the default print position)
TABS = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  (The 15th tab stop is ignored)
LINES/PAGE = 5 7
  (The number of lines per page will be 5, not 57)
COLUMNS/PAGE = TWO
  (The number of columns/page will be the default value)
BACKSPACE CHARACTER IS NUMBER 43 (¢)
  (¢ cannot be used as the backspace character)
```

IV. Command Operands and Command Words

A Command Operand is an imperative order to perform an immediate text-control function. Command Operands can be interspersed as desired throughout the text input in the form of Command Words. A Command Word is a string of one or more Command Operands in the order of desired execution, prefixed by the character ")", and followed by a blank. (See the "026 KEYPUNCH" and "029 KEYPUNCH" control cards in Section III).

Some Command Operands may be used in titles (including footers), as noted in the following paragraphs. In addition, some of the Command Operands ("D", "H", "I", "T", and "W") may be followed by a numeric quantity, as in ")T4".

The functions of the Command Operands are described below. Wherever a single letter appears in quotation marks (like "L"), its use as a Command Operand (like ")L") is implied; otherwise, that letter may appear freely in the text material. It is important to remember that all Command Operands are recognized by FORMAT only if they are in upper case.

A -- Enter the "as is" text mode. In this mode, each card image is an integral unit and is printed on a separate column-line without change to the spacing of text, except that the Special Operands, and certain of the Command Operands, take no print positions. Both of the Special Operands, and the Command Operands "z" and "F", are effective in this mode. The content of a card image beyond the effective column width is not printed. The "as is" text mode is ended when ")b" (where b = blank) occurs in the first two positions of the card image; FORMAT then resumes reading in normal text mode. (The Command Operand "A" is not recognized in "as-is" text mode, so ")A" cannot be used to terminate "as-is" mode.) The Command Operand "A" is not valid in titles.

C -- Begin the next text column. The Command Operand "C" is not valid in titles or footers.

Dn -- Tab, like "T", leaving a string of dots (periods) instead of blanks. The string of dots is prefaced by one blank. The character to be "dropped" may be changed from a dot to any other

character with the "DROP CHARACTER" control card. For a detailed description of the "D" Command Operand, see the description of the "T" Command Operand below. The Command Operand "D" is not valid in titles or footers.

E -- End the job, or end the footer or title; the next card will be read in control card mode. As many FORMAT jobs as desired may be stacked one behind the other. Jobs on cards must precede jobs which are tape resident.

F -- Begin capitalizing each word, continuing until another "F", "p", "S", or "V" Command Operand occurs. The Command Operand "F" is valid in titles and footers, and in "as-is" text mode.

Hn -- Reduce the column width starting with the next column-line. If no "hanging" (delayed) indent is currently in use, then n refers to the nth pair of arguments on the "INDENT COLUMN" control card; if n is blank, zero, or one, the first pair is referenced. If a hanging indent is currently in use, the nth pair of column indents replaces the pair in use, unless the nth pair is the one being used, in which case the hanging indent is turned off. If any hanging indent is in effect, a blank or zero value for n turns it off, as does the Command Operand "S". The Command Operands "H" and "I" may be used independently of each other. The Command Operand "H" is not valid in titles or footers.

In -- Reduce the column width immediately, and terminate the current column-line. If no immediate indent is currently in use, then n refers to the nth pair of arguments on the "INDENT COLUMN" control card; if n is blank, zero, or one, the first pair is referenced. If an immediate indent is currently in use, the nth pair of column indents replaces the pair in use, unless the nth pair is the one being used, in which case the immediate indent is turned off. If any immediate indent is in effect, a blank or zero value for n turns it off, as does the Command Operand "S". The Command Operands "I" and "H" may be used independently of each other. The Command Operand "I" is not valid in titles or footers.

J -- Start the next column-line. The action of this Command Operand is identical to the action of "L", except that it is not valid in titles and is effective at any line in the text. Unlike "L", it is not ignored at the top of a column.

K -- Keep the following text, until the next occurrence of "K", in the current text column, if possible. (A segment of text delimited by "K" Command Operands is called "kept text" or a "keep".) Otherwise, start this block of text in the next text column. The Command Operand "K" is not valid in titles and footers, and terminates the column-line on which it occurs. If this Command Operand is used, the program requires dataset reference number 2. Backspaces do not work properly inside "kept" text. See the description of the "W" Command Operand also.

L -- Start the next column-line, if not at the first line of a text column. The action of this Command Operand is similar to the action produced by striking the "return" button on an electric typewriter. "L" is valid in titles. If "L" is used in a title, the next printer line is begun; if "L" is used in the body of the document the next column-line is begun, leaving ("SPACING" - 1) blank lines between. If "L" is used at the top of a text column, it is ignored.

M -- Begin centering text within the column-line, and continue doing so for successive lines until another "M", "P", or "S" occurs. The Command Operand "M" is not valid in titles or footers; if centering is required in a title or footer, the appropriate number of non-trivial blanks may be used (see Section V).

P -- Begin a new paragraph, leaving the number of printer lines specified by the "SEPARATION LINES" control card (or its default) between paragraphs and indenting the number of print positions specified by the "PARAGRAPH INDENT" control card (or its default). A new column is begun if at least two column-lines of the present column are not available for the new paragraph. "P" stops the action initiated by Command Operands "F", "M", "U", and "Z". If "CAPITALIZE AUTOMATICALLY" is in effect, the next text word is capitalized. "P" is not valid in titles.

S -- Begin a new page. "S" stops the action initiated by Command Operands "F", "H", "I", "M", "U", and "Z". If "CAPITALIZE AUTOMATICALLY" is in effect, the next text word is capitalized. "S" is not valid in titles.

Tn -- If n is blank or zero, tab to the next set tab position beyond the present position in the column-line. The action of

"T" corresponds to the action produced by striking the tabulate key on a typewriter. Right-justification, if in effect when "T" is used, will not be performed for the column-line on which the tab occurs. The "D" and "T" Command Operands may be followed by a number which specifies the tab stop to be used. That is, "T4" will cause a tabulation to the fourth tab position on the current column-line. If the command operand is used incorrectly, it will be ignored, and a diagnostic message will be printed. "T" is not valid in titles.

U -- Begin underlining, continuing until another "U", "P", or "S" occurs. At most 99 column-lines, or portions, may be underlined per page. Underlines neither begin nor end under the spaces skipped over by tabbing (produced by "T" and "D"). Underlines may or may not begin and end with non-trivial blanks, depending on the "NULL CHARACTER SWITCH" setting in effect (see the "NULL CHARACTER SWITCH" control card description in Section III). Underlines may or may not begin under punctuation characters, depending on the "UNDERLINE SWITCH" setting in effect (see the description of the "UNDERLINE SWITCH" control card in Section III). Individual characters within a word cannot be underlined except by backspacing (see the "BACKSPACE" control card description in Section III). The Command Operand "U" is not valid in titles or footers.

V -- Leave normal text mode, and begin to read in the next group of control cards. A control card group must immediately follow the card image containing the "V". All characters following "V" on the same card image are ignored. "V" stops the action initiated by the Command Operands "F" and "Z". If "CAPITALIZE AUTOMATICALLY" is in effect, the next text word is capitalized. The Command Operand "V" is not valid in titles or footers.

Wn -- Keep the next n column-lines in the same text column. If n column-lines do not remain in the current text column, start the next text column. "W" terminates the column-line on which it occurs. It is not valid in titles. Note that the "W" Command Operand is similar in effect to "K", but does not require the use of an additional dataset. "W" can be used to prevent "widows", which are small segments of text left alone at the bottom of a column.

Z -- Begin printing all letters in upper case, continuing until another "Z", "P", "S", or "V" occurs. The action of "Z" is equivalent to locking a typewriter keyboard in upper case, and

then unlocking it. "ø" is valid in titles. "ø" does not cause numbers to be printed as superscripts, it is not affected by punctuation characters, nor does it cause letters punched in lower case (as indicated by a "SPECIAL KEYPUNCH" control card) to be printed as capitals. (See the description of the "ø" Special Operand in Section V also.)

Summary of Command Operands

1. Command Operands allowed in titles and footers are: "E", "F", "L", and "ø".
2. Command Operands whose effect is ended by "P" are: "F", "H", "U", and "ø".
3. Command Operands whose effect is ended by "S" are: "F", "H", "I", "M", "U", and "ø".
4. Command Operands which terminate the column-line in which they appear are: "A", "C", "E", "I", "J", "K", "L", "M", "P", "S", "V", and "W".
5. Command Operands which cause capitalization of the next text letter (if "CAPITALIZE AUTOMATICALLY" is in effect) are: "P", "S", and "V".
6. Command Operands valid in "as-is" regions are: "F" and "ø".
7. Command Operands which terminate the effect of "U" are: "P", "S", and "U".
8. Command Operands which terminate the effect of "ø" are: "P", "S", "V", and "ø".
9. Command Operands which terminate centering (initiated by "M") are: "M", "P", and "S".
10. Command Operands which terminate the effect of "F" are: "F", "P", "S", and "V".
11. Command Operands which may be followed by a numeric quantity are: "Dn", "Hn", "In", "Tn", and "Wn".
12. Command Operands whose effect is turned on or off by alternate occurrences of the Command Operand are: "F", "H", "I", "K", "M", "U", and "ø".

Examples of Command Words

1. MEN ARE SLOW)L TO GRASP NEW IDEAS;

"to grasp" begins a new column-line.

2.)LTTTU~~U~~ NOW)U~~U~~ IS THE TIME.

A new line is begun, and beginning at the third tab position is printed: NOW is the time.

3. Note that ")CP" (meaning "begin next text column" followed by "start a new paragraph") does not produce the same effect as ")PC", which is effectively the same as ")C". This is because the "P" begins a paragraph, but the following "C" immediately starts a new column. Because "C" ends the column-line on which it occurs, the indent (if any) at the start of the paragraph was lost.

4.)M *)L ***)L *****)L *****)L *****)L ***)L *)M

This produces:

```

      *
      ***
      *****
      *****
      *****
      *****
      ***
      *
    
```

5. The text "blank) blank" is valid, and is not interpreted as the beginning of a command word. Thus, "(A+B) * C" produces: "(a+b) * c", but "(A+B) * C" produces "(a+b) * c" and an error diagnostic (code 700) for an undefined Command Operand [the "C"]. (The diagnostic listing at the end of this manual shows how the above error was diagnosed.)

6. The text ")~~U~~ FORMAT,)~~U~~" produces: "FORMAT," but the text ")~~U~~ FORMAT)~~U~~," produces: "FORMAT," and a diagnostic message for an undefined Command Operand, the comma. Thus, punctuation characters such as commas and periods should be placed immediately after the text word they would normally

follow, and any Command Words should then follow the blank after the punctuation character.

7.)H2 TEXT ... TEXT)H

This text material illustrates delayed column indentation, produced by Command Operand "H". Note that either one of the margins, or both margins (as in this example), can be drawn in, as the user desires. In this example, the second pair of column indentations was "(5,5)", so that both margins were indented 5 spaces. Note also that the final "H" could have been "H2".

8.)I3 TEXT ... TEXT)I3

This text material illustrates immediate column indentation, produced by Command Operand "I". In this example, the third pair of column indentations was "(5,0)", so that only the left margin was indented 5 spaces. The extra indent of the first line occurred because a paragraph was started by a "P" Command Operand.

9.)W10JJJJJJJJJJ

A block of 10 blank column-lines is left in the same text column (assuming of course that the "SPACING OF TEXT LINES" is 1). This is useful for the later insertion of a photograph, for example.

V. Special Operands for Capitalization and Special Characters, and the Non-Trivial Blank

FORMAT can produce upper and lower case and special characters in two ways. If the text input is punched with the Hollerith codes representing the characters desired (such as are produced by a terminal or by an upper and lower case keypunch, for example), the proper character representations on output are supplied directly by the hardware of the computer system. If, however, an upper and lower case keypunch or terminal is not used (or approximated by multi-punching on a standard keypunch), then upper and lower case and special characters can be produced using the Special Operands.

There are two Special Operands for use with standard IBM 029 and 026 type keypunches. "¢" is used for capitalization and numeric superscripts, and "!" is used to produce special characters. Both Special Operands are valid in "as-is" text mode. It is important to remember that neither of the Special Operands needs to be preceded by the ")" escape character.

The "¢" Special Operand:

A letter preceded immediately by "¢" is printed in upper case, a number so preceded is printed in superscript form, and any other symbol so preceded is printed preceded by the graphic "¢". If one of the "SPECIAL" control cards is in effect the translation of a number to a superscript is not made, unless only the "SPECIAL KEYPUNCH IS A 2741" has appeared. The "¢" Special Operand may be preceded by any character. The "¢" character must be multi-punched on an IBM 026 type keypunch. (Note that the "¢" Command Operand causes all following letters to be capitalized, whereas the "¢" Special Operand causes only the single, immediately following, letter to be capitalized. Refer back to the sample input in Section II for an example.)

The "!" Special Operand

A special character is defined as one which is neither a letter nor a number (normal or superscript) nor one of the following: *\$.-,/. A special character is produced whenever the string of characters "!"nn" is used, where nn is any number from 10 to 51: for example, !28 produces "n". If nn is not in the range from 10 to 51, then "!"nn" is printed.

The correspondence between the values for nn, the TN Print Train graphics, the EBCDIC hexadecimal character codes, and the punched card codes is shown below.

Nn	TN	hex	Card Code
10	(	8D	12-0-8-5
11	)	9D	12-11-8-5
12	+	8E	12-0-8-6
13	-	A0	11-0-8-1
14	{	8B	12-0-8-3
15	}	9B	12-11-8-3
16	[	AD	11-0-8-5
17	]	BD	12-11-0-8-5
18	≤	8C	12-0-8-4
19	≥	AE	11-0-8-6
20	±	9E	12-11-8-6
21	≠	BE	12-11-0-8-6
22	⊥	AB	11-0-8-3
23	⌋	BB	12-11-0-8-3
24	⌈	AC	11-0-8-4
25	⌌	BC	12-11-0-8-4
26	⊕	8F	12-0-8-7
27	⊖	BF	12-11-0-8-7
28	⊞	9C	12-11-8-4
29	⊟	9F	12-11-8-7
30	⊠	AF	11-0-8-7
31	∘	A1	11-0-1
32	ε	50	12
33	⌑	4F	12-8-7
34	⌒	5F	11-8-7
35	<	4C	12-8-4
36	=	7E	8-6
37	>	6E	0-8-6
38	+	4E	12-8-6
39	(	4D	12-8-5
40	)	5D	11-8-5
41	"	7F	8-7
42	'	7D	8-5
43	∕	4A	12-8-2
44	#	7B	8-3
45	%	6C	0-8-4
46	@	7C	8-4
47	-	6D	0-8-5
48	;	5E	11-8-6
49	:	7A	8-2
50	?	6F	0-8-7
51	!	5A	11-8-2

It should be noted that the special characters from nn = 32 through nn = 51 can be punched directly on the standard IBM 029 Keypunch, that the "!" character itself must be multi-punched on an IBM 026 type keypunch, and that !43 is not equivalent to either the "Z" Special Operand or the "Z" Command Operand.

The Non-Trivial Blank

The character produced by punches in the 0, 8, and 2 rows of a single card column (which has EBCDIC representation E0) is replaced by a "non-trivial" blank; i.e., one which is never eliminated by the program. The IBM 029 Keypunch has a key which provides this configuration of punches directly. The non-trivial blank is treated in all respects as if it were a non-blank character except that it may or may not be the first or last character underlined, and it may or may not be considered for centering purposes, both depending on the "NULL CHARACTER SWITCH" setting.

When using an input device such as a 2741 terminal which has no provision for entering the non-trivial blank, a special technique is available, through the use of the "NONTRIVIAL BLANK" control card. For example, if the input text contains no "&" characters, then the control card

NONTRIVIAL BLANK REPRESENTED BY 32 (&)

would cause subsequent appearances of &'s to be changed to non-trivial blanks, until the next "NONTRIVIAL BLANK" control card. Thus, if the "NULL CHARACTER SWITCH" is set to 1, the input text

) LLM <----|&&&&&) L &&&&&|---->) LLM

would cause the printed result to appear as shown below.

<----|
|---->

If the "NULL CHARACTER SWITCH" had been set to 2, then the result would have appeared as follows:

<----|
|---->

since the non-trivial blanks at the end of the first group of characters would not be ignored for centering.

Examples of the Usage of Special Operands

For these examples, it is assumed that "SPECIAL KEYPUNCH" and "SPECIAL PRINT TRAIN" are not in effect. Thus the input is in upper case, and the results will be in lower case unless a Command Operand or a Special Operand forces capitalization.

1. $\#1$ produces: 1
2. $\#P\#I\#R\#2$ produces: $PI\#R^2$
3. $)\#PI\#R2)\#$ produces: $PI\#R2$
4. $)\#PI\#R\#2)\#$ produces: $PI\#R^2$
5. $)F\text{TEXT1} \dots \text{TEXTN})F$ produces: Text1 ... Textn
6. $)F\text{TEXT}\#1 \dots T\#EXTN)F$ produces: Text¹ ... Textn
7. $D\#X\#2/D\#2\#Y$ produces: dx^2/d^2y
8. $\#E!10\#2!12\#3!11$ produces: $E^{(2+3)}$
9. $6!$ produces: 6!
10. $6!51$ produces: 6!
11. $\#I\text{WISH}\#I\text{HAD}53\#!\#$ produces: I wish I had 53 $\#!$
12. $\#ONCE\#I\text{HAD}25!43!48\#NOW\text{IT!}42S\text{GONE.}$
produces: Once I had 25 $\#$; Now it's gone.
13. $!52$ produces: 152
14. $!6$ produces: !6
15. $!24!27!25)L|\#X|)L!22!27!23$ produces:

$$\begin{bmatrix} X \end{bmatrix}$$
16. $!14x|x!210!15$ produces: $\{x|x\neq 0\}$

VI. The Editor Facility

The Editor facility can be used to change, override, copy, combine, list, and punch card image data sets; it can also locate words, phrases, and character strings within the text. The card image datasets read and written by the Editor will be called "tape datasets"; these will usually have been created initially by use of the "CREATE A TAPE", "LIST THE INPUT DATASET", or "PUNCH THE INPUT DATASET" control cards. The tape datasets may contain portions of jobs, whole jobs, or multiple jobs. We will refer to the input dataset to be edited as the "old master", and to the resulting output dataset as the "new master". The new master is constructed with all unneeded blanks removed, in the same "condensed" form as a dataset created by the "CREATE A TAPE" control card.

The functions provided by the Editor are requested by using a single Editor control card group, which must be the first and only control card group of the job, and which must be read from the System Input dataset. (See Section VII.) This Editor control card group must begin with the "EDITOR" or "TAPE INPUT" control card, which is then followed by the desired Editor control cards and modifications to the old master (if any), and it must end with the "GO" control card. FORMAT determines from the presence of the "EDITOR" or "TAPE INPUT" control card that an edit phase is to precede the document phase of the run. If the user requests an edit phase, then the subsequent document phase will use the result of the edit phase as its input; in addition, there can be no further document or edit phases. When the end of the Editor control card group is reached, no further reference will be made to the System Input dataset.

An error detected by the Editor means that the newly edited document will not be produced; however, the edit continues in order to detect as many errors as possible. It is clear that user errors make it impossible to know the intention of the user, and FORMAT therefore makes assumptions wherever necessary so that it can continue the edit. Thus, errors detected after the first error may be due to the assumptions made by the program, and not due to the user. Whether or not errors occurred during the edit, FORMAT always gives a listing of all the control card groups used, and a set of diagnostics if any were generated.

At the conclusion of an error-free edit, the document is produced from the new master, unless otherwise specified (see the "\$NO DOCUMENT" Editor control card), and the Editor control card group will appear first when the control card groups are

printed following the document. In the upper far right corner of each page of the document, FORMAT will print the first and last card image numbers from the new master that were used in producing that page. A listing of the latest tape input dataset (as described in the discussion of the "CREATE A TAPE" control card in Section III) is produced after a successful edit if the Editor control card group is begun by the "EDITOR" control card.

The Editor control cards are completely free-form, as described at the beginning of Section III. All Editor control cards (except "GO") begin with a "\$", which distinguishes them from ordinary control cards. During the edit phase, ordinary control cards are simply data to be edited from the old master, or added to the new master. Thus, the "EDITOR" and "TAPE INPUT" control cards are ordinary control cards; they simply initiate the edit phase. Due to a machine-dependent internal storage limitation, no Editor control card operand may exceed 32,767. The Editor control cards are described in the following paragraphs, and some examples of Editor control card groups will be given at the end of this section.

Editing the Old Master

Three Editor control cards are used to modify the tape input dataset (the old master) and produce a new tape input dataset (the new master): they are "\$INSERT", "\$DELETE", and "\$END CHANGES". Before describing the function of each control card in detail, we will give a brief description of the editing process itself.

To perform these functions, FORMAT first reads an Editor control card from the System Input dataset to determine the editing function desired. Material is then copied from the old master to the new master until the Editor finds the position on the old master where the insertion or deletion is to occur; this position is called the "edit point". After deleting material from the old master (if requested), FORMAT inserts new material (if provided) into the new master, until it encounters the next Editor control card.

In this way, FORMAT obeys each of the Editor control cards in turn, reading card images from the old master and writing card images on the new master. Since the old master contains data which can be used in the document phase, it can be read by the Editor in two ways: normal text, which runs freely from card image to card image, and single card images (ordinary control

cards, and "as-is" text cards). Thus, whereas the document phase reads its input in three modes (ordinary text, as-is text, and control card), the edit phase reads the old master and writes the new master in only two modes. These will be called word mode (containing the text of titles and footers, and ordinary text), and card mode (containing as-is text and ordinary control cards). Diagnostic 806 or 814 (see Section XII) is issued if a mode error occurs.

The old master (preferably file-protected) is read from dataset reference number 2, and the new master is written on dataset reference number 4. It is important to remember that the edit phase, unlike the document phase, reads from two sources: from the System Input dataset, which contains Editor control cards and changes to the old master; and from the old master, which is to be edited according to the instructions in the Editor control card group.

We will now describe the three Editor control cards used to perform the editing functions.

\$INSERT BEFORE CARD IMAGE a WORD b

The contents of the cards (if any) between this Editor control card and the next Editor control card are inserted into the new master at the specified edit point, which is determined as follows:

1. If the insertion refers to text or titles (the old master is being read in word mode), then the edit point is just before word b on card image a of the old master (where b is a count of only those words begun on card image a, and must be other than blank or zero).
2. If the insertion refers to "as-is" text or control cards (the old master is being read in card mode), then the edit point is just before card image a, and b must be blank or zero. (In card mode, insertions are made one card image at a time, and do not depend on the words on the card.)

The values of a (card image numbers) and b (numbers of text and title words begun on that card) to be used with the old master are found in the listing produced when the old master was created or last edited.

The cards containing the material to be inserted should be prepared in the same way as ordinary text, title, "as is", or control cards, as though the "CARD FIELD THRU 80" and "029

KEYPUNCH" control cards are in effect. These two control cards also pertain to the new master, because the "026 KEYPUNCH", "029 KEYPUNCH", and "CARD FIELD" control cards will be ignored as insertions. The "CONTROL CARD ENDS IN" control card may be inserted, and it will take effect during the edit.

\$DELETE CARD IMAGE a WORD b

or

\$DELETE CARD IMAGE a WORD b THRU CARD IMAGE c WORD d

The contents of the cards (if any) between this Editor control card and the next Editor control card are inserted into the new master at the edit point. Then, the material in the old master from a,b through and including c,d (if specified) is skipped over, and it will not appear in the new master. The description of the \$INSERT card applies, with the remarks concerning the value of b also applying to the value of d. If it is desired to delete from a,b to the end of the old master, the value 32767 may be given to c to reduce run time (no operand is required for d).

With a single exception, each \$INSERT and \$DELETE Editor control card must refer to an edit point in the old master beyond the last point referenced. The one exception to this rule is that multiple successive \$INSERT references to the same a and b are allowed; the insertions will appear in the same order in the new master.

No \$INSERT or \$DELETE control card can be allowed following a \$DUPLICATE or \$END CHANGES control card, because each of these places the edit point at the end of the old master, beyond which there is no legitimate point. No \$INSERT or \$DELETE control card is allowed in the same Editor control card group with \$MERGE or \$JOIN control cards; that is, changes and merges must be accomplished in separate runs.

\$END CHANGES

This control card is required following the last \$INSERT or \$DELETE control card, unless the end of the old master has been reached. It completes the new master by adding to it the unreferenced last portion of the old master. This control card is ignored when not required (that is, when the end of the old master has been reached).

Combining Data Sets

To combine tape input datasets into a single new tape input dataset (the "new master"):

\$MERGE TAPE INPUT DATASETS ON x1.....x8

The new master is produced at dataset reference number 2 and is an unchanged concatenation of the tape input datasets at the dataset reference numbers given in the operand field, in the order in which they are given. Up to eight dataset reference numbers may be specified in any order, and any may be specified more than once for multiple copies of particular tape input datasets. The valid dataset reference numbers are 9 and higher, and 4. The user must determine that all dataset reference numbers used have been generated into the operating system being used.

As many \$MERGE and \$JOIN control cards as desired may be used. No \$MERGE or \$JOIN control card is allowed in the same Editor control card group with \$INSERT, \$DELETE or \$DUPLICATE control cards; that is, merges must be accomplished in a separate run from changes and duplication.

It is good practice that the tape input datasets be file-protected.

\$JOIN TAPE INPUT DATASETS ON x1.....x8

This control card produces a resultant new master like the one produced by the \$MERGE control card, with one difference: all document-ending "E" Command Operands encountered on the tape input datasets referenced are changed to "V" Command Operands, except for those on the last dataset referenced. The effect of this is to combine the input for many jobs into input for one new job.

The remainder of the description of the \$MERGE control card applies to this control card.

Other Editor Control Cards**\$DUPLICATE OLD MASTER**

The old master is copied from the position at which the last \$INSERT or \$DELETE control card has left it; or, if no position was specified, from the beginning. The old master (preferably file-protected) is mounted at dataset reference number 2, and the copy is written at dataset reference number 4. This control card may be used to complete a new master begun by \$INSERT and \$DELETE control cards.

The \$DUPLICATE control card is not allowed in the same Editor control card group with \$MERGE or \$JOIN control cards.

\$NO DOCUMENT

This control card prevents production of the edited document, which otherwise follows a successful edit run. Any listing, punching, overriding, locating, and dictionary functions that may have been requested concerning the latest tape input dataset are unaffected.

\$OVERRIDE FIRST CONTROL CARD GROUP

The following cards (up to the next Editor or "GO" control card) are control cards which will override the first control card group on the tape input dataset when it is used to produce the document. No "TITLE" or "FOOTER" control card may override. No physical change is made to either master. FORMAT saves the overriding control card group, and uses it as part of the first group read from the just-completed new master at the start of the document phase. The overriding control cards will be inserted just before the first "TITLE", "FOOTER", or "GO" control card in the overridden group.

\$PUNCH

At the conclusion of the successful edit run and after the edited document is produced or bypassed, this control card results in the latest tape input dataset being copied onto the System Punch dataset.

\$LIST

This control card forces a listing of the new master (if any) at the conclusion of an edit run, successful or not. Following an unsuccessful edit, the listing is in upper case. In the listing, the Command Operands that appear on each card image are reiterated alongside the card images, in the right-hand portion of the page. This allows one to find Command Words rapidly, and to locate desired areas of the input text. Those symbols for which no graphics are expected are printed as asterisks in the listing.

\$OMIT LISTING OF NEW MASTER

The presence of this control card in the group of Editor control cards will suppress the listing of the new master following a successful edit. The default action is to produce the listing.

The production of a listing depends on a number of factors. If the Editor control card group was begun with the EDITOR control card, then a listing will be produced only if the edit was successful (in the absence of a \$LIST Editor control card). If the Editor control card group was begun with the TAPE INPUT control card, then a listing is provided only if the \$LIST Editor control card is included in the Editor control card group. The \$OMIT LISTING Editor control card always deletes the listing.

FORMAT will usually diagnose editing errors so that the cause of the error can be identified readily. If errors are expected, it is sometimes helpful to include the \$LIST Editor control card in the Editor control card group; the listing can then be scanned to see what actions were taken by FORMAT in handling the errors.

\$LOCATE THE FOLLOWING WORDS/PHRASES/STRINGS

This facility is intended primarily to assist in the task of index production; see the "DICTIONARY" control card also. The following cards (up to the next Editor or "GO" control card) contain arguments to be located (by card image number) in the latest input stream, according to the following rules:

- one search argument per card
- non-alphameric not b+/*\$ (b = blank) are ignored both in search arguments and in the text stream

- blanks are word delimiters only, both in search arguments and in the text stream
- a final non-blank character of "+" in a search argument means that all strings consisting of the preceding characters are to be located
- all blank search arguments, duplicate search arguments, and arguments consisting of a single "+" are ignored; a "+" in a search argument is ignored if the preceding string consists solely of a single character
- search arguments may be in any order
- a non-trivial blank in the input stream is treated as an ordinary blank, but a non-trivial blank in a search argument is not changed; thus, no strings can be located that match a search argument containing a non-trivial blank
- only ordinary and "as-is" text are searched on the input stream; Command Words, control cards, and titles are not
- comparisons are made on an upper case basis; if text or search arguments contain lower case letters, they are converted to upper case for the comparison
- dataset reference number 3 is required (see Section IX)

FORMAT scans the input text for words and strings that match a search argument, and accumulates as much data as it can hold before writing any output. When its tables are full (or when all the input text has been scanned), the program writes the results on the System Output dataset in alphabetic order, for that section of the input text, with the locations of the matching strings in ascending order of input card image number. The scan of the input text then begins again, if necessary.

A search argument with a non-letter as one of the first two characters is positioned at the beginning of the entries for the letter of the first two characters. Thus, the located strings which match "A*" and "*A" would both be found at the start of the list of search arguments beginning with the letter "A". A search argument which cannot be found is so annotated.

If the number of \$LOCATE arguments is too large, FORMAT will print a message on the System Output dataset, giving the number of the \$LOCATE argument which caused the table overflow. It and the remaining arguments can then be located in a subsequent computer run.

Examples of Editor Control Card Groups

```
EDITOR
$DELETE 10 5          (delete a single text or title word)
      \  THE COMPUTER (inserted text)
$INSERT 15            (insert before control card)
      LINES/PAGE = 70 (inserted control card)
$DE 16 0 18 6        (delete control cards and text)
      GO              (inserted control card)
      ) P THE DATA  (inserted text)
$END CHANGES
$PUNCH
$LIST                (force listing of new master)
GO
```

(This group will produce only a dictionary)

```
EDITOR
$DUPLICATE OLD MASTER
$NO DOCUMENT
$OVERRIDE AND PROVIDE A
      DICTIONARY      (override is just this one card)
GO
```

```
TAPE INPUT DATASET
$OVERRIDE CONTROL CARDS
      COLUMNS/PAGE = 2
      SPECIAL PRINT TRAIN
      LIST THE TAPE
$LOCATE THE FOLLOWING:
      TAPE+           (locates "tape", "tapestry", etc.)
      CONTROL CARD+
      OLD MASTER
      CARD IMAGE+
      COLUMN-LINE
      RIGHT JUSTIFICATION
      TEXT PROCESSING PROG+
GO
```

(This group combines 3 tapes and punches the result)

```
EDITOR
$PUNCH
$OMIT THE LISTING OF THE NEW MASTER
$JOIN TAPE DATASETS 4, 10, AND 9
GO
```

VII. Rules for Using FORMAT

A. General:

1. Each FORMAT job must begin with a control card group (the minimum control card group consists of the "GO" control card).

2. Title, footer, and text input must appear in the field specified on the "CARD FIELD" control card, or, if not used, in the default card field (card columns 1 through 80).

3. A FORMAT job is ended by the appearance of the "E" Command Operand. Multiple FORMAT jobs may be stacked one behind the other. If mixed card and tape resident jobs are to be run, the card jobs must precede the tape jobs (including an edit job); the card jobs must not use dataset reference number 2, however.

4. Command Words may appear freely interspersed throughout text and titles. Although no text or title word may begin with a ")" character [nor the appropriate "!nn" configuration for ")"], the ")" may be used textually when followed by a blank.

B. Titles and Footers:

1. The card images containing the title must immediately follow the "TITLE STARTS ON" control card and must be immediately followed by either the "FOOTER" or the "GO" control card.

2. The card images containing the footing title must immediately follow the "FOOTER" control card and must be immediately followed by either the "TITLE" or "GO" control card.

3. The "L", "F", "Z", and "E" Command Operands may be used in titles.

4. The text of a title or footer must be ended by the Command Operand "E".

5. The "L" Command Operand always acts as if single spacing were in effect, regardless of the operand field on the "SPACING OF TEXT LINES" control card (or its default). The "E" Command Operand, in addition to ending the title, also single spaces. Thus:

TITLE LINE 1) LL TITLE LINE 2

results in exactly one blank line between title lines, while

LAST TITLE LINE) LLE

results in a minimum of two blank lines separating the last title line from the body of the document.

6. Each title line begins in the print position specified (or the default, position 1) and ends when a Command Word containing either the "L" or "E" Command Operand is encountered, or else when the title line attempts to exceed the last printer position allowed to the line.

7. No right-justification is accorded to titles, since no right-most title limit is defined.

8. All hyphens appearing in titles are printed. Excess blanks are ignored. Special spacing may be achieved with non-trivial blanks.

9. The Special Operands may be used in titles.

C. Body of the Document:

1. Input blanks between words serve only as word delimiters (unless operating in the "as is" mode). Words are separated by a single blank, plus the number of blanks required to accomplish right-justification, if in effect (see the "JUSTIFICATION" control card for details).

2. Hyphens are not automatically introduced by FORMAT. A hyphen in the input stream is printed, and may be selected to be the last character on a column-line.

VIII. Summary of FORMAT Control Cards and Command Operands

The control cards are grouped below by the options to which they refer. Thus, the "JUSTIFICATION" and "NO JUSTIFICATION" control cards are paired because each refers to the right-justification option. Within each group certain default values will be assumed if no control card from that group is used.

Control Cards

If Omitted

BACKSPACE CHARACTER IS SPECIAL CHARACTER nnno backspaces

BETWEEN COLUMNS LEAVE x BLANKSx=2

CAPITALIZE AUTOMATICALLY

NO CAPITALIZATION AUTOMATICALLYassumed

CONTROL CARDS END IN COLUMN xx=80

CARD FIELD IS x THRU yx=1, y=80

CARD FIELD EXTENDS THRU y

CENTER TEXT ON LINE xx=5

START TEXT ON LINE x IN PRINT POSITION y

TEXT STARTS ON LINE x IN PRINT POSITION y

COLUMNS PER PAGE = xx=1

COPIES = xx=1

DARK PRINT EACH PAGE x TIMESx=1

OUTPUT MEDIUM IS TAPE

PRINT OUTPUT TAPE

CREATE A TAPE FROM CARD INPUT

LIST THE INPUT DATASET

PUNCH THE INPUT DATASET

CYCLE THE PAGE NUMBER

LEFT TOP POSITION FOR PAGE NUMBER

PAGE NUMBER STARTING AT xx=1

RIGHT TOP POSITION FOR PAGE NUMBERassumed

DICTIONARY OF WORDS USED

DROP CHARACTER FOR 'D' COMMAND IS xx=75 (dots)

TAPE INPUT DATASET

EDITOR

\$INSERT
\$DELETE
\$END CHANGES
\$MERGE TAPES
\$JOIN TAPES
\$DUPLICATE OLD MASTER
\$NO DOCUMENT
\$OMIT LISTING OF NEW MASTER
\$OVERRIDE
\$PUNCH
\$LIST
\$LOCATE

FOOTER ON LINE x POS'N y AFTER z BLANK LINES

TITLE STARTS ON LINE x IN PRINT POSITION y

GOerror

INDENT COLUMN (x1,y1).....(x7,y7) POSITIONSx's,y's = 0

JUSTIFICATIONassumed

NO JUSTIFICATION

LINES PER PAGE ARE xx=59

NONTRIVIAL BLANK REP'D BY SPECIAL CHAR nnnn=0

NULL CHARACTER SWITCH SET TO xx=1

PARAGRAPH INDENT IS x PRINT POSITIONSx=5

REPEAT TITLE ON EVERY PAGE

STOP PRINTING TITLEassumed

SENTENCES SEPARATED BY AT LEAST x SPACESx=1

SEPARATION LINES BETWEEN PARAGRAPHS ARE xx=1

SIDE BY SIDE COPIES

SPACING OF TEXT LINES IS xx=1

SPECIAL KEYPUNCH

SPECIAL KEYPUNCH IS A 2741

SPECIAL PRINTER TRAIN

TABS ARE SET AT x1.....x14tabs set to 0UNDERLINE SWITCH SET TO xx=0WIDTH OF COLUMNS IS x PRINT POSITIONSx=64

026 KEYPUNCH

029 KEYPUNCHassumed

```
A -- enter "as is" mode
C -- begin a new column
D -- tab to next tab stop, dropping dots
Dn-- tab to n-th tab stop, dropping dots
E -- end the title or the footer, or end the job
F -- capitalize first letters of words / stop
Hn-- indent (delayed) column using nth pair / restore
In-- indent (now) column using nth pair / restore
J -- always begin a new column-line
K -- keep the enclosed text in one text column
L -- begin a new column-line when not at top of column
M -- center text within the column-line / stop
P -- begin a new paragraph
S -- begin a new page
T -- tab to next tab stop
Tn-- tab to n-th tab stop
U -- underline / stop underlining
V -- read in the next group of control cards
Wn-- keep the next n lines in the same column
Z -- print in upper case only / stop
```

```
n 11111111122222222233333333344444444455
n 012345678901234567890123456789012345678901
```

The EBCDIC card code for "ø" is: 12-8-2
The EBCDIC card code for "!" is: 11-8-2

IX. Datasets Used by FORMAT

The correspondence between dataset reference numbers (DRN) and system dataset names (which are used in the name field of system control cards) is as follows:

DRN	OS/360
1	FT01F001
2	FT02F001
3	FT03F001
4	FT04F001
5	FT05F001
6	FT06F001
7	FT07F001
8	FT08F001
above	FTxxF001

Dataset reference numbers 5, 6, and 7 are assumed to apply respectively to the System Input dataset, the System Output dataset, and the System Punch dataset.

The user must verify that the dataset reference numbers he uses are in fact available; i.e., that they have been generated into the operating system in use at his installation.

All datasets created and used by FORMAT are formatted, sequential, and fixed length, and may be defined as blocked, if operating under Release 18 or later releases. This restriction in earlier releases is due to Data Management's inability to backspace a blocked dataset and not to the logic of this program. If blocked datasets are used with releases prior to 18 the results will be unpredictable.

Labeled tapes can be used by the program providing that they are acceptable to the operating system used. Labeling of tapes, if desired, is the responsibility of the user.

Before a file-protected tape can be read by OS/360, the message "xx IEC103D P" is typed on the console. The operator must respond with "reply xx,'U'", where xx is the on-line message number.

Under OS/360 the number of I/O buffers may be 1 or 2. The higher number is always preferable unless there is difficulty fitting the program into memory, in which case the number 1 should be specified where necessary; however, performance may be somewhat degraded.

All datasets created by FORMAT are ended by an "end-of-file" mark.

The following describes the datasets created and used by FORMAT:

Dataset Reference Number 1:

This dataset records control cards, user errors, and other information, and is always required. It may be direct access device or tape resident. Its record length is 97 bytes.

Dataset Reference Number 2:

This dataset is required only if one or both of the following apply:

1. "EDITOR", "TAPE INPUT", "CREATE A TAPE", "DICTIONARY", "LIST", and/or "PUNCH" has been specified
2. The Command Operand "K" has been used

This dataset is a card image set which may be resident either on tape or on a direct access device. If the Editor facility is being used, tape is preferable since the user may wish to keep this dataset, file-protect it, and use it again as an input dataset master.

Dataset Reference Number 3:

This dataset is required only if "DICTIONARY" or "\$LOCATE" has been specified. It contains 80 bytes per record and may be tape resident or (preferably) on a direct access device.

Dataset Reference Number 4:

This dataset is only required when producing a "new master" input dataset (or a duplicate of the "old master") in an edit run. It may also be (but not in the same run) an input dataset to be \$MERGED or \$JOINED in an edit run. Its specifications are identical to those for dataset reference number 2.

Dataset Reference Number 5:

This is the System Input dataset and is always required by the program. Its record length is always 80 bytes.

Dataset Reference Number 6:

This is the System Output dataset and is always required by the program. Its record length is 133 bytes, and ASA standard control characters are used.

Dataset Reference Number 7:

This is the System Punch dataset, and is only required if punched output has been requested. Its record length is always 80 bytes.

Dataset Reference Number 8:

This dataset is only required by FORMAT if any of the following control cards is specified:

OUTPUT IS TAPE
COPIES = 2 (or more)
PRINT OUTPUT TAPE

This dataset is a printer image (133 bytes per record) set which can be tape or direct access device resident.

Dataset Reference Numbers Above 8:

These may be used as input datasets to be \$MERGED or \$JOINED in an edit run. The specifications for these datasets are identical to those for dataset reference number 2.

2

X. Description of FORMAT for OS/360 and Suggested Control Cards

The distributed System/360 FORMAT object deck (produced by the Fortran H compiler) is set up to run as an OS/360 overlay job (the OVERLAY cards are included in the deck, but may be removed to run FORMAT in-line). As an overlay job it requires 48,648 (hex BE08) bytes of memory, including the subroutines from the full Fortran library of OS/360 Release 18 with the Fortran Extended Error Handling facility (but not including I/O buffers). FORMAT requires a minimum 64K System/360 or System/370 computer. In non-overlay form FORMAT requires a minimum of 79,272 (hex 135A8) bytes; it will run somewhat faster because fewer I/O operations will be required.

A suggested FORMAT run setup is as follows (note that the asterisks along the right margin are supposed to appear in column 72 of the JCL statements). The blocksizes for Dataset Reference Numbers 1, 3, and 8 were chosen to optimize storage space usage on a 2314 Direct Access Storage Facility.

```
//FORMAT JOB
//LKED EXEC PGM=IEWL,PARM='OVLY,XREF,LIST'
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=SYS1.FORTLIB,DISP=OLD
//SYSUT1 DD DISP=(,DELETE),UNIT=2314,SPACE=(CYL,(3,2))
//SYSLMOD DD DSN=GOSET(MAIN),DISP=(NEW,PASS),UNIT=2314,
//          SPACE=(TRK,(12,2,2)),VOLUME=SER=
//SYSLIN DD *
```

Distributed OS/360 FORMAT object deck
--

```
/*
//GO EXEC PGM=*.LKED.SYSLMOD
//PT06F001 DD SYSOUT=A
//PT07F001 DD UNIT=SYSCP
//PT01F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(3,1)),
// DCB=(RECFM=FB,LRECL=97,BLKSIZE=7275,BUFNO=2)
//PT03F001 DD UNIT=2314,DISP=(,DELETE),SPACE=(CYL,(6,1)),
// DCB=(BUFNO=2,RECFM=FB,LRECL=80,BLKSIZE=7280)
//PT02F001 DD UNIT=(____,DEFER),LABEL=(,NL),
// VOLUME=(,RETAIN,,,SER=OLDMAS),
// DCB=(BUFNO=2,RECFM=FB,LRECL=80,BLKSIZE=8000)
//PT04F001 DD UNIT=(____,DEFER),LABEL=(,NL),
```

```

//          VOLUME=(,RETAIN,,,SER=NEWMAS),          *
//          DCB=(BUFNO=2,RECFM=FB,LRECL=80,BLKSIZE=8000)
//FT08F001 DD UNIT=(_____,DEFER),LABEL=(,NL),        *
//          VOLUME=(,RETAIN,,,SER=OUTPUT),          *
//          DCB=(BUFNO=2,RECFM=FBA,LRECL=133,BLKSIZE=3458)
//FT05F001 DD DATA

```

FORMAT job(s)

/*

Of the datasets defined above, only FT01F001, FT05F001, and FT06F001 are always required. See Section IX for more information.

The FORMAT distribution tape consists of three files written at a recording density of 800 BPI on a 9-track tape, with no labels. All logical records are 80 bytes long, and each physical record is 1600 bytes long. The first file contains the object deck (including Linkage Editor control statements); the second file contains the FORMAT job which produces this manual; the third file contains the Fortran source statements from which the object deck was produced.

XI. Hints and Suggestions

A. Document Phase

1. The TITLE and FOOTER control cards, along with their following title and footer texts, must be the last control cards to appear in a control card group before the GO control card.

2. If the text for a title or footer is not ended with the ")E" Command Operand, FORMAT will search for it by including as much of the following material as possible into the "title". This naturally leads to a document of unusual proportions.

3. When ending an "as-is" region (initiated by the "A" Command Operand), the card containing the ") " in the initial columns should contain no other text.

4. If an erroneous control card is found, it is treated by FORMAT as a "GO" card. This means that any following control cards will be read in text mode; in particular, if a "TITLE" card follows the bad control card, the ")E" that ends the title (or footer, of course) will appear to be the ")E" that ends the text input.

5. When setting up tab stops and column indents, remember that a tab stop in (say) column 10 is equivalent to an indent of 9 spaces -- that is, the line position where the text will begin after indenting is 1 larger than the number of spaces indented.

B. Edit Phase

1. A successful edit does not imply a successful document, since conflicting information may have been edited into the new master.

2. Control cards written onto the new master are under control of the

CONTROL CARD ENDS IN COLUMN nn

card currently in effect.

3. During an edit, the method used to search for Editor control cards can occasionally cause a non-control card to be mistaken for an Editor control card. (During an edit, each card in the Editor control card group must be checked to see if it is an Editor control card, or text to be inserted into the new master.) The valid Editor control card characters are shown in the leftmost column of the table below; the invalid combinations that will be mistaken for the valid combinations are shown in the right columns.

<u>Valid</u>	<u>Invalid</u>
\$ME	\$J5 \$KV \$LN
\$OV	\$N5 \$PN \$QE
\$DE	\$A5 \$CN \$BV
\$IN	\$G5 \$HV
\$EN	\$C5 \$FE \$DV
\$DU	\$C4 \$EM \$FD
\$PU	\$O4 \$QM \$RD
\$NO	\$L6 \$MW \$OE
\$LI	\$KR \$JZ
\$LO	\$KW \$J6
\$JO	\$KF
\$OM	\$M4 \$NU \$PD

Editor Control Card Equivalences

To avoid such errors, (1) arrange the text to be inserted so that the first nonblank character on the input card is not a "\$", or (2) be sure that the first three characters are not one of the invalid combinations.

4. The DICTIONARY feature requires that the source text be in upper case; text entered with the "SPECIAL KEYPUNCH" control card in effect may not be processable by this facility.

XII. Error Handling and Diagnostic Messages

With the exception of errors made during an edit run, user errors do not abrogate the document. When a user error is found, the program notes the error, assumes appropriate values for the erroneous data, and continues. The Editor does not allow a document to be produced unless the edit was error-free; however, the edit itself continues to completion regardless of user errors.

The error diagnostics (if any) are written onto the System Output dataset at the conclusion of each job. Each diagnostic consists of a textual description of the error and the page number, column number, and line number being produced when it occurred. If the error was in the input text, the character number within the line where the error occurred is given; if the error occurred within a control card group, then the group number is given; and if the error can be localized to a particular control card or Editor insertion card, then the card number is given. Also listed for each error is a code number that refers to a paragraph below, which gives additional information about the error and describes action taken by the program when it occurs.

212. CONTROL CARD OPERAND IN ERROR

An operand on the control card specified is outside the legal range or is otherwise in error. If the error occurs on an Editor control card, the control card is ignored. Otherwise, the previous value of the parameters involved or, if none, the default values are used.

218. UNRECOGNIZED CONTROL CARD

The specified control card is unrecognizable. It is treated as if it were the "SPECIAL PRINTER" and the "GO" control cards. If the input stream is not on the System Input dataset, it is backspaced and the unrecognizable control card is reread as text.

219. NUMBER OF PRINT POSITIONS REQUIRED NOT AVAILABLE

The number of print positions required by this control card group exceeds the number available. The document is forced leftward, the width of the text columns may be redefined to be the largest value possible, and the number of print positions between columns may be set to 2.

220. TITLE/FOOTER TOO LONG

The title or footer is not ended after the last line allotted to the page is filled. The title or footer is ended and the program looks for a control card. If issued for a title, the "STOP TITLE" control card is simulated.

237. TABS NOT IN ASCENDING ORDER

The tabs set in the specified control card group are not in ascending order. Starting with the first tab set out of order, the tabs are set to the last position on the column-line.

249. CONTROL CARD NOT FIRST, OR ON DATASET OTHER THAN 5

The "EDITOR" or "TAPE INPUT DATASET" control card specified is either not the first card of the job, or else it has been read from a dataset which is not the System Input dataset (dataset reference number 5). It is ignored, and any following Editor control cards or insertions will not be properly interpreted.

267. TAB IMPROPERLY SET

In the specified control card group a tab is set at a position beyond the end of the column-line. The erroneously set tab and the tabs which follow it are set to the last position on the column-line.

269. IMPROPER STARTING LINE FOR DOCUMENT TEXT

The body of the document is positioned improperly by the specified control card group. The corrective action taken is to begin the text immediately following the title (but not above line 5), and the text is extended through the last line on the page.

289. IMPROPER CONTROL CARD ORDER

The referenced control card is neither the "TITLE", "FOOTER", nor "GO" control card. An attempt is made to allow the present control card order.

300. INDENTS TOO LARGE

The cumulative indents in effect have reduced the effective column-line width to zero or less. All column indents are turned off at the indicated character position.

304. CHARACTER STRING LENGTH EXCEEDS COLUMN WIDTH

A string of non-blank, unhyphenated characters at the indicated character position is longer than the column-line. It is printed without hyphenation over as many lines as are required to contain it.

327. **TAB COMMAND OPERAND IMPROPERLY USED**
The Command Operand "T" or "D" at the indicated character position is beyond the position of any tab set, or is not to the right of the current character position, or is in an indented portion of the column-line, or an unset tab has been used. The Command Operand is ignored.
513. **NUMBER OF UNDERLINE SEGMENTS ON PAGE EXCEEDS 99**
At the indicated character position more than 99 column-lines, or portions of column-lines, have been underlined on this document page. Those in excess of 99 are ignored.
700. **UNDEFINED COMMAND OPERAND**
A Command Word at the indicated character position (before the line is justified) contains an undefined Command Operand. It, and the rest of the Command Word, are treated as text. The ")" is also printed if the undefined Command Operand is the first in the Command Word. If the error was detected during an edit, the number given for the erroneous control card will be that of the last one read before the error was detected. If the invalid Command Operand is a ")", then FORMAT will treat it as the start of a new Command Word if it is not followed by a blank.
800. **UNEXPECTED END OF INPUT**
An unexpected end of the input stream has occurred, caused by an omitted "GO" or "\$END CHANGES" control card, or by an omitted "E" Command Operand. Some output may be lost.
802. **NEW MASTER ALREADY FINISHED**
An attempt has been made by the specified control card to continue the new master after the end of the old master has been reached. For example, a "\$DELETE" control card may have occurred after a "\$DUPLICATE" control card. The edit continues.
804. **EDIT FAILED BECAUSE OF ABOVE ERROR(S) OR BECAUSE NEW MASTER NOT FINISHED**
Errors already noted have occurred during the edit ended by the referenced card, or else the new master has not been ended because the end of the old master has not been reached or referenced. The job is terminated.
805. **REFERENCED WORD NOT LOCATED**
The word referenced on the specified "\$INSERT" control card, or the first word referenced on the specified "\$DELETE" control card cannot be located. The edit continues.

806. INPUT/OPERAND MODE ERROR

The mode of the operand on the specified "\$INSERT" or "\$DELETE" control card differs from the present mode of the new input dataset (new master). That is, a word number is specified and the new master is in an "as is" or control card region (card mode), or no word number is specified and the new master is in a text region (word mode). The edit continues.

807. END OF \$DELETE FIELD NOT FOUND

The end of the field to be deleted, referenced on the specified "\$DELETE" control card, cannot be located. The edit continues.

814. NON-TEXT MODE NOT ENDED

An "as is" or control card region edited into the middle of a text card image has not been ended before the specified control card. The edit continues in "text" (word) mode.

847. \$INSERT/DELETE/DUPLICATE AND \$MERGE/JOIN NOT ALLOWED IN SAME RUN

Editor control cards "\$INSERT", "\$DELETE", and "\$DUPLICATE" may not appear in the same run with "\$MERGE" and "\$JOIN" control cards. That is, merges must be accomplished in a separate run from changes and duplication. The indicated control card is in violation of this rule. The edit continues.

857. NOT ALLOWED

The "TITLE" or "FOOTER" control card indicated is not permitted as an overriding control card. The edit continues.

922. NO TEXT AFTER TAB(S)

The last tab on the indicated line is not followed by text.

997. TOO MANY BACKSPACES ON ONE PAGE

Too many backspaces have been specified on the current page. The first 99 have been handled, but any after the 100th will be treated as normal text characters.

XIII. Appendix

The following pages were produced at the conclusion of the computer run producing this manual. The "COLUMNS PER PAGE = 9" control card, the first control card in the first control card group, is intentionally faulty, and produces the first diagnostic. The fifth example of Command Words (at the end of Section IV) produces the second.

INDEX

A

ASA, 52
 Asterisk, 12, 41
 "As is", 24, 4, 7, 12, 18, 25,
 28, 31, 37, 42, 45, 55,
 60, see also Command
 Operand "A"
 "As is" text mode, 4, 7, 24,
 31, 60

B

"BACKSPACE CHARACTER", 9, 10,
 23, 27, 46, 60
 Backspacing of files, 50, 57
 "BETWEEN COLUMNS", 10, 21, 46
 Blank, 1, 2, 4-12, 16, 27, 41,
 42, 45, see also Non-
 trivial blank
 Blank lines, 14, 16, 19, 26,
 45
 Blocked datasets, 50

C

Capitalization, see Special
 Operand "z", Command
 Operands "z" and "F", and
 "CAPITALIZE AUTOMATICALLY"
 "CAPITALIZE AUTOMATICALLY",
 11, 26, 27, 46
 "CARD FIELD", 11, 12, 37, 38,
 44, 46
 Centering, 7, 17, 20, see also
 Command Operand "M" and
 "CENTER TEXT ON"
 "CENTER TEXT ON", 11, 46
 "COLUMNS PER PAGE", 11, 21,
 46, 61
 Command Operands, 2, 4-8, 12,
 24, 28, 29, 41, 49
 "A", 24, 28, 55
 "C", 24, 28
 "D", 24, 14, 27, 28, 59

"E", 25, 7, 21, 28, 39, 44,
 45, 55, 59
 "F", 25-28, 44
 "H", 25, 6, 7, 28
 "I", 25, 6, 28
 "J", 25, 28
 "K", 26, 10, 27, 28, 51
 "L", 26, 5, 25, 28, 44, 45
 "M", 26, 7, 17, 28
 "P", 26, 5, 11, 25, 27, 28
 "S", 26, 11, 25, 27, 28
 "T", 25-28, 59
 "U", 27, 17, 26, 28
 "V", 27, 5, 25, 28, 39
 "W", 27, 28
 "z", 27, 28, 31, 44

Command Word, 24, 2, 4, 7, 22,
 41, 42, 44
 Concatenation, 39
 Condensing, 12, 18, 35
 Console message, 50
 "CONTROL CARDS END IN", 11,
 38, 46, 55
 Control card group, 4, 6, 9,
 15, 27, 35-41, 43, 44, 49,
 55-57, 61
 Control card mode, 4, 5, 7
 Control variables, 4, 5, 7
 "COPIES", 12, 17, 18, 46, 52
 "CREATE A TAPE", 12, 16, 18,
 35, 46, 51
 "CYCLE THE PAGE NUMBER", 13,
 46

D

"\$DELETE", 38, 36, 39, 40, 47,
 59, 60
 "\$DUPLICATE", 40, 38, 39, 47,
 59, 60
 "DARK PRINT", 13, 23, 46
 Dataset reference number 1, 51
 Dataset reference number 2,
 51, 12, 14, 20, 37, 40, 44
 Dataset reference number 3,
 51, 14, 42
 Dataset reference number 4,
 51, 37, 39, 40

Dataset reference number 5,
52, 58, see also System
Input Dataset
Dataset reference number 6,
52, see also System Output
Dataset
Dataset reference number 7,
52, see also System Punch
Dataset
Dataset reference number 8,
52, 17
Diagnostic messages, 57, 4, 7,
18, 27, 29, 37, 61
"DICTIONARY OF WORDS USED",
13, 40, 41, 47, 51, 56
Document phase, 4, 55
Dots, 24
"DROP CHARACTER", 14, 24, 47

E

"\$END CHANGES", 38, 36, 47, 59
EBCDIC, 12, 14, 19, 32, 49
Edit phase, 4, 14, 55
Editor, 35, 14, 13, 20, 56, 57
"EDITOR", 14, 20, 41, 47, 51,
58
Escape character ")", 4, 5,
12, 22, 24, 31, 44, 59

F

"FOOTER ON LINE", 14, 40, 44,
47, 55, 58, 60
Fortran, 2, 53, 54
Free-form, 1, 4, 5, 9
FT01F001, 50, 53, 54
FT02F001, 50, 53
FT03F001, 50, 53
FT04F001, 50, 53
FT05F001, 50, 54
FT06F001, 50, 53, 54
FT07F001, 50, 53
FT08F001, 50, 54

G

"GO", 15, 4, 6, 9, 35, 36, 40,
44, 47, 55, 57, 58, 59

H

Hanging indent, see Command
Operand "H"
Hyphen, 45
Hyphenation, 2, 15, 58

I

"\$INSERT", 37, 36, 38-41, 47,
59, 60
Immediate indent, see Command
Operand "I"
"INDENTATION OF THE COLUMN",
15, 6, 25, 47
Index, 1, 14, 41
I/O buffers, 50, 53

J

"\$JOIN", 39, 38, 47, 51, 52,
60
"JUSTIFICATION", 15, 45, 57

K

Keep, see Command Operand "K"
Keypunch, 19, 22, 24, 28, 31,
33, 34, 47, 56

L

"\$LIST", 41, 47
"\$LOCATE", 41, 8, 14, 42, 47,
51
"LEFT TOP POSITION FOR PAGE
NUMBER", 15, 46

"LINES PER PAGE", 16, 21, 47
 "LIST THE INPUT", 16, 35, 46,
 51

M

"\$MERGE", 39, 38, 47, 51, 52,
 60
 Memory, 50, 53
 Mode, see As-is text mode,
 Normal text mode, Control
 card mode, Output mode,
 and Editor
 Multi-punching, 31, 33

N

"\$NO DOCUMENT", 40, 35, 47
 "NONTRIVIAL BLANK", 16, 33, 47
 Non-trivial blank, 33, 16-18,
 42, 45
 "NO CAPITALIZATION
 AUTOMATICALLY", 16, 11, 46
 "NO JUSTIFICATION", 16, 15,
 18, 23, 47
 Normal text mode, 4, 5, 7, 15
 "NULL CHARACTER SWITCH", 17,
 27, 33, 47

O

"\$OVERRIDE", 40, 47
 "\$OMIT LISTING", 41, 47
 Object deck, 53, 54
 Operator, 50
 OS/360, 2, 50, 53
 "OUTPUT MEDIUM IS TAPE", 12,
 46, 52
 Output mode, 2
 Overlay, 53
 Overprinting, see "BACKSPACE
 CHARACTER" and "DARK
 PRINTING"

P

"\$PUNCH", 40, 46, 47
 "PAGE NUMBER STARTING AT", 17,
 46
 Paragraph, 3, 5, 11, 16, 17,
 19, 26, 29, 30, see also
 Command Operand "p"
 "PARAGRAPH INDENT", 17, 26, 47
 Parenthesis, see Escape
 Character
 Performance, 50
 Period, 14, see also Dots
 Phase, see Edit Phase and
 Document Phase
 Plus sign, 41, 42
 Printer, 19
 "PRINT OUTPUT TAPE", 17, 46,
 52
 Printer train, see "SPECIAL
 PRINTER TRAIN" and TN
 Print Train
 "PUNCH THE INPUT", 18, 35, 51

R

Record length, 12, 54
 "REPEAT TITLE", 18, 20, 47
 Reread, 57
 Restriction, 50
 "RIGHT TOP POSITION FOR PAGE
 NUMBER", 18, 46

S

Search, 41
 Search argument, 41, 42
 Segments, 14, 59
 "SENTENCES SEPARATED BY", 11,
 18, 47
 "SEPARATION LINES BETWEEN
 PARAGRAPHS", 19, 26
 Sequential, 50
 Setup, 53

"SIDE BY SIDE COPIES", 19, 21, 47
 "SPACING OF TEXT LINES", 19, 26, 44, 47
 Special characters, 1, 2, 9, 14, 16, 31
 "SPECIAL KEYPUNCH", 19, 14, 31, 47, 56
 Special Operands, 31, 4, 6, 8, 24, 34, 49
 "ø", 31, 6, 19, 33, 34
 "!", 31-34
 "SPECIAL PRINTER TRAIN", 19, 12, 48, 57
 Spelling, 14
 Stacked, 7, 25, 44
 "START TEXT ON", see "TEXT STARTS ON"
 "STOP PRINTING TITLE", 20, 47, 57
 String, 24, 31, 35, 41, 42, 58
 Subscripts, 2
 Superscripts, 19, 28, 31
 System Input Dataset, 2, 12, 14, 20, 36, 37, 50, 52, 57, 58
 System Output Dataset, 2, 9, 13, 17, 42, 50, 52, 57
 System Punch Dataset, 2, 18, 40, 50, 52

T

"TABS ARE SET AT", 20, 23, 48
 Tabulation, see Command Operands "D" and "T" and "TABS ARE SET AT"
 "TAPE INPUT DATASET", 20, 35, 36, 41, 47, 51, 58
 Tape labels, 50, 54
 "TEXT STARTS ON", 20, 21, 23
 "TITLE STARTS ON", 20, 23, 40, 44, 46, 47, 55, 58, 60
 TN Print Train, 2, 32, 49
 Translation, 19, 31

U

Underlining, 17, 21, 27, see also Command Operand "U"

W

Widows, see Command Operand "W"
 "WIDTH OF COLUMNS", 21, 48

"026 KEYPUNCH", 22, 12, 38, 48
 "029 KEYPUNCH", 22, 12, 38, 48

FORMAT RELEASE 5 CONTROL CARDS

GROUP	PAGE	COLUMN	LINE	NO.	
1	1	1	0	1	COLUMNS PER PAGE = 9 (SEE 13 5; SEE FOLLOWING DIAGNOSTIC. WILL SET 1 COLISE)
				2	START TEXT ON LINE 14, PP 12
				3	CAP OPTION
				4	LINES/PAGE = 56
				5	WIDTH OF COLUMNS IS 60 CHARACTERS
				6	PAGE NUMBER IS NULL
				7	NONTRIVIAL BLANK IS REPRESENTED BY SPECIAL CHARACTER 30 (-)
				8	INDENT THE COLUMN (8,3), (5,5), (5,3), (3,0), (6,0), (8,0), (2,0)
				9	GO
2	3	1	14	10	START TEXT ON LINE 8 IN POSITION 10
				11	WIDTH OF COLUMNS = 64
				12	REPEAT TITLE
				13	TABS AT 8 AND 63
				14	TITLE: LINE 1, PRINT POSITION 25
				15	GO
3	4	1	8	16	BACKSPACE CHARACTER IS NUMBER 46 (0)
				17	CYCLE PAGE NUMBERS
				18	RIGHT SIDE PAGE NUMBER TO START
				19	PAGE NO. = 1
				20	GO
4	16	1	28	21	NO JUSTIFICATION
				22	GO
5	16	1	34	23	JUSTIFICATION
				24	BACKSPACE CHARACTER RESET TO 0 (TURN BACKSPACING OFF)
				25	GO
6	21	1	29	26	BACKSPACE CHARACTER IS 50 (?)
				27	GO
7	21	1	34	28	BACKSPACE 0 (RESET)
				29	GO
8	31	1	56	30	TABS ARE SET AT 20 25 30 36
				31	NULL CHARACTER SWITCH SET TO 2 (USE NONTRIVIALS FOR CENTERING)
				32	GO
9	46	1	8	33	TABS ARE AT 10 AND 50
				34	GO
10	50	1	15	35	TABS SET TO 24 33
				36	GO
11	52	1	33	37	TAB SET TO 10
				38	GO
12	56	1	22	39	TABS AT 22, 32
				40	GO

FORMAT RELEASE 5 CONTROL CARDS

GROUP	PAGE	COLUMN	LINE	NO.
-------	------	--------	------	-----

13	57	1	8	41
				42

14	62	1	8	43
----	----	---	---	----

44

45

46

47

48

49

50

51

TAB SET AT 15
GOCOLUMNS/PAGE = 2
WIDTH = 30
BETWEEN COLUMNS = 4
PAGE NUMBER IS NULL
NO CAPITALIZATION AUTOMATICALLY
NO JUSTIFICATION
STOP TITLE
TITLE ON LINE 5
GO

CODE	PAGE	COLUMN	LINE	CHAR/GROUP/CARD	ERROR
212	1	1	0	2	CONTROL CARD OPERAND IN ERROR
700	29	1	47	8	UNDEFINED COMMAND OPERAND

SPECIAL PRINTER TRAIN (REMOVE THIS CARD TO PRINT IN UPPER AND LOWER CASE)
COLUMNS PER PAGE = 9 (MAX IS 8; SEE FOLLOWING DIAGNOSTIC. WILL GET 1 COLUMN)
START TEXT ON LINE 14, PP 12

CAP OPTION

LINES/PAGE = 56

WIDTH OF COLUMNS IS 60 CHARACTERS

PAGE NUMBER IS NULL

NONTRIVIAL BLANK IS REPRESENTED BY SPECIAL CHARACTER 34 (~)

INDENT THE COLUMN (8,0), (5,5), (5,0), (3,0), (6,0), (4,0), (2,0)

GO

THE DATA DECK GENERATING THIS MANUAL WILL PRODUCE IT IN UPPER AND LOWER CASE IF
THE FIRST CARD, THE)# FORMAT)# CONTROL CARD)# "SPECIAL PRINT TRAIN",)# IS R
EMOVED.)S)MF THE)# FORMAT)# MANUAL)LLL GERALD M. BERNS)JJJJJJJJJJJJ)LL
LLLLL #RELEASE 5)LLLLLF #MODIFICATIONS AND #ADDITIONS BY)PLL JOHN R. EHRMAN)L
COMPUTATION GROUP)L STANFORD LINEAR ACCELERATOR CENTER)L STANFORD, CALIFORNIA
94305)LLLL #JULY 1971)SV

START TEXT ON LINE 8 IN POSITION 10

WIDTH OF COLUMNS = 64

REPEAT TITLE

TABS AT 8 AND 63

TITLE: LINE 1, PRINT POSITION 25

)# FORMAT ---)#F A TEXT PROCESSING PROGRAM)#E

GO

)#JHM CONTENTS)M#LLLLLL #I.)T #SUMMARY OF #FACILITIES)D ~1)LL #I#I.)T #INTR
DUCTION)D ~3)LL #I#I#I.)T #CONTROL #CARDS)D ~9)L)T ~#EXAMPLES OF #CORREC
T #CONTROL #CARDS)D 23)LT ~#EXAMPLES OF #FAULTY #CONTROL #CARDS)D 23)LL #I#
V.)T #COMMAND #OPERANDS AND #COMMAND #WORDS)D 24)LT ~#SUMMARY OF #COMMAND #O
PERANDS)D 28)LT ~#EXAMPLES OF #COMMAND #WORDS)D 29)LL #V.)T #SPECIAL #OPER
ANDS FOR #CAPITALIZATION AND #SPECIAL)LT #CHARACTERS, AND THE #NON-#TRIVIAL #BL
ANK)D 31)LT ~#EXAMPLES OF THE #USAGE OF #SPECIAL #OPERANDS)D 34)LL #V#I.)T
#THE #EDITOR #FACILITY)D 35)LT ~#EXAMPLES OF #EDITOR #CONTROL #CARD #GROUPS
)D 43)LL #V#I#I.)T #RULES FOR #USING)# FORMAT)#D 44)LL# VIII.)#T #SUMMARY
OF)# FORMAT)# #CONTROL #CARDS AND #COMMAND #OPERANDS)D 46)LL# IX.)#T #DATAS
ETS #USED BY)# FORMAT)#D 50)LL# X.)#T #DESCRIPTION OF)# FORMAT)# FOR #O#S/
360,)LT ~#AND #SUGGESTED #CONTROL #CARDS)D 53)LL# XI.)#T #HINTS AND #SUGGEST
IONS)D 55)LL# XII.)T# #ERROR #HANDLING AND #DIAGNOSTIC #MESSAGES)D 57)LL# X
III.)#T #APPENDIX)D 61)LLT #INDEX)LLT #CONTROL #CARDS)LLT #DIAGNOSTICS)SV
BACKSPACE CHARACTER IS NUMBER 46 (@)

CYCLE PAGE NUMBERS

RIGHT SIDE PAGE NUMBER TO START

PAGE NO. = 1

GO

#I.)U #SUMMARY OF #FACILITIES)ULLP)# FORMAT)# IS A PROGRAM FOR #SYSTEM/360 A
ND #SYSTEM/370 DESIGNED TO MEET THE NEED FOR A RAPID METHOD OF EDITING AND PRODU
CING PAPERS, REPORTS, AND OTHER FINISHED AND REPRODUCIBLE DOCUMENTS DIRECTLY ON
THE SYSTEM PRINTER, USING UPPER AND LOWER CASE AND SPECIAL CHARACTERS. IT HAS FA
CILITIES WHICH SIMPLIFY THE TASK OF INDEX CONSTRUCTION. INPUT TO THE PROGRAM IS
FREE-FORM CARD-IMAGE TEXT. THE DOCUMENT IS FORMATTED AND CONTROLLED ACCORDING TO
CONTROL CARDS AND #COMMAND #WORDS INTERSPERSED THROUGHOUT THE INPUT.)# FORMAT
)# IS A SINGLE PROGRAM REQUIRING NO AUXILIARY PROGRAMS FOR ITS OPERATION.)P VIA
ENTIRELY FREE-FORM CONTROL CARDS THE USER MAY SPECIFY:)LL ~~~~~#AUTOMATIC CA
PITALIZATION OF ALL SENTENCES)L ~~~~~#NUMBER OF TEXT COLUMNS PER PAGE)L ~~~~
~#WIDTH OF TEXT COLUMNS)L ~~~~~#NUMBER OF LINES PER PAGE)L ~~~~~#NUMBER
OF PRINT POSITIONS BETWEEN TEXT COLUMNS)L ---)HT #PAGE NUMBERING AND FIRST PAG
E NUMBER (OR NO NUMBERING))HLH ~~~~~#LOCATION OF PAGE NUMBER ON THE RIGHT, TH
E LEFT, OR ALTERNATING)HL ~~~~~)H #NUMBER OF PRINT POSITIONS FOR PARAGRAPH IN
DENTATION)HL ~~~~~)H #NUMBER OF PRINT POSITIONS FOR COLUMN INDENTATION)HL --
~#LINE SPACING (SINGLE SPACING, DOUBLE SPACING, ETC.))L ~~~~~#NUMBER OF L
INES BETWEEN PARAGRAPHS)L ~~~~~#RIGHT-JUSTIFICATION OF TEXT (OR NOT))L ~~~~~

[illegible]

[illegible]

AR TO THE ACTION CAUSED BY STRIKING THE "RETURN" KEY ON A TYPEWRITER: THE END OF THE CURRENT LINE IS SIGNALLED, AND THE CARRIAGE IS POSITIONED AT THE START OF THE FOLLOWING LINE. THUS, THE `#COMMAND #WORD "-!40#L#L~"` WOULD CAUSE THE LINE OF TEXT IN WHICH IT APPEARS TO TERMINATE, AND THE FOLLOWING LINE TO BE SKIPPED. THIS EXAMPLE OF A COMMAND WORD CONTAINS `)U TWO )U #COMMAND #OPERANDS, "#L" AND "#L"`; THIS SHOWS HOW `#COMMAND #OPERANDS` ARE GROUPED TO FORM `#COMMAND #WORDS`. `)P` WE WILL NOW LOOK AT A SIMPLE EXAMPLE OF `)# FORMAT )# INPUT`: SUPPOSE WE WISH TO PRINT THE FIRST PART OF THE TEXT MATERIAL SHOWN IN THE FIGURE. THE INPUT TEXT COULD BE PREPARED AS FOLLOWS: `)LL#W2#A`

`!40#P THIS IS THE BEGINNING OF A PARAGRAPH; THE SIZE OF THE INDENT AT THE START OF THE PARAGRAPH MAY BE SPECIFIED ON A CONTROL CARD. !40#L NOW, !40#L THIS MATERIAL BEGINS A NEW COLUMN-LINE: THAT IS, IT STARTS A NEW LINE WITHIN THE CURRENT COLUMN OF TEXT MATERIAL. !40#L#L#L`

`)#LP` SEVERAL IMPORTANT POINTS ARE ILLUSTRATED IN THIS EXAMPLE. FIRST, THE INPUT TO `)# FORMAT )#` IS ENTIRELY FREE-FORM: THE USER MAY LEAVE AS MANY SPACES BETWEEN INPUT WORDS AS HE LIKES, AND `)# FORMAT )#` WILL IGNORE THE EXCESS BLANKS AS IT COLLECTS WORDS TO BE PLACED IN THE PAGE IMAGE. SECOND, THERE IS NO NEED TO START A NEW `)U INPUT )U LINE` WHEN A NEW `)U OUTPUT )U LINE` IS DESIRED; THE `"#L" #COMMAND #OPERAND` WILL START A NEW LINE ON THE OUTPUT PAGE. `)#LL )P` AS THE INPUT CARDS ARE READ BY `)# FORMAT, )#` IT MAY BE NECESSARY TO CHANGE SOME OF THE CONTROL VARIABLES WHICH DETERMINE THE ARRANGEMENT OF THE TEXT ON THE PAGE. FOR EXAMPLE, THE USER MAY WANT TO CHANGE FROM ONE COLUMN PER PAGE TO TWO (AS WAS DONE TO PRODUCE THE INDEX FOR THIS MANUAL). TO GO FROM NORMAL TEXT MODE BACK TO CONTROL CARD MODE, A `#COMMAND #WORD` IS PLACED IN THE INPUT STREAM WHICH ENDS WITH THE `#COMMAND #OPERAND "#V"`. THE REST OF THE CARD FOLLOWING THE `"#V"` IS IGNORED, AND `)# FORMAT )#` BEGINS READING CONTROL CARDS WITH THE NEXT INPUT CARD. THUS, THE USER CAN DYNAMICALLY MODIFY THE LAYOUT OF THE TEXT ON THE PAGE, AND CAN CHANGE THE VALUES OF THE CONTROL VARIABLES. AS BEFORE, THE END OF THE CONTROL CARD GROUP IS SIGNALLED BY A `"#G#O"` CONTROL CARD. `)P` TO ILLUSTRATE, SUPPOSE WE WISH TO SET THE MARGIN INDENTS TO BE ZERO SPACES AT THE LEFT AND TEN SPACES AT THE RIGHT, AS IN THE SECOND PORTION OF THE FIGURE ABOVE. THE NECESSARY INPUT COULD BE PREPARED AS FOLLOWS:

`: )LL#W2A`

`!40#V`

INDENTATION OF THE COLUMN IS (0,10) POSITIONS

`GO`

`!40#I !43THIS TEXT MATERIAL ILLUSTRATES`

THE USE OF AN INDENT: THE RIGHT MARGIN HAS BEEN INDENTED AN ADDITIONAL 10 SPACES. `!40#I#L#L#L`

`)#LP` IN THIS EXAMPLE, THE `"#I" #COMMAND #OPERAND` WAS USED TO CONTROL INDENTATION OF THE MARGINS. ITS OPERATION IS LIKE THAT OF AN "ON-OFF" SWITCH: EACH APPEARANCE OF THE `"#I" #COMMAND #OPERAND` CAUSES INDENTATION TO BEGIN (IF IT WAS NOT ALREADY IN EFFECT) OR TO END (IF IT WAS IN EFFECT). IT IS NOT LIKE THE `"#L" #COMMAND #OPERAND`, WHICH CAUSES A NEW LINE EACH TIME IT APPEARS; `"#I" DOES NOT CAUSE ADDITIONAL INDENTATION EACH TIME IT APPEARS, BUT TURNS THE INDENTATION ON OR OFF. THE "#" SIGN PRECEDING THE WORD )# "THIS" )#` IS CALLED A `)PU SPECIAL OPERAND. )PU` IT CAUSES THE IMMEDIATELY FOLLOWING LETTER (THE `"#T"`) TO BE CAPITALIZED IN THE PRINTED OUTPUT. `#SPECIAL #OPERANDS` WILL BE DISCUSSED IN `#SECTION #V`. `)P` WE OBSERVE THAT THE NEXT SEGMENT OF TEXT MATERIAL IN THE FIGURE ALSO REQUIRES AN INDENTATION. THE CONTROL CARD WHICH ALLOWS US TO SET THE AMOUNT OF INDENTATION (THE `)# "INDENTATION OF THE COLUMN" )# CONTROL CARD`) CAN SPECIFY UP TO SEVEN DIFFERENT INDENTATIONS. THUS, RATHER THAN PREPARE ANOTHER CONTROL CARD, WE WILL GO BACK AND CHANGE THE PREVIOUS INPUT MATERIAL SO THAT IT WILL CONTROL BOTH OF THE INDENTED SEGMENTS OF TEXT IN THE FIGURE. THE USE OF THE `"#H" #COMMAND #OPERAND` WILL BE EXPLAINED SHORTLY. `)LL#W2A`

`!40#V`

INDENTATION OF THE COLUMN IS (0,10), (7,8) POSITIONS

GO

!40ZI !43THIS TEXT MATERIAL ILLUSTRATES
THE USE OF AN INDENT: THE RIGHT MARGIN HAS BEEN INDENTED AN
ADDITIONAL 10 SPACES. !40ZILLLH2 THIS TEXT MATERIAL
ILLUSTRATES THE USE OF A HANGING, OR DELAYED, INDENT: THE
TEXT IS NOT INDENTED UNTIL THE LINE FOLLOWING THE FIRST
LINE OF TEXT. !40ZH2LL

)LP AS NOTED ABOVE FOR "ZI", THE "ZH" COMMAND OPERAND WORKS LIKE AN ON-OFF SWITCH. AN ADDITIONAL FEATURE ILLUSTRATED IN THE ABOVE EXAMPLE IS THE "2" FOLLOWING THE "ZH" COMMAND OPERAND, WHICH MEANS THAT THE)U SECOND)U PAIR OF COLUMN INDENTATIONS IS TO BE USED IN DETERMINING THE NUMBER OF POSITIONS TO INDENT. SEVERAL OTHER COMMAND OPERANDS MAY BE FOLLOWED BY A NUMBER; THEY ARE DISCUSSED IN SECTION IIV.)P THE LAST TEXT SEGMENT IN THE FIGURE COULD BE PREPARED AS FOLLOWS:)LLW2A

!40M THIS TEXT !40L MATERIAL !40L IS !40L CENTERED! !40M

)LP THE "M" COMMAND OPERAND CAUSES CENTERING OF THE PRINTED TEXT TO BEGIN OR END. IT IS LIKE THE "ZI" AND "ZH" COMMAND OPERANDS IN BEING LIKE AN ON-OFF SWITCH, BUT "M" DOES NOT DEPEND ON A CONTROL CARD TO DETERMINE THE AMOUNTS OF SPACING TO BE PERFORMED.)P FORMAT)D DETERMINES THAT THE END OF THE INPUT HAS BEEN REACHED WHEN IT DETECTS A COMMAND WORD ENDING WITH THE COMMAND OPERAND "E". (THIS MEANS, OF COURSE, THAT THE END OF THE INPUT SHOULD OCCUR IN NORMAL TEXT MODE, NOT IN "AS-IS" TEXT MODE OR IN CONTROL CARD MODE.))P FORMAT)D THEN PRINTS THE FINAL TEXT PAGE, FOLLOWED BY A LIST OF ALL CONTROL CARDS READ, AND THE DIAGNOSTICS (IF ANY). AT THIS POINT,)P FORMAT)D WILL RE-INITIALIZE ITSELF TO READ A FRESH JOB, STARTING TO READ IN CONTROL CARD MODE JUST AS IT DID AT THE VERY BEGINNING. THUS, MULTIPLE DOCUMENTS MAY BE PRODUCED IN A SINGLE COMPUTER RUN.)P TO ILLUSTRATE A COMPLETE)P FORMAT)D JOB, WE WILL NOW GIVE A COMPLETE SET OF INPUT "CARD IMAGES" THAT COULD BE USED TO PRODUCE THE FIGURE AT THE START OF THIS SECTION. THE READER IS NOT EXPECTED TO UNDERSTAND ALL OF THE)P FORMAT)D TECHNIQUES USED, ALTHOUGH MOST OF THEM WILL BE FAMILIAR.)LLW3A

CAPITALIZE AUTOMATICALLY

WIDTH OF COLUMN IS 52 PRINT POSITIONS

LINES PER PAGE ARE 32

TEXT STARTS ON LINE 5, IN PRINT POSITION 1

COLUMNS PER PAGE = 1

LEFT TOP POSITION FOR PAGE NUMBER

PAGE NUMBER STARTING AT 12

TITLE STARTS ON LINE 1, IN PRINT POSITION 24

!40F TITLE !40FE

FOOTER STARTS ON LINE 32, IN PRINT POSITION 23

!40F FOOTER !40FE

GO

!40P THIS IS THE BEGINNING OF A PARAGRAPH; THE
SIZE OF THE INDENT AT THE START OF THE
PARAGRAPH MAY BE SPECIFIED ON A CONTROL CARD. !40L NOW, !40L
THIS

MATERIAL BEGINS A NEW COLUMN-LINE: THAT IS, IT STARTS
A NEW LINE WITHIN THE CURRENT COLUMN OF TEXT MATERIAL. !40LLLL
INDENTATION OF THE COLUMN IS (0,10), (7,8) POSITIONS

GO

!40ZI !43THIS TEXT MATERIAL ILLUSTRATES

THE USE OF AN INDENT: THE RIGHT MARGIN HAS BEEN INDENTED AN
ADDITIONAL 10 SPACES. !40ZILLLH2 THIS TEXT MATERIAL
ILLUSTRATES THE USE OF A HANGING, OR DELAYED, INDENT: THE
TEXT IS NOT INDENTED UNTIL THE LINE FOLLOWING THE FIRST
LINE OF TEXT. !40ZH2LL

!40M THIS TEXT !40L MATERIAL !40L IS !40L CENTERED! !40MH

) ALL) P THIS EXAMPLE SHOWS THE THREE LEVELS OF CONTROL PROVIDED BY) FORMAT.) CONTROL CARDS PROVIDE GLOBAL CONTROLS;) COMMAND) WORDS PROVIDE CONTROLS AT THE WORD LEVEL; AND) SPECIAL) OPERANDS PROVIDE CONTROLS AT THE CHARACTER LEVEL.) P)) FORMAT) PROVIDES A NUMBER OF OTHER POWERFUL CAPABILITIES SUCH AS THE)) DICTIONARY,) LOCATE,) AND) EDITOR) FACILITIES. THE BEGINNER SHOULD EXPERIMENT WITH SIMPLE TEXT INPUT UNTIL SOME FAMILIARITY WITH) FORMAT) HAS BEEN ATTAINED. AS A START, STUDY THE INPUT WHICH PRODUCED THE EXAMPLES ABOVE. THEN PUNCH THE ABOVE) FORMAT) JOB ON CARDS, ADD THE NECESSARY) JOB) CONTROL CARDS (SEE) SECTION) X; AN EXPERIENCED PROGRAMMER CAN HELP PREPARE THEM), AND RUN THE JOB ON THE COMPUTER. THEN GENERATE SOME SIMPLE INPUT TEXT, AND EXPERIMENT WITH OTHER) FORMAT) FEATURES.) P A SUGGESTED SEQUENCE FOR READING THIS MANUAL IS TO SKIM) SECTIONS) I) I) I THROUGH) V, AND THE HINTS AND SUGGESTIONS GIVEN IN) SECTION) X) I. THEN, AFTER STUDYING THE ABOVE INPUT AND RUNNING A FEW SIMPLE PROBLEMS, GO BACK AND STUDY THOSE SECTIONS MORE CAREFULLY. AS MORE EXPERIENCE IS GAINED, THE OTHER PARTS OF THE MANUAL MAY BE CONSULTED AS NEEDED.) S) I) I.) U) CONTROL) CARD) S) ULLP) EVERY) FORMAT) JOB MUST BEGIN WITH A CONTROL CARD GROUP, WHICH IS DEFINED AS A GROUP OF CONTROL CARDS ENDING WITH THE) "GO") CONTROL CARD (ALL OTHER CONTROL CARDS ARE OPTIONAL). THE POSITION OF A CONTROL CARD WITHIN A CONTROL CARD GROUP IS NOT SIGNIFICANT, UNLESS SPECIFIED IN ITS DESCRIPTION. A CONTROL CARD GROUP MAY APPEAR AT ANY POINT IN THE INPUT TEXT STREAM (SEE THE) "V") COMMAND) OPERAND IN) SECTION) I) V). DEFAULT VALUES FOR EACH OPTION ARE ASSUMED IF NO CONTROL CARD PERTAINING TO THAT PARTICULAR OPTION HAS BEEN SUPPLIED IN ANY CONTROL CARD GROUP IN THE JOB. THE DEFAULT VALUES ARE SUMMARIZED IN) SECTION) V) I) I. THE VALUES OF MOST OF THE OPTIONS CAN BE VARIED AS NEEDED DURING THE READING OF THE INPUT AND THE FORMATTING OF THE DOCUMENT. HOWEVER, SOME OF THE OPTIONS CANNOT BE RESET ONCE THEY HAVE BEEN SET, AS NOTED IN THEIR DESCRIPTIONS.) P THE FORMAT OF EACH CONTROL CARD IS ENTIRELY FREE-FORM, AS LONG AS THE FIRST 3 NON-BLANK CHARACTERS OF EACH CONTROL CARD ARE AS SPECIFIED BY THE SUGGESTED CONTROL CARD NAME, AND THE CONTROL CARD IS ON ONE CARD IMAGE.) (FORMAT) SCANS CONTROL CARDS BY SAVING THE FIRST THREE NON-BLANK CHARACTERS, AND THEN SEARCHING FOR THE NUMBERS THAT GIVE THE VALUES OF THE PARAMETERS.) AT THE CONCLUSION OF EACH JOB, THE CONTROL CARDS USED FOR THAT JOB ARE LISTED BY GROUP ON THE) SYSTEM) OUTPUT DATA SET.) P WE WILL NOW GIVE THE SPECIFICATIONS FOR EACH OF THE CONTROL CARDS IN TURN. IN SOME OF THE DESCRIPTIONS, IT IS STATED THAT SOME OPTION MAY OR MAY NOT BE USED IN A TITLE; IN ALL SUCH CASES, THE STATEMENT APPLIES TO FOOTERS ALSO. EXAMPLES OF CORRECT AND FAULTY CONTROL CARDS WILL BE GIVEN AT THE END OF THIS SECTION. IN SOME OF THE CONTROL CARDS, NUMERIC OPERANDS MAY BE REQUIRED. THESE ARE REPRESENTED BY LOWER-CASE LETTERS SUCH AS) U X,) U Y,) U Z,) U OR) U NN;) U AN OPERAND SUCH AS) U X) U IS) U NOT) U LIMITED TO A SINGLE DIGIT.) L L L L L L W 4 BACKSPACE CHARACTER IS SPECIAL CHARACTER NUMBER) U NN) U) P TO SIMULATE THE ACTION OF THE BACKSPACE KEY ON A TYPEWRITER, ONE OF THE SPECIAL CHARACTERS (DESCRIBED IN) SECTION) V) MAY BE DESIGNATED AS THE "BACKSPACE" CHARACTER, EXCEPT FOR SPECIAL CHARACTERS NUMBERED 43 () AND 51 (!). THE ACTION OF THE BACKSPACE CHARACTER IS AS FOLLOWS: THE CHARACTER TO BE PRINTED OVER AND THE OVERPRINT CHARACTER ARE SEPARATED BY THE BACKSPACE CHARACTER, WITH A FEW MINOR EXCEPTIONS. IF THE BACKSPACE CHARACTER IS FOLLOWED BY A BLANK, THEN IT IS ASSUMED THAT NO OVERPRINT WAS DESIRED, AND THE BACKSPACE CHARACTER WILL PRINT NORMALLY. MULTIPLE BACKSPACES ARE IGNORED, AND HAVE NO MORE EFFECT THAN A SINGLE ONE; THEY ALL CAUSE ONLY A SINGLE BACKSPACE, AND THE ONLY CHARACTER WHICH WILL OVERPRINT THE CHARACTER PRECEDING THE FIRST BACKSPACE WILL BE THE CHARACTER FOLLOWING THE LAST BACKSPACE.) P THE NUMBER) U NN) U GIVEN ON THE CONTROL CARD MUST LIE BETWEEN 10 AND 50; IF IT DOES NOT, BACKSPACING WILL BE TURNED OFF AND NO CHARACTER WILL BE RECOGNIZED AS A BACKSPACE. NOTE THAT THE BACKSPACE CHARACTER, WHEN USED IN THE INPUT TEXT, MAY BE IN ITS ACTUAL (CHARACTER) FORM OR IN ITS SPECIAL (!NN) FORM. THE DEFAULT ACTION IS THAT NO BACKSPACES ARE RECOGNIZED.) P TO GIVE SOME EXAMPLES: SUPPOSE THE BACKSPACE CHARACTER IS NUMBER 50, THE QUESTION MARK (?). THEN THE INPUT CHARACTERS) 0 ? - WOULD PRODUCE) 0 @ - , / ? 0 WOULD PRODUCE / @ 0, AND LETT ? _ ER WOULD PRODUCE LETT @ ER. NOTE THAT SPECIAL CHARACTERS MAY BE USED FOR OVERPRINTING, SO THAT) 5 1 5 0 ! 5 1 3 3 WOULD PRODUCE) @ ! 3 3. THE FIGURE BELOW MAKES USE OF BACKSPACING TO PRINT THE DIVIDERS AT THE INSIDE EDGES OF THE BOXES: THE CHARACTERS ! 2 7 ? ! 2 5 PRODUCE ! 2 7 @ ! 2 5, A

029)# (EBCDIC))# MODE, REGARDLESS OF THEIR MODE IN THE ORIGINAL CARD INPUT DATASET.)P AT THE CONCLUSION OF THE RUN THE INPUT DATASET ON DATASET REFERENCE NUMBER 2 (THE CONDENSED INPUT) IS LISTED, WITH CARD IMAGE NUMBERS AND NUMBERED TEXT AND TITLE WORDS, ONTO THE #SYSTEM #OUTPUT DATASET. IF THE LISTING IS PRINTED IN UPPER CASE ONLY (DUE EITHER TO ERRORS OR TO THE PRESENCE OF THE)# "SPECIAL PRINT TRAIN")# CONTROL CARD), THEN AN ASTERISK WILL REPLACE EACH CHARACTER FOR WHICH NO GRAPHIC IS LIKELY TO BE ASSOCIATED. THE #COMMAND #OPERANDS CONTAINED IN EACH CARD IMAGE ARE LISTED AGAIN ALONGSIDE EACH CARD IMAGE.)P THE PRIMARY USE OF THIS CONTROL CARD IS TO PRODUCE A CARD IMAGE INPUT DATASET THAT CAN BE SAVED FOR LATER EDITING; SEE #SECTION #V#I FOR A DESCRIPTION OF THE #EDITOR FACILITY.)L LLL#W4 CYCLE THE PAGE NUMBER)#P #IF PAGE NUMBERING HAS BEEN REQUESTED (BY THE)# LEFT TOP POSITION")# OR)# "RIGHT TOP POSITION")# CONTROL CARD), THEN THE PAGE NUMBER WILL BE ALTERNATED BETWEEN THE LEFT AND RIGHT TOP CORNERS ON SUCCESSIVE DOCUMENT PAGES. THE PAGE NUMBER APPEARS ON LINE 1 ALIGNED WITH THE APPROPRIATE BORDER OF THE TEXT. THE DEFAULT ACTION IS THAT THE FIRST PAGE NUMBER IS ALIGNED WITH THE RIGHT TEXT BORDER. (SEE THE)# "LEFT TOP POSITION", "RIGHT TOP POSITION",)# AND)# "PAGE NUMBER")# CONTROL CARDS.) ONCE CYCLING OF THE PAGE NUMBER HAS BEEN REQUESTED, IT STAYS IN EFFECT FOR THE REMAINDER OF THAT JOB.)LLL#W4 DARKEN PRINT EACH PAGE)#U X)#U TIMES)#P NORMALLY, EACH LINE ON THE OUTPUT PAGE WILL BE PRINTED ONCE. IF X HAS A VALUE OF 2 OR 3, EACH LINE WILL BE PRINTED SUCCESSIVELY ON TOP OF ITSELF UNTIL IT HAS BEEN PRINTED A TOTAL OF X TIMES. THIS ALLOWS DARKER PRINTING OF THE PAGE, AND IF THE PRINTER IS WELL-ADJUSTED AND THE PRINTER RIBBON IS NEITHER TOO NEW NOR TOO OLD, THE TEXT IS PRINTED WITHOUT THE NORMAL BLUR AND GRAIN FROM THE RIBBON. IF X IS 0, IT IS SET TO 1, AND IF IT IS GREATER THAN 3, IT IS SET TO 3. THE NUMBER OF TIMES EACH LINE IS PRINTED IS DETERMINED BY THE VALUE OF X IN EFFECT AT THE TIME THE ENTIRE PAGE IS PRINTED, SO IT IS)U NOT)U POSSIBLE TO PRINT PORTIONS OF A PAGE IN "BOLDFACE". THE DEFAULT IS SINGLE PRINTING.)LLL#W4 DICTIONARY OF WORDS USED)P AN ALPHABETIZED LIST, 6 COLUMNS PER PAGE, OF ALL SIGNIFICANT WORDS IN THE INPUT STREAM, WITH A COUNT OF THE OCCURRENCES OF EACH, IS WRITTEN ONTO THE #SYSTEM #OUTPUT DATASET AT THE CONCLUSION OF THE LAST)# FORMAT)# JOB. THIS DICTIONARY, IN UPPER CASE, IS FORMED ACCORDING TO THE FOLLOWING RULES:)LLI4W1H5 !30--NO WORD OF FEWER THAN 3 LETTERS IS LISTED)HLH5 !30--ALL NON-LETTERS ARE TREATED AS WORD DELIMITERS, EXCEPT FOR "E" WHICH IS IGNORED)HLH5 !30--#COMMAND #WORDS ARE IGNORED)HLH5 !30--CASE OF THE PRINTED TEXT IS IGNORED, BUT THE INPUT)U MUST BE IN UPPER CASE)U (SEE THE)# "SPECIAL KEYPUNCH")# CONTROL CARD))HLH5 !30--TEXT, TITLES, AND CONTROL CARDS ALIKE ARE SCANNED)HLH5 !30--WORDS LONGER THAN 40 LETTERS ARE BROKEN UP INTO 40 LETTER SEGMENTS)HLH5 !30--94 COMMON WORDS (SUCH AS "THOUGH", "ALSO", AND "WHERE") ARE SUPPRESSED)IHP THE DICTIONARY IS USEFUL FOR DETERMINING A ROUGH LIST OF CANDIDATES FOR AN INDEX, AND FOR A SPELLING CHECK. THE)# "\$LOCATE")# #EDITOR CONTROL CARD CAN BE USED (IN THE EDIT PHASE) TO FIND THE LOCATION OF "INDEX CANDIDATES" IN CONTEXT.)P THE)# DICTIONARY)# FACILITY USES DATASET REFERENCE NUMBERS 2 AND 3 (SEE #SECTION #I#X FOR DETAILS).)LLL#W4 DROP CHARACTER FOR 'D' COMMAND IS)#U X)U)P WHEN A TAB COMMAND IS USED TO SKIP OVER BLANK POSITIONS IN A COLUMN LINE, THE SPACES CAN OPTIONALLY BE FILLED WITH A CHARACTER SUCH AS A DOT. THIS CHARACTER IS CALLED THE "DROP" CHARACTER, SINCE IT MAY BE THOUGHT OF AS BEING "DROPPED BEHIND" AS THE LINE POSITION MOVES TO THE RIGHT. NORMALLY, THE CHARACTER DROPPED BY THE "#D" #COMMAND #OPERAND (SEE #SECTION #I#V) WILL BE A PERIOD. THIS CONTROL CARD MAY BE USED TO CHANGE THAT CHARACTER, AS FOLLOWS: IF X IS A NUMBER BETWEEN 10 AND 51, THEN THE DROP CHARACTER WILL BE THE CORRESPONDING SPECIAL CHARACTER; IF X LIES BETWEEN 64 AND 255, THE DROP CHARACTER WILL BE THE)# EBCDIC)# CHARACTER WHOSE REPRESENTATION HAS THAT VALUE; IF IT IS ZERO OR OMITTED, THEN THE DROP CHARACTER WILL BE RESET TO A PERIOD. THE DEFAULT CHARACTER IS A PERIOD. AS AN EXAMPLE, THE CONTROL CARD)# "DROP 30")# WOULD DROP "BULLETS" (!30) WHEN THE "#D" COMMAND OPERAND IS USED.)LLL#W4 EDITOR)P #THIS CONTROL CARD INVOKES THE)# #FORMAT)# #EDITOR, WHICH IS DESCRIBED IN #SECTION #V#I. IF USED, THIS CONTROL CARD MUST BE THE FIRST OF THE JOB AND MUST BE PART OF THE #SYSTEM #INPUT DATASET.)LLL#W4 FOOTER ON LINE)#U X)#U PRINT POSITION)#U Y)#U PRECEDED BY)#U Z)#U BLANK LINES)#P #THE FOOTER IS PLACED INTO THE PRINT PAGE BEGINNING ON LINE)#U X)U AT PRINT POSITION)#U Y,)U AND IS SEPARATED FROM THE LAST LINE OF TEXT

BY AT LEAST)U Z)U BLANK LINES. THIS CONTROL CARD, IF USED, MUST BE FOLLOWED IMMEDIATELY BY THE FOOTER TEXT. THE TEXT OF THE FOOTER)U MUST)U BE ENDED BY THE "E" COMMAND OPERAND. AFTER THE FOOTER TEXT, THE ONLY ALLOWABLE CONTROL CARD IS THE)E "TITLE")E OR THE)E "GO")E CONTROL CARD. THE FOOTER APPEARS ON EVERY DOCUMENT PAGE UNTIL IT IS REPLACED (THROUGH THE USE OF ANOTHER)E "FOOTER")E CONTROL CARD.) THE DEFAULT VALUE FOR X IS THE LAST LINE OF THE DOCUMENT PAGE, THE DEFAULT VALUE FOR Y IS THE PRINT POSITION OF THE LEFT TEXT BORDER OF THE DOCUMENT, AND THE DEFAULT VALUE FOR Z IS 2.)LLLW4 G0)P THIS IS THE ONLY CONTROL CARD REQUIRED BY)E FORMAT. GO)E SIGNALS THE END OF A CONTROL CARD GROUP, AND INITIATES PROCESSING IN NORMAL TEXT MODE.)LLLW4 INDENTATION OF THE COLUMN IS)U (X1,Y1),..., (X7,Y7))U POSITIONS)P THIS CONTROL CARD, WHEN USED WITH THE "H" AND "I" COMMAND OPERANDS, ENABLES THE USER TO REDUCE THE WIDTH OF TEXT COLUMNS BY X POSITIONS ON THE LEFT AND Y POSITIONS ON THE RIGHT. SEVEN PAIRS OF COLUMN INDENTATIONS MAY BE SPECIFIED. THE DEFAULT ACTION IS THAT ALL X'S AND Y'S ARE ZERO.)LLLW4 JUSTIFICATION)P TEXT IN THE DOCUMENT BODY IS RIGHT-JUSTIFIED WITHIN COLUMN-LINES, EXCEPT WHEN A COLUMN-LINE IS TERMINATED BY A COMMAND WORD, OR WHEN THE LINE CONTAINS TABS. AFTER READING THE INPUT AND ELIMINATING ALL EXTRA BLANKS,)E FORMAT)E THEN PERFORMS RIGHT-JUSTIFICATION BY INTRODUCING THE NECESSARY NUMBER OF EXTRA BLANKS, ONE TO EACH WORD DELIMITER, WORKING ALTERNATELY FROM THE RIGHT END OF THE LINE LEFTWARD AND THE LEFT END OF THE LINE RIGHTWARD ON SUCCESSIVE LINES. THE NUMBER OF BLANKS BETWEEN INPUT TEXT WORDS IS IGNORED.)E FORMAT)E DOES NO HYPHENATION, WHICH MEANS THAT COLUMN-LINES CONTAINING LONG WORDS MAY HAVE LARGE GAPS BETWEEN WORDS. THE DEFAULT ACTION IS RIGHT-JUSTIFICATION. (SEE THE)E "NO JUSTIFICATION")E CONTROL CARD.))LLLW4 LEFT TOP POSITION FOR PAGE NUMBER)P THE PAGE NUMBER (IF ANY) IS PLACED ON LINE 1 ALIGNED WITH THE LEFT TEXT BORDER. THE DEFAULT ACTION IS THAT IT IS ALIGNED WITH THE RIGHT TEXT BORDER. (SEE THE)E "CYCLE PAGE NUMBER", "PAGE NUMBER",)E AND)E "RIGHT TOP POSITION")E CONTROL CARDS.))LLLW4 LINES PER PAGE ARE)U X)U THE NUMBER OF LINES OF ALL KINDS (INCLUDING TEXT LINES, PARAGRAPH SEPARATION LINES, TITLE LINES, AND BLANK LINES) WHICH ARE ALLOWED ON A DOCUMENT PAGE IS X. THE OPERAND MAY BE ANY NUMBER IN THE RANGE 5 THROUGH 1000. THE DEFAULT NUMBER IS 59.)LLLW4 LIST THE INPUT DATASET)P THIS CONTROL CARD IS IDENTICAL IN EFFECT TO THE)E "CREATE A TAPE")E CONTROL CARD.)LLLW4 NO CAPITALIZATION AUTOMATICALLY)P NO CAPITALS ARE AUTOMATICALLY PRODUCED. THIS IS ALSO THE DEFAULT. (SEE THE)E "CAPITALIZE AUTOMATICALLY")E CONTROL CARD.))LLLW4V

NO JUSTIFICATION

GO

)E NO JUSTIFICATION)P THE TEXT IS NOT RIGHT-JUSTIFIED (AS ILLUSTRATED IN THIS PARAGRAPH, WHICH WILL HAVE AN UNEVEN RIGHT MARGIN.) THE NUMBER OF BLANKS BETWEEN INPUT TEXT WORDS IS IGNORED. THE DEFAULT ACTION IS RIGHT-JUSTIFICATION OF TEXT .)V

JUSTIFICATION

BACKSPACE CHARACTER RESET TO 0 (TURN BACKSPACING OFF)

GO

)LLLW4 NONTRIVIAL BLANK IS REPRESENTED BY SPECIAL CHARACTER)U NN)U)P TO FACILITATE THE USE OF THE NON-TRIVIAL (OR NON-ELIMINATABLE) BLANK FROM DEVICES (SUCH AS IBM 2741 TERMINALS) WHICH DO NOT ALLOW IT TO BE ENTERED IN THE SOURCE STREAM, THE USER MAY MAKE THE APPEARANCE OF ONE OF THE SPECIAL CHARACTERS BE EQUIVALENT TO THE PRESENCE OF A NON-TRIVIAL BLANK. THE NUMBER)U NN)U MUST BE BETWEEN 10 AND 51; OTHERWISE NO CHARACTER WILL BE REPLACED BY THE NON-TRIVIAL BLANK WHEN IT IS ENCOUNTERED. NOTE THAT THE)U ACTUAL)U SPECIAL CHARACTER MUST BE PRESENT TO BE REPLACED, AND NOT THE "SPECIAL CHARACTER REPRESENTATION" !NN, WHICH WILL BE TREATED NORMALLY. FOR EXAMPLE, IF THE NONTRIVIAL BLANK IS REPRESENTED BY SPECIAL CHARACTER NUMBER 46 (@), THEN THE TEXT "HERE@@@@@THERE" WOULD BE PRINTED AS "HERE-----THERE", AND THE NON-TRIVIAL BLANKS ARE NOT ELIMINATED AS ORDINARY BLANKS WOULD BE. THIS EQUIVALENCE ALSO TAKES EFFECT IN TITLES AND FOOTERS. THE DEFAULT IS THAT NO SUCH EQUIVALENCE IS MADE.)LLLW4 NULL CHARACTER SWITCH SET TO)U X)U)P NON-TRIVIAL BLANKS (OR)U NULL)U CHARACTERS) ARE NORMALLY IGNORED FOR CENTERING OR UNDERLINING PURPOSES WHEN THEY ARE AT THE END OF A WORD. IF X HAS THE VALUE 2, THEY WILL NOT BE IGNORED WHEN CENTERING AND UNDERLINING (UNDER CON

TROL OF THE "M" AND "U" COMMAND OPERANDS, RESPECTIVELY). IF X HAS ANY OTHER VALUE, IT WILL BE SET TO 1, WHICH IMPLIES THAT NULL CHARACTERS WILL BE TREATED NORMALLY. THE DEFAULT SETTING IS 1. THE EFFECT OF THIS CARD DOES NOT APPLY IN TITLES OR FOOTERS.)LLLW4 OUTPUT MEDIUM IS TAPE)P THE OUTPUT FROM)E FORMAT)E IS WRITTEN ONTO DATASET REFERENCE NUMBER 8 FROM THE POINT AT WHICH THIS CONTROL CARD IS READ. AT THE CONCLUSION OF THE JOB(S) THE TAPE IS COPIED ONTO THE)SYSTEM)OUTPUT DATASET THE NUMBER OF TIMES SPECIFIED ON THE LAST READ)E "COPIES =)E X" CONTROL CARD; OR ONCE, IF MULTIPLE COPIES ARE NOT SPECIFIED. THE TAPE CAN THEN BE LISTED AT SOME OTHER TIME, USING THE)E "PRINT OUTPUT TAPE")E CONTROL CARD.)LLLW4 PAGE NUMBER STARTING AT)EU X)UP THE PAGE NUMBER STARTS AT X (IF NON-BLANK AND NON-ZERO) AND IS PLACED ON LINE 1 OF EACH DOCUMENT PAGE. IF X IS ZERO OR BLANK, PAGE NUMBERING IS SUPPRESSED. THE DEFAULT PAGE NUMBER IS 1. (SEE THE)E "CYCLE PAGE NUMBER", "LEFT TOP POSITION",)E AND)E "RIGHT TOP POSITION")E CONTROL CARDS.) IF PAGE NUMBERING IS REQUESTED (BY THE)E "RIGHT TOP POSITION")E OR)E "LEFT TOP POSITION")E CONTROL CARDS), THEN ENOUGH CHARACTER POSITIONS MUST BE RESERVED AT BOTH THE TOP LEFT AND TOP RIGHT CORNERS OF THE PAGE FOR THE DIGITS OF THE PAGE NUMBER, WHETHER OR NOT THE NUMBER WILL ACTUALLY APPEAR IN BOTH POSITIONS.)LLLW4 PARAGRAPH INDENT IS)EU X)UP THE NUMBER OF PRINT POSITIONS SKIPPED AT THE START OF A PARAGRAPH IS X. THE DEFAULT INDENTATION IS 5 PRINT POSITIONS.)LLLW4 PRINT OUTPUT TAPE)P THE PRESENCE OF THIS CONTROL CARD MEANS THAT THE USER HAS PLACED A)PROGRAM-GENERATED OUTPUT DATASET (USUALLY A TAPE) ONTO DATASET REFERENCE NUMBER 8, AND THAT HE WISHES TO LIST IT ONTO THE)SYSTEM)OUTPUT DATASET THE NUMBER OF TIMES SPECIFIED ON THE MOST RECENT)E "COPIES =)E X" CONTROL CARD; OR, IF NONE, ONCE. THIS ACTION IS IMMEDIATE, NO DOCUMENT IS FORMED FROM AN INPUT DATASET, AND NO CONTROL CARDS OR ERROR DIAGNOSTICS RELATING TO THE CURRENT INPUT ARE WRITTEN. IT IS SUGGESTED THAT THE TAPE BE FILE-PROTECTED.)LLLW4 PUNCH THE INPUT DATASET)P THE EFFECT OF THIS CONTROL CARD IS IDENTICAL TO THAT OF THE)E "CREATE A TAPE")E CONTROL CARD, AND IN ADDITION, THE NEWLY CREATED CONDENSED INPUT DECK IS PUNCHED (I.E., WRITTEN ONTO THE)SYSTEM)PUNCH DATASET) AS WELL AS LISTED AT THE CONCLUSION OF THE RUN.)LLLW4 REPEAT TITLE ON EVERY PAGE)EP THE TITLE (IF ANY) IS PRINTED ON EVERY PAGE OF THE DOCUMENT. THE DEFAULT ACTION LIMITS THE APPEARANCE OF THE TITLE TO THE NEXT PAGE PRODUCED. (SEE THE)E "STOP PRINTING TITLE")E CONTROL CARD.) NOTE THAT THE TITLE AND FOOTER PRINTED FOR A GIVEN PAGE OF TEXT ARE THOSE IN EFFECT WHEN THE END OF THE CURRENT PAGE IS REACHED. THIS MEANS THAT CHANGING THE TITLE OR FOOTER WHEN TEXT IS BEING ACCUMULATED IN MID-PAGE WILL PLACE THE NEW TITLE OR FOOTER ON THE CURRENT PAGE, REPLACING THE OLD ONE (POSSIBLY BEFORE IT WAS EXPECTED TO).)LLLW4 RIGHT TOP POSITION FOR PAGE NUMBER)EP THE PAGE NUMBER (IF ANY) IS PLACED ON LINE 1 ALIGNED WITH THE RIGHT TEXT BORDER. THE DEFAULT ACTION IS THE SAME AS THE ACTION OF THIS CONTROL CARD. (SEE THE)E "CYCLE PAGE NUMBER", "LEFT TOP POSITION",)E AND)E "PAGE NUMBER")E CONTROL CARDS.))LLLW4 SENTENCES SEPARATED BY AT LEAST)EU X)UP SPACES)EP TEXT SENTENCES ARE SEPARATED ON THE SAME COLUMN-LINE (WHEN NOT IN AN "AS IS" REGION; SEE)SECTION)IV) BY X BLANKS WITH)E "NO JUSTIFICATION")E IN EFFECT, AND BY A MINIMUM OF X BLANKS WITH)E "JUSTIFICATION")E IN EFFECT. THE VALUE OF X MAY BE 1 OR 2; IF IT IS NOT 2, IT WILL BE SET TO 1. THE DEFAULT VALUE FOR X IS 1. NOTE THAT)E FORMAT)E WILL INSERT A)U MINIMUM)U OF X BLANKS, SO THAT EXTRA BLANKS MIGHT APPEAR. IF AN EXACT NUMBER OF SPACES IS NEEDED, USE THE NON-TRIVIAL BLANK.)LLLW4 SEPARATION LINES BETWEEN PARAGRAPHS ARE)EU X)UP THE NUMBER OF BLANK PRINTER LINES BETWEEN PARAGRAPHS IS X. THE DEFAULT NUMBER IS 1.)LLLW4 SIDE BY SIDE COPIES)EP TWO COPIES OF THE DOCUMENT ARE PRODUCED SIMULTANEOUSLY, SIDE BY SIDE. THE DEFAULT ACTION IS NOT TO PRINT SIDE BY SIDE COPIES. THERE MUST BE ENOUGH SPACE TO FIT TWO COPIES OF THE PRINTED TEXT, AND AT LEAST ONE SEPARATING SPACE, INTO A 132-CHARACTER PRINT LINE.)LLLW4 SPACING OF TEXT LINES IS)EU X)UP THE SPACING FOR THE DOCUMENT IS X (E.G., X=-1 MEANS SINGLE SPACING, X=-2 MEANS DOUBLE SPACING, ETC.). THE DEFAULT ASSUMPTION IS SINGLE SPACING.)LLLW5 SPECIAL KEYPUNCH)L SPECIAL KEYPUNCH IS A 2741)EP THIS CONTROL CARD SPECIFIES THAT THE TEXT INPUT ORIGINATED ON AN UPPER AND LOWER CASE KEYPUNCH, OR ON ANY DEVICE PRODUCING THE SPECIFIC)E EBCDIC)E CODE FOR EACH CHARACTER DESIRED. CASE IS NOT ALTERED BY THE PROGRAM, AND (FOR THE FIRST FORM OF THIS CONTROL CARD) THE)SPECIAL)OPERAND "Z" DOES NOT PRODUCE SUPERSCRIPTS.

THE DEFAULT ACTION IS THE USUAL CASE AND SUPERSCRIPIT CONVERSION. THE)# "SPECIAL KEYPUNCH")# CONTROL CARD ALLOWS THE USER TO PREPARE UPPER AND LOWER CASE INPUT FROM A TERMINAL, BUT STILL BE ABLE TO OBTAIN SUPERSCRIPITS THROUGH THE USE OF THE "#N" #SPECIAL #OPERAND TECHNIQUE. IF THERE IS ANY OTHER NUMERIC QUANTITY ON THIS CONTROL CARD, THE NORMAL)# "SPECIAL KEYPUNCH")# WILL BE ASSUMED. NOTE THAT THE EFFECT OF THIS CARD CAN BE CHANGED FROM 2741 MODE TO NORMAL)# SPECIAL KEYPUNCH)# MODE AND BACK, BUT THERE IS NO WAY TO RETURN FROM EITHER TO THE NORMAL MODE, WHERE UPPER-CASE-ONLY INPUT IS ASSUMED.)LLL#W4 SPECIAL PRINTER TRAIN)#P THIS CONTROL CARD IMPLIES THAT THE ULTIMATE PRINTER OF THE DOCUMENT CANNOT PRINT LOWER CASE OR SUPERSCRIPIT CHARACTERS. THEREFORE, NO TRANSLATION TO LOWER CASE OR SUPERSCRIPITS IS MADE. THE DEFAULT ACTION IS THAT THE TRANSLATION IS MADE.)LLL#W4 STOP PRINTING TITLE ON EVERY PAGE)#P #THE TITLE (IF ANY) IS ONLY PRINTED ON THE NEXT PAGE PRODUCED. THE DEFAULT ACTION IS THE SAME AS THE ACTION OF THIS CONTROL CARD. (#SEE THE)# "REPEAT TITLE")# CONTROL CARD.))LLL#W4 TABS ARE SET AT)#U X1,...,X14)UP #THE OPERAND FIELDS OF THIS CONTROL CARD SPECIFY THE POSITIONS OF UP TO 14 TAB STOPS. THESE ARE GIVEN AS RELATIVE CHARACTER POSITIONS WITHIN THE TEXT COLUMN-LINE (E.G., A TAB SET AT 10 MEANS THAT THE TAB FIELD BEGINS IN CHARACTER POSITION 10 IN EACH COLUMN-LINE; THE FIRST WORD FOLLOWING A TAB FROM ANY POSITION IN THE COLUMN-LINE BEFORE POSITION 10 WILL BE PLACED IN THE LINE STARTING IN CHARACTER POSITION 10). TABS MUST BE SET IN ASCENDING ORDER, AND NO TAB MAY BE SET AT A POSITION GREATER THAN THE COLUMN WIDTH. THE ACTION OF THIS CONTROL CARD IS ANALOGOUS TO THE ACTION OF THE "TAB SET" KEY ON A TYPEWRITER. THE DEFAULT ACTION IS THAT NO TABS ARE SET.)LLL#W4 TAPE INPUT DATASET)P #THE PRESENCE OF THIS CONTROL CARD MEANS THAT THE USER HAS PLACED A TAPE INPUT DATASET ONTO DATASET REFERENCE NUMBER 2. IT IS SUGGESTED THAT THE TAPE BE FILE-PROTECTED. IF THIS CONTROL CARD IS USED IT MUST BE THE FIRST CONTROL CARD OF THE JOB AND MUST BE PART OF THE #SYSTEM #INPUT DATASET. THIS CONTROL CARD CALLS THE)# FORMAT)# #EDITOR; THEREFORE, #EDITOR CONTROL CARDS MAY FOLLOW IT (SEE #SECTION #V#I). IT IS IDENTICAL IN EFFECT TO THE)# "EDITOR")# CONTROL CARD, EXCEPT THAT NO LISTING OF THE EDITED TAPE IS REQUESTED.)LLL#W6 TEXT STARTS ON LINE)#U X)U# IN PRINT POSITION)#U Y)UL OR)L)# START TEXT ON LINE)#U X)U# IN PRINT POSITION)#U Y)UP #THE FIRST LINE OF THE TEXT IS PRINTER LINE X, AND THE FIRST PRINT POSITION IS Y. THE DEFAULT IS LINE 5 AND THE DOCUMENT IS CENTERED ON THE PRINTER. THESE CONTROL CARDS ARE ENTIRELY EQUIVALENT, AND THE TWO FORMS ARE PROVIDED AS A CONVENIENCE.)LLL#W4 TITLE STARTS ON LINE)#U X)U# IN PRINT POSITION)#U Y)UP #THE FIRST PRINTER LINE OF THE TITLE IS X, AND THE FIRST PRINT POSITION OF THE TITLE IS Y. THIS CONTROL CARD, IF USED, MUST BE FOLLOWED IMMEDIATELY BY THE CARD IMAGES CONTAINING THE TITLE AND THE "#G#O" OR)# "FOOTER")# CONTROL CARD. (#REMEMBER THAT THE TITLE TEXT MUST END WITH THE "#E" #COMMAND #OPERAND.) THE TITLE MUST BE POSITIONED ABOVE THE BODY OF THE DOCUMENT. CONFLICTS OF THE TITLE WITH THE PAGE NUMBER ARE RESOLVED IN FAVOR OF THE PAGE NUMBER, AT BOTH OF THE TOP CORNERS OF THE PAGE. THE DEFAULT TITLE LINE IS PRINTER LINE 2, AND THE DEFAULT PRINT POSITION IS THAT OF THE LEFT TEXT BORDER.)LLL#W4 UNDERLINE SWITCH SET TO)#U X)UP IF X IS NOT ZERO, THE UNDERLINING ALGORITHM OF)# FORMAT)# IS MODIFIED SO THAT THE LEADING AND TRAILING CHARACTERS OF AN UNDERLINED STRING WILL NOT BE UNDERLINED IF THEY ARE ANY OF THE FOLLOWING TEN PUNCTUATION OR SPECIAL CHARACTERS: PERIOD, COMMA, COLON, SEMICOLON, QUESTION MARK, EXCLAMATION POINT, QUOTATION MARK, A POSTROPHE, AND LEFT OR RIGHT PARENTHESIS. IF X IS ZERO OR BLANK, ALL CHARACTERS IN THE STRING ARE UNDERLINED. THE DEFAULT VALUE OF X IS ZERO. TO GIVE AN EXAMPLE, SUPPOSE THE INPUT TEXT REQUIRES THAT ((#X)) BE UNDERLINED. THEN)V BACKSPACE CHARACTER IS 50 (?)

GO

)LLM (?_ (?_#X?)?)?_ AND ((#X?)?))LLM WOULD BE PRODUCED BY SETTING THE UNDERLINE SWITCH TO ZERO OR NONZERO RESPECTIVELY.)V

BACKSPACE 0 (RESET)

GO

)LLL#W4 WIDTH OF COLUMNS IS)#U X)U# PRINT POSITIONS)#P #THE WIDTH IN PRINT POSITIONS OF EACH TEXT COLUMN IS X. THE DEFAULT WIDTH IS 64 PRINT POSITIONS. IF A SINGLE COLUMN PER PAGE IS SPECIFIED, AND THE WIDTH IS CHOSEN TO BE 132 CHARACTER S (THE MAXIMUM), THEN THE MAXIMUM NUMBER OF LINES IS 59.)# FORMAT)# ALLOTS 59*

132, OR 7788, CHARACTERS FOR EACH PAGE OF THE TEXT. THE MAXIMUM ALLOWABLE COLUMN WIDTH, W, IS COMPUTED AS FOLLOWS:)LLW3A

$$W = \frac{16 \text{ MIN } (7788 / L, 132) + 17 / C - S + 1 - B * (N - 1)}{N}$$

)
)LW5 WHERE:)LL L = LINES PER PAGE (FROM)L "LINES PER PAGE")L CONTROL CARD))
 L C = 2 IF)L "SIDE BY SIDE")L CONTROL CARD IS IN EFFECT;)L L = 1 OTHERWISE)L
 S = STARTING PRINT POSITION (FROM)L "TEXT STARTS ON")L CONTROL CARD))L B = S
 PACES BETWEEN COLUMNS (FROM)L "BETWEEN COLUMNS")L CONTROL CARD))L N = NUMBER
 OF TEXT COLUMNS (FROM)L "COLUMNS/PAGE")L CONTROL CARD))LLW4 026 KEYPUNCH)L
 P THIS CONTROL CARD SPECIFIES THAT THE)L COMMAND)L WORDS USED IN THE INPUT CARDS
 HAVE BEEN PUNCHED ON AN)L IBM)L 026)L KEYPUNCH, OR ANY KEYPUNCHING DEVICE WHICH
 PUNCHES 12-8-4 FOR THE CHARACTER USED TO BEGIN)L COMMAND)L WORDS !16NOMINALLY ")"
 !17. THE DEFAULT ACTION ASSUMES THE)L IBM)L 029 CONFIGURATION FOR THE ")", WHI
 CH IS 11-8-5.)LLW4 029 KEYPUNCH)L P THIS CONTROL CARD SPECIFIES THAT THE)L CO
 MMAND)L WORDS USED IN THE INPUT CARDS HAVE BEEN PUNCHED ON AN)L IBM)L 029)L KEYP
 UNCH, OR ANY KEYPUNCHING DEVICE WHICH PUNCHES 11-8-5 FOR THE CHARACTER USED TO B
 EGIN)L COMMAND)L WORDS !16NOMINALLY ")"!17. THE DEFAULT ACTION IS THE SAME AS THE
 ACTION PRODUCED BY THIS CONTROL CARD.)SU)L EXAMPLES OF)L CORRECT)L CONTROL)L CARDS
)ULLLA
 !161!17)L CARD)L COLUMNS.)L !1680!17

TAB SET 5 10 15 20

TABS ARE SET AT PRINT POSITIONS 5, 10, 15 AND 20

TABULATE TO 5 10 15 20

TAB5 10*

15

20

START THE TEXT ON LINE 10, PRINT POSITION 20

START THE TEXT ON LINE 10 PRINT POSITION 20

START DOCUMENT: LINE = 10, P. POSITION = 20

STA 10 20

TEXT STARTS ON LINE 10, PRINT POSITION 20

TEXT STARTS IN DEFAULT POSITION

S T A R T T E X T 0 10

START TEXT 0 10

TEXT STARTS ON LINE 5 IN PRINT POSITION 10

TITLE

TITLE STARTS ON LINE 5 ABOVE LEFT TEXT BORDER

BACKSPACE CHARACTER IS NUMBER 50 (THE QUESTION MARK)

BACK 50 (?)

DAR 2

NO J

MEANS DON'T JUSTIFY

)

)L L L L U)L EXAMPLES OF)L FAULTY)L CONTROL)L CARDS)L U L L)L L A

GO NOW

BEGIN TEXT

)L ()L AN UNRECOGNIZABLE CONTROL CARD IS TREATED AS A ")L G)L O " CARD)L)L

START TEXT: 7

)L ()L TEXT STARTS ON LINE 7 IN THE DEFAULT PRINT POSITION)L)L

TABS = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

)L ()L THE 15TH TAB STOP IS IGNORED)L)L

LINES/PAGE = 5 7

)L ()L THE NUMBER OF LINES PER PAGE WILL BE 5, NOT 57)L)L

COLUMNS/PAGE = TWO

)L ()L THE NUMBER OF COLUMNS/PAGE WILL BE THE DEFAULT VALUE)L)L

BACKSPACE CHARACTER IS NUMBER 43 (#)

)# (# CANNOT BE USED AS THE BACKSPACE CHARACTER)

) S #I#V.) U #COMMAND #OPERANDS AND #COMMAND #WORDS) ULLP #A #COMMAND #OPERAND IS AN IMPERATIVE ORDER TO PERFORM AN IMMEDIATE TEXT-CONTROL FUNCTION. #COMMAND #OPERANDS CAN BE INTERSPERSED AS DESIRED THROUGHOUT THE TEXT INPUT IN THE FORM OF #COMMAND #WORDS. A #COMMAND #WORD IS A STRING OF ONE OR MORE #COMMAND #OPERANDS IN THE ORDER OF DESIRED EXECUTION, PREFIXED BY THE CHARACTER ")", AND FOLLOWED BY A BLANK. (SEE THE)# "026 KEYPUNCH")# AND)# "029 KEYPUNCH")# CONTROL CARDS IN #SECTION)# III).)#P SOME #COMMAND #OPERANDS MAY BE USED IN TITLES (INCLUDING FOOTERS), AS NOTED IN THE FOLLOWING PARAGRAPHS. IN ADDITION, SOME OF THE #COMMAND #OPERANDS ("#D", "#H", "#I", "#T", AND "#W") MAY BE FOLLOWED BY A NUMERIC QUANTITY, AS IN ")#T4".)#P THE FUNCTIONS OF THE #COMMAND #OPERANDS ARE DESCRIBED BELOW. WHEREVER A SINGLE LETTER APPEARS IN QUOTATION MARKS (LIKE "#L"), ITS USE AS A #COMMAND #OPERAND (LIKE ")#L") IS IMPLIED; OTHERWISE, THAT LETTER MAY APPEAR FREELY IN THE TEXT MATERIAL. IT IS IMPORTANT TO REMEMBER THAT ALL #COMMAND #OPERANDS ARE RECOGNIZED BY)# FORMAT)# ONLY IF THEY ARE IN UPPER CASE.) LLLLLL #A -- #ENTER THE "AS IS" TEXT MODE. IN THIS MODE, EACH CARD IMAGE IS AN INTEGRAL UNIT AND IS PRINTED ON A SEPARATE COLUMN-LINE WITHOUT CHANGE TO THE SPACING OF TEXT, EXCEPT THAT THE #SPECIAL #OPERANDS, AND CERTAIN OF THE #COMMAND #OPERANDS, TAKE NO PRINT POSITIONS. BOTH OF THE #SPECIAL #OPERANDS, AND THE #COMMAND #OPERANDS "#I" AND "#F", ARE EFFECTIVE IN THIS MODE. THE CONTENT OF A CARD IMAGE BEYOND THE EFFECTIVE COLUMN WIDTH IS NOT PRINTED. THE "AS IS" TEXT MODE IS ENDED WHEN ")#B" (WHERE B = BLANK) OCCURS IN THE)#U FIRST TWO)#U POSITIONS OF THE CARD IMAGE;)# FORMAT)# THEN RESUMES READING IN NORMAL TEXT MODE. (THE #COMMAND #OPERAND "#A" IS NOT RECOGNIZED IN "AS-IS" TEXT MODE, SO ")#A" CANNOT BE USED TO TERMINATE "AS-IS" MODE.) THE #COMMAND #OPERAND "#A" IS NOT VALID IN TITLES.) LLL #C -- #BEGIN THE NEXT TEXT COLUMN. THE #COMMAND #OPERAND "#C" IS NOT VALID IN TITLES OR FOOTERS.) LLLW2 #DN -- #TAB, LIKE "#T", LEAVING A STRING OF DOTS (PERIODS) INSTEAD OF BLANKS. THE STRING OF DOTS IS PREFACED BY ONE BLANK. THE CHARACTER TO BE "DROPPED" MAY BE CHANGED FROM A DOT TO ANY OTHER CHARACTER WITH THE)# "DROP CHARACTER")# CONTROL CARD. FOR A DETAILED DESCRIPTION OF THE "#D" #COMMAND #OPERAND, SEE THE DESCRIPTION OF THE "#T" #COMMAND #OPERAND BELOW. THE #COMMAND #OPERAND "#D" IS NOT VALID IN TITLES OR FOOTERS.) LLLW2 #E -- #END THE JOB, OR END THE FOOTER OR TITLE; THE NEXT CARD WILL BE READ IN CONTROL CARD MODE. AS MANY)# FORMAT)# JOBS AS DESIRED MAY BE STACKED ONE BEHIND THE OTHER. JOBS ON CARDS MUST PRECEDE JOBS WHICH ARE TAPE RESIDENT.) LLLW2 #F -- #BEGIN CAPITALIZING EACH WORD, CONTINUING UNTIL ANOTHER "#F", "#P", "#S", OR "#V" #COMMAND #OPERAND OCCURS. THE #COMMAND #OPERAND "#F" IS VALID IN TITLES AND FOOTERS, AND IN "AS-IS" TEXT MODE.) LLL #HN -- #REDUCE THE COLUMN WIDTH STARTING WITH THE)#U NEXT)#U COLUMN-LINE. IF)#U NO)#U "HANGING" (DELAYED) INDENT IS CURRENTLY IN USE, THEN N REFERS TO THE NTH PAIR OF ARGUMENTS ON THE)# "INDENT COLUMN")# CONTROL CARD; IF N IS BLANK, ZERO, OR ONE, THE FIRST PAIR IS REFERENCED. IF A HANGING INDENT)#U IS)#U CURRENTLY IN USE, THE NTH PAIR OF COLUMN INDENTS REPLACES THE PAIR IN USE, UNLESS THE NTH PAIR IS THE ONE BEING USED, IN WHICH CASE THE HANGING INDENT IS TURNED OFF. IF ANY HANGING INDENT IS IN EFFECT, A BLANK OR ZERO VALUE FOR N TURNS IT OFF, AS DOES THE #COMMAND #OPERAND "#S". THE #COMMAND #OPERANDS "#H" AND "#I" MAY BE USED INDEPENDENTLY OF EACH OTHER. THE #COMMAND #OPERAND "#H" IS NOT VALID IN TITLES OR FOOTERS.) LLL #IN -- #REDUCE THE COLUMN WIDTH IMMEDIATELY, AND TERMINATE THE CURRENT COLUMN-LINE. IF)#U NO)#U IMMEDIATE INDENT IS CURRENTLY IN USE, THEN N REFERS TO THE NTH PAIR OF ARGUMENTS ON THE)# "INDENT COLUMN")# CONTROL CARD; IF N IS BLANK, ZERO, OR ONE, THE FIRST PAIR IS REFERENCED. IF AN IMMEDIATE INDENT)#U IS)#U CURRENTLY IN USE, THE NTH PAIR OF COLUMN INDENTS REPLACES THE PAIR IN USE, UNLESS THE NTH PAIR IS THE ONE BEING USED, IN WHICH CASE THE IMMEDIATE INDENT IS TURNED OFF. IF)#U ANY)#U IMMEDIATE INDENT IS IN EFFECT, A BLANK OR ZERO VALUE FOR N TURNS IT OFF, AS DOES THE #COMMAND #OPERAND "#S". THE #COMMAND #OPERANDS "#I" AND "#H" MAY BE USED INDEPENDENTLY OF EACH OTHER. THE #COMMAND #OPERAND "#I" IS NOT VALID IN TITLES OR FOOTERS.) LLL #J -- #START THE NEXT COLUMN-LINE. THE ACTION OF THIS #COMMAND #OPERAND IS IDENTICAL TO THE ACTION OF "#L", EXCEPT THAT IT IS NOT VALID IN TITLES AND IS EFFECTIVE AT)#U ANY)#U LINE IN THE TEXT.

UNLIKE "EL", IT IS NOT IGNORED AT THE TOP OF A COLUMN.) LLLW2 EK -- KEEP THE FOLLOWING TEXT, UNTIL THE NEXT OCCURRENCE OF "EK", IN THE CURRENT TEXT COLUMN, IF POSSIBLE. (A SEGMENT OF TEXT DELIMITED BY "EK" COMMAND OPERANDS IS CALLED "KEEP TEXT" OR A "KEEP".) OTHERWISE, START THIS BLOCK OF TEXT IN THE NEXT TEXT COLUMN. THE COMMAND OPERAND "EK" IS NOT VALID IN TITLES AND FOOTERS, AND TERMINATES THE COLUMN-LINE ON WHICH IT OCCURS. IF THIS COMMAND OPERAND IS USED, THE PROGRAM REQUIRES DATASET REFERENCE NUMBER 2. BACKSPACES DO NOT WORK PROPERLY INSIDE "KEEP" TEXT. SEE THE DESCRIPTION OF THE "EW" COMMAND OPERAND ALSO.) LLLW2 EL -- START THE NEXT COLUMN-LINE, IF NOT AT THE FIRST LINE OF A TEXT COLUMN. THE ACTION OF THIS COMMAND OPERAND IS SIMILAR TO THE ACTION PRODUCED BY STRIKING THE "RETURN" BUTTON ON AN ELECTRIC TYPEWRITER. "EL" IS VALID IN TITLES. IF "EL" IS USED IN A TITLE, THE NEXT PRINTER LINE IS BEGUN; IF "EL" IS USED IN THE BODY OF THE DOCUMENT THE NEXT COLUMN-LINE IS BEGUN, LEAVING) ("SPACING" - 1)) BLANK LINES BETWEEN. IF "EL" IS USED AT THE TOP OF A TEXT COLUMN, IT IS IGNORED.) LLL EN -- BEGIN CENTERING TEXT WITHIN THE COLUMN-LINE, AND CONTINUE DOING SO FOR SUCCESSIVE LINES UNTIL ANOTHER "EN", "EP", OR "ES" OCCURS. THE COMMAND OPERAND "EN" IS NOT VALID IN TITLES OR FOOTERS; IF CENTERING IS REQUIRED IN A TITLE OR FOOTER, THE APPROPRIATE NUMBER OF NON-TRIVIAL BLANKS MAY BE USED (SEE SECTION EV).) LLL EP -- BEGIN A NEW PARAGRAPH, LEAVING THE NUMBER OF PRINTER LINES SPECIFIED BY THE) "SEPARATION LINES") CONTROL CARD (OR ITS DEFAULT) BETWEEN PARAGRAPHS AND INDENTING THE NUMBER OF PRINT POSITIONS SPECIFIED BY THE) "PARAGRAPH INDENT") CONTROL CARD (OR ITS DEFAULT). A NEW COLUMN IS BEGUN IF AT LEAST TWO COLUMN-LINES OF THE PRESENT COLUMN ARE NOT AVAILABLE FOR THE NEW PARAGRAPH. "EP" STOPS THE ACTION INITIATED BY COMMAND OPERANDS) "P", "H", "U",) AND "E". IF) "CAPITALIZE AUTOMATICALLY") IS IN EFFECT, THE NEXT TEXT WORD IS CAPITALIZED. "EP" IS NOT VALID IN TITLES.) LLL ES -- BEGIN A NEW PAGE. "ES" STOPS THE ACTION INITIATED BY COMMAND OPERANDS) "P", "H", "I", "M", "U",) AND "E". IF) "CAPITALIZE AUTOMATICALLY") IS IN EFFECT, THE NEXT TEXT WORD IS CAPITALIZED. "ES" IS NOT VALID IN TITLES.) LLL ETN -- IF N IS BLANK OR ZERO, TAB TO THE NEXT SET TAB POSITION BEYOND THE PRESENT POSITION IN THE COLUMN-LINE. THE ACTION OF "ET" CORRESPONDS TO THE ACTION PRODUCED BY STRIKING THE TABULATE KEY ON A TYPEWRITER. RIGHT-JUSTIFICATION, IF IN EFFECT WHEN "ET" IS USED, WILL NOT BE PERFORMED FOR THE COLUMN-LINE ON WHICH THE TAB OCCURS. THE "ED" AND "ET" COMMAND OPERANDS MAY BE FOLLOWED BY A NUMBER WHICH SPECIFIES THE TAB STOP TO BE USED. THAT IS, "140ET4" WILL CAUSE A TABULATION TO THE FOURTH TAB POSITION ON THE CURRENT COLUMN-LINE. IF THE COMMAND OPERAND IS USED INCORRECTLY, IT WILL BE IGNORED, AND A DIAGNOSTIC MESSAGE WILL BE PRINTED. "ET" IS NOT VALID IN TITLES.) LLL EU -- BEGIN UNDERLINING, CONTINUING UNTIL ANOTHER "EU", "EP", OR "ES" OCCURS. AT MOST 9 COLUMN-LINES, OR PORTIONS, MAY BE UNDERLINED PER PAGE. UNDERLINES NEITHER BEGIN NOR END UNDER THE SPACES SKIPPED OVER BY TABBING (PRODUCED BY "ET" AND "ED"). UNDERLINES MAY OR MAY NOT BEGIN AND END WITH NON-TRIVIAL BLANKS, DEPENDING ON THE) "NULL CHARACTER SWITCH") SETTING IN EFFECT (SEE THE) "NULL CHARACTER SWITCH") CONTROL CARD DESCRIPTION IN SECTION I1I1I1). UNDERLINES MAY OR MAY NOT BEGIN UNDER PUNCTUATION CHARACTERS, DEPENDING ON THE) "UNDERLINE SWITCH") SETTING IN EFFECT (SEE THE DESCRIPTION OF THE) "UNDERLINE SWITCH") CONTROL CARD IN SECTION I1I1I1). INDIVIDUAL CHARACTERS WITHIN A WORD CANNOT BE UNDERLINED EXCEPT BY BACKSPACING (SEE THE) "BACKSPACE") CONTROL CARD DESCRIPTION IN SECTION I1I1I1). THE COMMAND OPERAND "EU" IS NOT VALID IN TITLES OR FOOTERS.) LLLW2 EV -- LEAVE NORMAL TEXT MODE, AND BEGIN TO READ IN THE NEXT GROUP OF CONTROL CARDS. A CONTROL CARD GROUP MUST IMMEDIATELY FOLLOW THE CARD IMAGE CONTAINING THE "EV". ALL CHARACTERS FOLLOWING "EV" ON THE SAME CARD IMAGE ARE IGNORED. "EV" STOPS THE ACTION INITIATED BY THE COMMAND OPERANDS "EP" AND "E". IF) "CAPITALIZE AUTOMATICALLY") IS IN EFFECT, THE NEXT TEXT WORD IS CAPITALIZED. THE COMMAND OPERAND "EV" IS NOT VALID IN TITLES OR FOOTERS.) LLLW2 EWN -- KEEP THE NEXT N COLUMN-LINES IN THE SAME TEXT COLUMN. IF N COLUMN-LINES DO NOT REMAIN IN THE CURRENT TEXT COLUMN, START THE NEXT TEXT COLUMN. "EW" TERMINATES THE COLUMN-LINE ON WHICH IT OCCURS. IT IS NOT VALID IN TITLES. NOTE THAT THE "EW" COMMAND OPERAND IS SIMILAR IN EFFECT TO "EK", BUT DOES NOT REQUIRE THE USE OF AN ADDITIONAL DATASET. "EW" CAN BE USED TO PREVENT "WIDOWS", WHICH ARE SMALL SEGMENTS OF TEXT LEFT ALONE AT THE BOTTOM OF A COLUMN.) LLLW2 E -- BEGIN PRINTING ALL LETTERS

RS IN UPPER CASE, CONTINUING UNTIL ANOTHER "Z", "ZP", "ZS", OR "ZV" OCCURS. THE ACTION OF "Z" IS EQUIVALENT TO LOCKING A TYPEWRITER KEYBOARD IN UPPER CASE, AND THEN UNLOCKING IT. "Z" IS VALID IN TITLES. "Z" DOES NOT CAUSE NUMBERS TO BE PRINTED AS SUPERSCRIPITS, IT IS NOT AFFECTED BY PUNCTUATION CHARACTERS, NOR DOES IT CAUSE LETTERS PUNCHED IN LOWER CASE (AS INDICATED BY A)Z "SPECIAL KEYPUNCH")Z CONTROL CARD) TO BE PRINTED AS CAPITALS. (ZSEE THE DESCRIPTION OF THE "Z" ZSPECIAL ZOPERAND IN ZSECTION ZV ALSO.))LLLLW3U ZSUMMARY OF ZCOMMAND ZOPERANDS)ULLL 1. ZCOMMAND ZOPERANDS ALLOWED IN TITLES AND FOOTERS ARE: "ZE", "ZP", "ZL", AND "Z".)LL 2. ZCOMMAND ZOPERANDS WHOSE EFFECT IS ENDED BY "ZP" ARE: "ZP", "ZH", "ZU", AND "Z".)LL 3. ZCOMMAND ZOPERANDS WHOSE EFFECT IS ENDED BY "ZS" ARE: "ZP", "ZH", "ZU", "ZL", "ZV", AND "Z".)LL 4. ZCOMMAND ZOPERANDS WHICH TERMINATE THE COLUMN-LINE IN WHICH THEY APPEAR ARE: "ZA", "ZC", "ZE", "ZL", "ZJ", "ZK", "ZL", "ZH", "ZP", "ZS", "ZV", AND "ZW".)LL 5. ZCOMMAND ZOPERANDS WHICH CAUSE CAPITALIZATION OF THE NEXT TEXT LETTER (IF)Z "CAPITALIZE AUTOMATICALLY")Z IS IN EFFECT) ARE: "ZP", "ZS", AND "ZV".)LL 6. ZCOMMAND ZOPERANDS VALID IN "AS-IS" REGIONS ARE: "ZP" AND "Z".)LL 7. ZCOMMAND ZOPERANDS WHICH TERMINATE THE EFFECT OF "ZU" ARE: "ZP", "ZS", AND "ZU".)LL 8. ZCOMMAND ZOPERANDS WHICH TERMINATE THE EFFECT OF "Z" ARE: "ZP", "ZS", "ZV", AND "Z".)LL 9. ZCOMMAND ZOPERANDS WHICH TERMINATE CENTERING (INITIATED BY "ZH") ARE: "ZH", "ZP", AND "ZS".)LL 10. ZCOMMAND ZOPERANDS WHICH TERMINATE THE EFFECT OF "ZP" ARE: "ZP", "ZP", "ZS", AND "ZV".)LL 11. ZCOMMAND ZOPERANDS WHICH MAY BE FOLLOWED BY A NUMERIC QUANTITY ARE: "ZDN", "ZHN", "ZIN", "ZTN", AND "ZWN".)LL 12. ZCOMMAND ZOPERANDS WHOSE EFFECT IS TURNED ON OR OFF BY ALTERNATE OCCURRENCES OF THE ZCOMMAND ZOPERAND ARE: "ZP", "ZH", "ZL", "ZK", "ZH", "ZU", AND "Z".)SU ZEXAMPLES OF ZCOMMAND ZWORDS)U)LLLLA

1.)Z MEN ARE SLOW)L TO GRASP NEW IDEAS;)Z

)
)L "TO GRASP" BEGINS A NEW COLUMN-LINE.)LLLLA

2.)Z)LTTTUZ NOW)UZ IS THE TIME.

)
)LZ ZA NEW LINE IS BEGUN, AND BEGINNING AT THE THIRD TAB POSITION IS PRINTED:)U Z NOW)UZ IS THE TIME.)LLP 3. NOTE THAT ")ZCZP" (MEANING "BEGIN NEXT TEXT COLUMN" FOLLOWED BY "START A NEW PARAGRAPH") DOES NOT PRODUCE THE SAME EFFECT AS ")ZP ZC", WHICH IS EFFECTIVELY THE SAME AS ")ZC". THIS IS BECAUSE THE "ZP" BEGINS A PARAGRAPH, BUT THE FOLLOWING "ZC" IMMEDIATELY STARTS A NEW COLUMN. BECAUSE "ZC" ENDS THE COLUMN-LINE ON WHICH IT OCCURS, THE INDENT (IF ANY) AT THE START OF THE PARAGRAPH WAS LOST.)LLLA

4.)M *)L ***)L *****)L *****)L *****)L ***)L *)M

)
)Z ZTHIS PRODUCES:)LLM *)L ***)L *****)L *****)L *****)L ***)L *)MLP 5. THE TEXT "BLANK) BLANK" IS VALID, AND IS NOT INTERPRETED AS THE BEGINNING OF A COMMAND WORD. THUS,)Z "(A+B-)~*~C")Z PRODUCES: "(A+B-)~*~C", BUT)Z "(A+B-)~*~C")Z PRODUCES "(A+B)*C" AND AN ERROR DIAGNOSTIC (CODE 700) FOR AN UNDEFINED ZCOMMAND ZOPERAND !16THE "!!17. (ZTHE DIAGNOSTIC LISTING AT THE END OF THIS MANUAL SHOWS HOW THE ABOVE ERROR WAS DIAGNOSED.))LLP 6. THE TEXT ")!43~ZFZFZF~MZAET~)!43" PRODUCES:)Z "FORMAT,")Z BUT THE TEXT ")!43~ZFZFZF~MZAET~)!43," PRODUCES:)Z "FORMAT~,")Z AND A DIAGNOSTIC MESSAGE FOR AN UNDEFINED ZCOMMAND ZOPERAND, THE COMMA. THUS, PUNCTUATION CHARACTERS SUCH AS COMMAS AND PERIODS SHOULD BE PLACED IMMEDIATELY AFTER THE TEXT WORD THEY WOULD NORMALLY FOLLOW, AND ANY ZCOMMAND ZWORDS SHOULD THEN FOLLOW THE BLANK AFTER THE PUNCTUATION CHARACTER.)LLL W4A

7.)Z)H2 TEXT ... TEXT)H)Z

)
)LH2 ZTHIS TEXT MATERIAL ILLUSTRATES DELAYED COLUMN INDENTATION, PRODUCED BY ZCOMMAND ZOPERAND "ZH". NOTE THAT EITHER ONE OF THE MARGINS, OR BOTH MARGINS (AS IN THIS EXAMPLE), CAN BE DRAWN IN, AS THE USER DESIRES. IN THIS EXAMPLE, THE SECOND PAIR OF COLUMN INDENTATIONS WAS "(5,5)", SO THAT BOTH MARGINS WERE INDENTED 5 SPACES. NOTE ALSO THAT THE FINAL "ZH" COULD HAVE BEEN "ZH2".)LLLLA

8.)Z)I3 TEXT ... TEXT)I3)Z

)
)I3P THIS TEXT MATERIAL ILLUSTRATES IMMEDIATE COLUMN INDENTATION, PRODUCED BY ZC

OMMAND OPERAND "I". IN THIS EXAMPLE, THE THIRD PAIR OF COLUMN INDENTATIONS WAS "(5,0)", SO THAT ONLY THE LEFT MARGIN WAS INDENTED 5 SPACES. THE EXTRA INDENT OF THE FIRST LINE OCCURRED BECAUSE A PARAGRAPH WAS STARTED BY A "P" COMMAND OPERAND.)ILLLA

9.) ()W10JJJJJJJJJJ) (

) L A BLOCK OF 10 BLANK COLUMN-LINES IS LEFT IN THE SAME TEXT COLUMN (ASSUMING OF COURSE THAT THE) ("SPACING OF TEXT LINES") (IS 1). THIS IS USEFUL FOR THE LATER INSERTION OF A PHOTOGRAPH, FOR EXAMPLE.)S V.)H4U SPECIAL OPERANDS FOR CAPITALIZATION AND SPECIAL CHARACTERS, AND THE NON-TRIVIAL (BLANK) HLLP) (FORMAT) (CAN PRODUCE UPPER AND LOWER CASE AND SPECIAL CHARACTERS IN TWO WAYS. IF THE TEXT INPUT IS PUNCHED WITH THE HOLLERITH CODES REPRESENTING THE CHARACTERS DESIRED (SUCH AS ARE PRODUCED BY A TERMINAL OR BY AN UPPER AND LOWER CASE KEYPU NCH, FOR EXAMPLE), THE PROPER CHARACTER REPRESENTATIONS ON OUTPUT ARE SUPPLIED DIRECTLY BY THE HARDWARE OF THE COMPUTER SYSTEM. IF, HOWEVER, AN UPPER AND LOWER CASE KEYPUNCH OR TERMINAL IS NOT USED (OR APPROXIMATED BY MULTI-PUNCHING ON A STANDARD KEYPUNCH), THEN UPPER AND LOWER CASE AND SPECIAL CHARACTERS CAN BE PRODUCED USING THE SPECIAL OPERANDS.)P THERE ARE TWO SPECIAL OPERANDS FOR USE WITH STANDARD) (IBM) (029 AND 026 TYPE KEYPUNCHES. "Z" IS USED FOR CAPITALIZATION AND NUMERIC SUPERSCRIPTS, AND "!" -IS USED TO PRODUCE SPECIAL CHARACTERS. BOTH SPECIAL OPERANDS ARE VALID IN "AS-IS" TEXT MODE. IT IS IMPORTANT TO REMEMBER THAT NEITHER OF THE SPECIAL OPERANDS NEEDS TO BE PRECEDED BY THE ")" ESCAPE CHARACTER.)LLLLW4U THE "Z" SPECIAL OPERAND:)ULP A LETTER PRECEDED IMMEDIATELY BY "Z" IS PRINTED IN UPPER CASE, A NUMBER SO PRECEDED IS PRINTED IN SUPERSCRIP T FORM, AND ANY OTHER SYMBOL SO PRECEDED IS PRINTED PRECEDED BY THE GRAPHIC "Z". IF ONE OF THE) ("SPECIAL") (CONTROL CARDS IS IN EFFECT THE TRANSLATION OF A NUMBER TO A SUPERSCRIP T IS NOT MADE, UNLESS ONLY THE) ("SPECIAL KEYPUNCH IS A 2741 ") (HAS APPEARED. THE "Z" SPECIAL OPERAND MAY BE PRECEDED BY ANY CHARACTER. THE "!" CHARACTER MUST BE MULTI-PUNCHED ON AN) (IBM) (026 TYPE KEYPUNCH. (NOTE THAT THE "Z")U COMMAND)U OPERAND CAUSES ALL FOLLOWING LETTERS TO BE CAPITALIZED, WHEREAS THE "Z")U SPECIAL)U OPERAND CAUSES ONLY THE SINGLE, IMMEDIATELY FOLLOWING, LETTER TO BE CAPITALIZED. REFER BACK TO THE SAMPLE INPUT IN SECTION I I I FOR AN EXAMPLE.))LLW4U THE "!" SPECIAL OPERAND)ULP A SPECIAL CHARACTER IS DEFINED AS ONE WHICH IS NEITHER A LETTER NOR A NUMBER (NORMAL OR SUPERSCRIP T) NOR ONE OF THE FOLLOWING: *\$.-,/. A SPECIAL CHARACTER IS PRODUCED WHENEVER THE STRING OF CHARACTERS "I NN" IS USED, WHERE NN IS ANY NUMBER FROM 10 TO 51: FOR EXAMPLE, !5128 PRODUCES "I28". IF NN IS NOT IN THE RANGE FROM 10 TO 51, THEN "I NN" IS PRINTED.)V

TABS ARE SET AT 20 25 30 36

NULL CHARACTER SWITCH SET TO 2 (USE NONTRIVIALS FOR CENTERING)

GO

)P THE CORRESPONDENCE BETWEEN THE VALUES FOR NN, THE TEN PRINT TRAIN GRAPHICS, THE) (EBCDIC) (HEXADECIMAL CHARACTER CODES, AND THE PUNCHED CARD CODES IS SHOWN BELOW.)LLL)T NN)T TEN)T HEX)T CARD)CODE)LL)T 10)T !10)T 8D)T 12-0-8-5)L)T 11)T !11)T 9D)T 12-11-8-5)L)T 12)T !12)T 8E)T 12-0-8-6)L)T 13)T !13)T A0)T 11-0-8-1)L)T 14)T !14)T 8B)T 12-0-8-3)L)T 15)T !15)T 9B)T 12-11-8-3)L)T 16)T !16)T AD)T 11-0-8-5)L)T 17)T !17)T BD)T 12-11-0-8-5)L)T 18)T !18)T 8C)T 12-0-8-4)L)T 19)T !19)T AE)T 11-0-8-6)L)T 20)T !20)T 9E)T 12-11-8-6)L)T 21)T !21)T BE)T 12-11-0-8-6)L)T 22)T !22)T AB)T 11-0-8-3)L)T 23)T !23)T BB)T 12-11-0-8-3)L)T 24)T !24)T AC)T 11-0-8-4)L)T 25)T !25)T BC)T 12-11-0-8-4)L)T 26)T !26)T 8F)T 12-0-8-7)L)T 27)T !27)T BF)T 12-11-0-8-7)L)T 28)T !28)T 9C)T 12-11-8-4)L)T 29)T !29)T 9F)T 12-11-8-7)L)T 30)T !30)T AF)T 11-0-8-7)L)T 31)T !31)T A1)T 11-0-1)L)T 32)T !32)T 50)T 12)L)T 33)T !33)T 4F)T 12-8-7)L)T 34)T !34)T 5F)T 11-8-7)L)T 35)T !35)T 4C)T 12-8-4)L)T 36)T !36)T 7E)T 8-6)L)T 37)T !37)T 6E)T 0-8-6)L)T 38)T !38)T 4E)T 12-8-6)L)T 39)T !39)T 4D)T 12-8-5)L)T 40)T !40)T 5D)T 11-8-5)L)T 41)T !41)T 7F)T 8-7)L)T 42)T !42)T 7D)T 8-5)L)T 43)T !43)T 4A)T 12-8-2)L)T 44)T !44)T 7B)T 8-3)L)T 45)T !45)T 6C)T 0-8-4)L)T 46)T !46)T 7C)T 8-4)L)T 47)T !47)T 6D)T 0-8-5)L)T 48)T !48)T 5E)T 11-8-6)L)T 49)T !49)T

7A)T 8-2)L)T 50)T !50)T 6F)T 0-8-7)L)T 51)T !51)T 5A)T 11-8-2)SP #I
T SHOULD BE NOTED THAT THE SPECIAL CHARACTERS FROM NN = 32 THROUGH NN = 51 CAN BE
PUNCHED DIRECTLY ON THE STANDARD)# IBM)# 029 #KEYPUNCH, THAT THE "!"~ CHARAC
TER ITSELF MUST BE MULTI-PUNCHED ON AN)# IBM)# 026 TYPE KEYPUNCH, AND THAT !51
43 IS NOT EQUIVALENT TO EITHER THE "Z" #SPECIAL #OPERAND OR THE "Z" #COMMAND #OP
ERAND.)LLLW4U #THE #NON-#TRIVIAL #BLANK)ULP THE CHARACTER PRODUCED BY PUNCHES
IN THE 0, 8, AND 2 ROWS OF A SINGLE CARD COLUMN (WHICH HAS)# EBCDIC)# REPRESENT
ATION #EO) IS REPLACED BY A "NON-TRIVIAL" BLANK; I.E., ONE WHICH IS NEVER ELIMI
NATED BY THE PROGRAM. THE #IB#M 029 #KEYPUNCH HAS A KEY WHICH PROVIDES THIS CON
FIGURATION OF PUNCHES DIRECTLY. THE NON-TRIVIAL BLANK IS TREATED IN ALL RESPECTS
AS IF IT WERE A NON-BLANK CHARACTER EXCEPT THAT IT MAY OR MAY NOT BE THE FIRST
OR LAST CHARACTER UNDERLINED, AND IT MAY OR MAY NOT BE CONSIDERED FOR CENTERING
PURPOSES, BOTH DEPENDING ON THE)# "NULL CHARACTER SWITCH")# SETTING.)P WHEN U
SING AN INPUT DEVICE SUCH AS A 2741 TERMINAL WHICH HAS NO PROVISION FOR ENTERING
THE NON-TRIVIAL BLANK, A SPECIAL TECHNIQUE IS AVAILABLE, THROUGH THE USE OF THE
)# "NONTRIVIAL BLANK")# CONTROL CARD. FOR EXAMPLE, IF THE INPUT TEXT CONTAINS
NO "&" CHARACTERS, THEN THE CONTROL CARD)LLM NONTRIVIAL BLANK REPRESENTED BY 3
2 (&))LLM WOULD CAUSE SUBSEQUENT APPEARANCES OF &'S TO BE CHANGED TO NON-TRIVI
AL BLANKS, UNTIL THE NEXT)# "NONTRIVIAL BLANK")# CONTROL CARD. THUS, IF THE)#
"NULL CHARACTER SWITCH")# IS SET TO 1, THE INPUT TEXT)LLM !40LLM <----|&&&&&&
!40L &&&&&&!----> !40LLM)LLM WOULD CAUSE THE PRINTED RESULT TO APPEAR AS SHOWN
BELOW.)LLM <----|)L ----|---->)LLM IF THE)# "NULL CHARACTER SWITCH")# HA
D BEEN SET TO 2, THEN THE RESULT WOULD HAVE APPEARED AS FOLLOWS:)LLM <----|----
-)L ----|---->)LLM SINCE THE NON-TRIVIAL BLANKS AT THE END OF THE FIRST GROUP
OF CHARACTERS WOULD NOT BE IGNORED FOR CENTERING.)SU #EXAMPLES OF THE #USAGE O
F #SPECIAL #OPERANDS)ULP FOR THESE EXAMPLES, IT IS ASSUMED THAT)# "SPECIAL KE
YPUNCH")# AND)# "SPECIAL PRINT TRAIN")# ARE NOT IN EFFECT. THUS THE INPUT IS
IN UPPER CASE, AND THE RESULTS WILL BE IN LOWER CASE UNLESS A #COMMAND #OPERAND
OR A #SPECIAL #OPERAND FORCES CAPITALIZATION.)LLA

1. !431 PRODUCES: #1
2. !43#P!43#I#R!432 PRODUCES: #P#I#R#2
3. !40!43)# PI*#2)# !40!43 PRODUCES:)# PI*#2)#
4. !40!43 #P#I#R!432 !40!43 PRODUCES:)# PI*#2)#
5. !40#P)# TEXT1 ... TEXTN)# !40#P PRODUCES:)P TEXT1 ... TEXTN)P
6. !40#P)# TEXT!431 ... T!43EXTN)# !40#P PRODUCES:)P TEXT#1 ... T#EXTN)P
7. #D!43#X!432/#D!432!43#Y PRODUCES: D#X#2/D#2#Y
8. !43#E!5110!432!5112!433!5111 PRODUCES: #E!10#2!12#3!11
9. 6!51~PRODUCES: 6!
10. 6!5151 PRODUCES: 6!51
11.)# !43I WISH !43I HAD 53!43!51)# ~PRODUCES: #I WISH #I HAD 53#I
12.)# !43ONCE !43I HAD 25!5143!5148 !43NOW IT!5142S GONE.)#
PRODUCES: #ONCE #I HAD 25!43!48 #NOW IT!42S GONE.
13. !5152 PRODUCES: !52
14. !516 PRODUCES: !6
15. !5124!5127!5125 !40#L !43#X! !40#L !5122!5127!5123 PRODUCES:
!24!27!25

|X|
:22:27:23

16.. 15114X|X151210:5115 PRODUCES: 114X|X1210:15

)
)S# VI.)#U #THE #EDITOR #FACILITY)ULLP #THE #EDITOR FACILITY CAN BE USED TO CH
ANGE, OVERRIDE, COPY, COMBINE, LIST, AND PUNCH CARD IMAGE DATA SETS; IT CAN ALSO
LOCATE WORDS, PHRASES, AND CHARACTER STRINGS WITHIN THE TEXT. THE CARD IMAGE DA
TASETS READ AND WRITTEN BY THE #EDITOR WILL BE CALLED "TAPE DATASETS"; THESE WIL
L USUALLY HAVE BEEN CREATED INITIALLY BY USE OF THE)# "CREATE A TAPE", "LIST TH
E INPUT DATASET",)# OR)# "PUNCH THE INPUT DATASET")# CONTROL CARDS. THE TAPE
DATASETS MAY CONTAIN PORTIONS OF JOBS, WHOLE JOBS, OR MULTIPLE JOBS. WE WILL REF
ER TO THE INPUT DATASET TO BE EDITED AS THE "OLD MASTER", AND TO THE RESULTING O
UTPUT DATASET AS THE "NEW MASTER". THE NEW MASTER IS CONSTRUCTED WITH ALL UNNEED
ED BLANKS REMOVED, IN THE SAME "CONDENSED" FORM AS A DATASET CREATED BY THE)# "
CREATE A TAPE")# CONTROL CARD.)P THE FUNCTIONS PROVIDED BY THE #EDITOR ARE REQ
UESTED BY USING A)U SINGLE)U #EDITOR CONTROL CARD GROUP, WHICH MUST BE THE FIR
ST AND ONLY CONTROL CARD GROUP OF THE JOB, AND WHICH MUST BE READ FROM THE #SYST
EM #INPUT DATASET. (#SEE #SECTION #I#I#I.) THIS #EDITOR CONTROL CARD GROUP MUST
BEGIN WITH THE)# "EDITOR")# OR)# "TAPE INPUT")# CONTROL CARD, WHICH IS THEN
FOLLOWED BY THE DESIRED #EDITOR CONTROL CARDS AND MODIFICATIONS TO THE OLD MASTE
R (IF ANY), AND IT MUST END WITH THE "#G#O" CONTROL CARD.)# FORMAT)# DETERMINE
S FROM THE PRESENCE OF THE)# "EDITOR")# OR)# "TAPE INPUT")# CONTROL CARD THA
T AN EDIT PHASE IS TO PRECEDE THE DOCUMENT PHASE OF THE RUN. IF THE USER REQUEST
S AN EDIT PHASE, THEN THE SUBSEQUENT DOCUMENT PHASE WILL USE THE RESULT OF THE E
DIT PHASE AS ITS INPUT; IN ADDITION, THERE CAN BE)U NO)U FURTHER DOCUMENT OR E
DIT PHASES. WHEN THE END OF THE #EDITOR CONTROL CARD GROUP IS REACHED, NO FURTHER
REFERENCE WILL BE MADE TO THE #SYSTEM #INPUT DATASET.)P AN ERROR DETECTED BY
THE #EDITOR MEANS THAT THE NEWLY EDITED DOCUMENT WILL NOT BE PRODUCED; HOWEVER,
THE EDIT CONTINUES IN ORDER TO DETECT AS MANY ERRORS AS POSSIBLE. IT IS CLEAR TH
AT USER ERRORS MAKE IT IMPOSSIBLE TO KNOW THE INTENTION OF THE USER, AND)# FORM
AT)# THEREFORE MAKES ASSUMPTIONS WHEREVER NECESSARY SO THAT IT CAN CONTINUE THE
EDIT. THUS, ERRORS DETECTED AFTER THE FIRST ERROR MAY BE DUE TO THE ASSUMPTIONS
MADE BY THE PROGRAM, AND NOT DUE TO THE USER. WHETHER OR NOT ERRORS OCCURRED DU
RING THE EDIT,)# FORMAT)# ALWAYS GIVES A LISTING OF ALL THE CONTROL CARD GROUP
S USED, AND A SET OF DIAGNOSTICS IF ANY WERE GENERATED.)P AT THE CONCLUSION OF
AN ERROR-FREE EDIT, THE DOCUMENT IS PRODUCED FROM THE NEW MASTER, UNLESS OTHERWI
SE SPECIFIED (SEE THE)# "\$NO DOCUMENT")# #EDITOR CONTROL CARD), AND THE #EDITO
R CONTROL CARD GROUP WILL APPEAR FIRST WHEN THE CONTROL CARD GROUPS ARE PRINTED
FOLLOWING THE DOCUMENT. IN THE UPPER FAR RIGHT CORNER OF EACH PAGE OF THE DOCUME
NT,)# FORMAT)# WILL PRINT THE FIRST AND LAST CARD IMAGE NUMBERS FROM THE NEW M
ASTER THAT WERE USED IN PRODUCING THAT PAGE. A LISTING OF THE LATEST TAPE INPUT
DATASET (AS DESCRIBED IN THE DISCUSSION OF THE)# "CREATE A TAPE")# CONTROL CAR
D IN #SECTION #I#I#I) IS PRODUCED AFTER A SUCCESSFUL EDIT IF THE #EDITOR CONTROL
CARD GROUP IS BEGUN BY THE)# "EDITOR")# CONTROL CARD.)P THE #EDITOR CONTROL
CARDS ARE COMPLETELY FREE-FORM, AS DESCRIBED AT THE BEGINNING OF #SECTION #I#I#I
. ALL #EDITOR CONTROL CARDS (EXCEPT "#G#O") BEGIN WITH A "\$", WHICH DISTINGUISHE
S THEM FROM ORDINARY CONTROL CARDS. DURING THE EDIT PHASE, ORDINARY CONTROL CAR
DS ARE SIMPLY DATA TO BE EDITED FROM THE OLD MASTER, OR ADDED TO THE NEW MASTER.
THUS, THE)# "EDITOR")# AND)# "TAPE INPUT")# CONTROL CARDS ARE)U ORDINARY)U
CONTROL CARDS; THEY SIMPLY INITIATE THE EDIT PHASE. DUE TO A MACHINE-DEPENDENT
INTERNAL STORAGE LIMITATION, NO #EDITOR CONTROL CARD OPERAND MAY EXCEED 32,767.
THE #EDITOR CONTROL CARDS ARE DESCRIBED IN THE FOLLOWING PARAGRAPHS, AND SOME EX
AMPLES OF #EDITOR CONTROL CARD GROUPS WILL BE GIVEN AT THE END OF THIS SECTION.
)LLLLW4U #EDITING THE #OLD #MASTER)ULLP THREE #EDITOR CONTROL CARDS ARE USER TO
MODIFY THE TAPE INPUT DATASET (THE OLD MASTER) AND PRODUCE A NEW TAPE INPUT DAT
ASET (THE NEW MASTER): THEY ARE)# "\$INSERT", "\$DELETE",)# AND)# "\$END CHANGES
".)# BEFORE DESCRIBING THE FUNCTION OF EACH CONTROL CARD IN DETAIL, WE WILL GIV
E A BRIEF DESCRIPTION OF THE EDITING PROCESS ITSELF.)P TO PERFORM THESE FUNCTIO
NS,)# FORMAT)# FIRST READS AN #EDITOR CONTROL CARD FROM THE #SYSTEM #INPUT DAT

ASET TO DETERMINE THE EDITING FUNCTION DESIRED. MATERIAL IS THEN COPIED FROM THE OLD MASTER TO THE NEW MASTER UNTIL THE EDITOR FINDS THE POSITION ON THE OLD MASTER WHERE THE INSERTION OR DELETION IS TO OCCUR; THIS POSITION IS CALLED THE "EDIT POINT". AFTER DELETING MATERIAL FROM THE OLD MASTER (IF REQUESTED),)# FORMAT)# INSERTS NEW MATERIAL (IF PROVIDED) INTO THE NEW MASTER, UNTIL IT ENCOUNTERS THE NEXT EDITOR CONTROL CARD.)P IN THIS WAY,)# FORMAT)# OBEYS EACH OF THE EDITOR CONTROL CARDS IN TURN, READING CARD IMAGES FROM THE OLD MASTER AND WRITING CARD IMAGES ON THE NEW MASTER. SINCE THE OLD MASTER CONTAINS DATA WHICH CAN BE USED IN THE DOCUMENT PHASE, IT CAN BE READ BY THE EDITOR IN TWO WAYS: NORMAL TEXT, WHICH RUNS FREELY FROM CARD IMAGE TO CARD IMAGE, AND SINGLE CARD IMAGES (ORDINARY CONTROL CARDS, AND "AS-IS" TEXT CARDS). THUS, WHEREAS THE DOCUMENT PHASE READS ITS INPUT IN THREE MODES (ORDINARY TEXT, AS-IS TEXT, AND CONTROL CARD), THE EDIT PHASE READS THE OLD MASTER AND WRITES THE NEW MASTER IN ONLY)U TWO)U MODES. THESE WILL BE CALLED)U WORD)U MODE (CONTAINING THE TEXT OF TITLES AND FOOTERS, AND ORDINARY TEXT), AND)U CARD)U MODE (CONTAINING AS-IS TEXT AND ORDINARY CONTROL CARDS). DIAGNOSTIC 806 OR 814 (SEE SECTION X#I#I) IS ISSUED IF A MODE ERROR OCCURS.)P THE OLD MASTER (PREFERABLY FILE-PROTECTED) IS READ FROM DATASET REFERENCE NUMBER 2, AND THE NEW MASTER IS WRITTEN ON DATASET REFERENCE NUMBER 4. IT IS IMPORTANT TO REMEMBER THAT THE EDIT PHASE, UNLIKE THE DOCUMENT PHASE, READS FROM)U TWO)U SOURCES: FROM THE SYSTEM INPUT DATASET, WHICH CONTAINS EDITOR CONTROL CARDS AND CHANGES TO THE OLD MASTER; AND FROM THE OLD MASTER, WHICH IS TO BE EDITED ACCORDING TO THE INSTRUCTIONS IN THE EDITOR CONTROL CARD GROUP.)P WE WILL NOW DESCRIBE THE THREE EDITOR CONTROL CARDS USED TO PERFORM THE EDITING FUNCTIONS.)LL#W4 \$INSERT BEFORE CARD IMAGE)#U A)#U WORD)#U B)UP #THE CONTENTS OF THE CARDS (IF ANY) BETWEEN THIS EDITOR CONTROL CARD AND THE NEXT EDITOR CONTROL CARD ARE INSERTED INTO THE NEW MASTER AT THE SPECIFIED EDIT POINT, WHICH IS DETERMINED AS FOLLOWS:)LLW2H4 1.-#IF THE INSERTION REFERS TO TEXT OR TITLES (THE OLD MASTER IS BEING READ IN WORD MODE), THEN THE EDIT POINT IS JUST)U BEFORE)U WORD)U B)U ON CARD IMAGE)U A)U OF THE OLD MASTER (WHERE B IS A COUNT OF ONLY THOSE WORDS)U BEGUN)U ON CARD IMAGE A, AND MUST BE OTHER THAN BLANK OR ZERO).)H4LLW2H4 2.-#IF THE INSERTION REFERS TO "AS-IS" TEXT OR CONTROL CARDS (THE OLD MASTER IS BEING READ IN CARD MODE), THEN THE EDIT POINT IS JUST BEFORE CARD IMAGE)U A,)U AND)U B)U MUST BE BLANK OR ZERO. (#IN CARD MODE, INSERTIONS ARE MADE ONE CARD IMAGE AT A TIME, AND DO NOT DEPEND ON THE WORDS ON THE CARD.))H4LL #THE VALUES OF)U A)U (CARD IMAGE NUMBERS) AND)U B)U (NUMBERS OF TEXT AND TITLE WORDS BEGUN ON THAT CARD) TO BE USED WITH THE OLD MASTER ARE FOUND IN THE LISTING PRODUCED WHEN THE OLD MASTER WAS CREATED OR LAST EDITED.)P THE CARDS CONTAINING THE MATERIAL TO BE INSERTED SHOULD BE PREPARED IN THE SAME WAY AS ORDINARY TEXT, TITLE, "AS IS", OR CONTROL CARDS, AS THOUGH THE)# "CARD FIELD THRU 80")# AND)# "029 KEYPUNCH")# CONTROL CARDS ARE IN EFFECT. THESE TWO CONTROL CARDS ALSO PERTAIN TO THE NEW MASTER, BECAUSE THE)# "026 KEYPUNCH", "029 KEYPUNCH",)# AND)# "CARD FIELD")# CONTROL CARDS WILL BE IGNORED AS INSERTIONS. THE)# "CONTROL CARD ENDS IN")# CONTROL CARD MAY BE INSERTED, AND IT WILL TAKE EFFECT DURING THE EDIT.)LL#W4 \$DELETE CARD IMAGE)#U A)#U WORD)#U B)UL OR)L)# \$DELETE CARD IMAGE)#U A)#U WORD)#U B)#U THRU CARD IMAGE)#U C)#U WORD)#U D)UP THE CONTENTS OF THE CARDS (IF ANY) BETWEEN THIS EDITOR CONTROL CARD AND THE NEXT EDITOR CONTROL CARD ARE INSERTED INTO THE NEW MASTER AT THE EDIT POINT. THEN, THE MATERIAL IN THE OLD MASTER FROM A,B THROUGH AND INCLUDING C, D (IF SPECIFIED) IS SKIPPED OVER, AND IT WILL NOT APPEAR IN THE NEW MASTER. THE DESCRIPTION OF THE)# \$INSERT)# CARD APPLIES, WITH THE REMARKS CONCERNING THE VALUE OF B ALSO APPLYING TO THE VALUE OF D. IF IT IS DESIRED TO DELETE FROM A,B TO THE END OF THE OLD MASTER, THE VALUE 32767 MAY BE GIVEN TO C TO REDUCE RUN TIME (NO OPERAND IS REQUIRED FOR D).)P WITH A SINGLE EXCEPTION, EACH)# \$INSERT)# AND)# \$DELETE)# EDITOR CONTROL CARD MUST REFER TO AN EDIT POINT IN THE OLD MASTER BEYOND THE LAST POINT REFERENCED. THE ONE EXCEPTION TO THIS RULE IS THAT MULTIPLE SUCCESSIVE)# \$INSERT)# REFERENCES TO THE)U SAME)U A AND B ARE ALLOWED; THE INSERTIONS WILL APPEAR IN THE SAME ORDER IN THE NEW MASTER.)P NO)# \$INSERT)# OR)# \$DELETE)# CONTROL CARD CAN BE ALLOWED FOLLOWING A)# \$DUPLICATE)# OR)# \$END CHANGES)# CONTROL CARD, BECAUSE EACH OF THESE PLACES THE EDIT POINT AT THE END OF THE OLD MASTER, BEYOND WHICH THERE IS NO LEGITIMATE POINT. NO)#

\$INSERT)# OR)# \$DELETE)# CONTROL CARD IS ALLOWED IN THE SAME #EDITOR CONTROL CARD GROUP WITH)# \$MERGE)# OR)# \$JOIN)# CONTROL CARDS; THAT IS, CHANGES AND MERGES MUST BE ACCOMPLISHED IN SEPARATE RUNS.)LLL#W4 \$END CHANGES)#P #THIS CONTROL CARD IS REQUIRED FOLLOWING THE LAST)# \$INSERT)# OR)# \$DELETE)# CONTROL CARD, UNLESS THE END OF THE OLD MASTER HAS BEEN REACHED. IT COMPLETES THE NEW MASTER BY ADDING TO IT THE UNREFERENCED LAST PORTION OF THE OLD MASTER. THIS CONTROL CARD IS IGNORED WHEN NOT REQUIRED (THAT IS, WHEN THE END OF THE OLD MASTER HAS BEEN REACHED).)LLLL#W4UP COMBINING DATA SETS)UFLLLP TO COMBINE TAPE INPUT DATASETS INTO A SINGLE NEW TAPE INPUT DATASET (THE "NEW MASTER"):)LLL#W4 \$MERGE TAPE INPUT DATASETS ON)#U X1,...,X8)UP #THE NEW MASTER IS PRODUCED AT DATASET REFERENCE NUMBER 2 AND IS AN UNCHANGED CONCATENATION OF THE TAPE INPUT DATASETS AT THE DATASET REFERENCE NUMBERS GIVEN IN THE OPERAND FIELD, IN THE ORDER IN WHICH THEY ARE GIVEN. UP TO EIGHT DATASET REFERENCE NUMBERS MAY BE SPECIFIED IN ANY ORDER, AND ANY MAY BE SPECIFIED MORE THAN ONCE FOR MULTIPLE COPIES OF PARTICULAR TAPE INPUT DATASETS. THE VALID DATASET REFERENCE NUMBERS ARE 9 AND HIGHER, AND 4. THE USER MUST DETERMINE THAT ALL DATASET REFERENCE NUMBERS USED HAVE BEEN GENERATED INTO THE OPERATING SYSTEM BEING USED.)P AS MANY)# \$MERGE)# AND)# \$JOIN)# CONTROL CARDS AS DESIRED MAY BE USED. NO)# \$MERGE)# OR)# \$JOIN)# CONTROL CARD IS ALLOWED IN THE SAME #EDITOR CONTROL CARD GROUP WITH)# \$INSERT, \$DELETE)# OR)# \$DUPLICATE)# CONTROL CARDS; THAT IS, MERGES MUST BE ACCOMPLISHED IN A SEPARATE RUN FROM CHANGES AND DUPLICATION.)P IT IS GOOD PRACTICE THAT THE TAPE INPUT DATASETS BE FILE-PROTECTED.)LLL#W4 \$JOIN TAPE INPUT DATASETS ON)#U X1,...,X8)UP #THIS CONTROL CARD PRODUCES A RESULTANT NEW MASTER LIKE THE ONE PRODUCED BY THE)# \$MERGE)# CONTROL CARD, WITH ONE DIFFERENCE: ALL DOCUMENT-ENDING "#E" #COMMAND #OPERANDS ENCOUNTERED ON THE TAPE INPUT DATASETS REFERENCED ARE CHANGED TO "#V" #COMMAND #OPERANDS, EXCEPT FOR THOSE ON THE LAST DATASET REFERENCED. THE EFFECT OF THIS IS TO COMBINE THE INPUT FOR MANY JOBS INTO INPUT FOR ONE NEW JOB.)P THE REMAINDER OF THE DESCRIPTION OF THE)# \$MERGE)# CONTROL CARD APPLIES TO THIS CONTROL CARD.)LLLL#W6U #OTHER #EDITOR #CONTROL #CARDS)ULL# \$DUPLICATE OLD MASTER)#P #THE OLD MASTER IS COPIED FROM THE POSITION AT WHICH THE LAST)# \$INSERT)# OR)# \$DELETE)# CONTROL CARD HAS LEFT IT; OR, IF NO POSITION WAS SPECIFIED, FROM THE BEGINNING. THE OLD MASTER (PREFERABLY FILE-PROTECTED) IS MOUNTED AT DATASET REFERENCE NUMBER 2, AND THE COPY IS WRITTEN AT DATASET REFERENCE NUMBER 4. THIS CONTROL CARD MAY BE USED TO COMPLETE A NEW MASTER BEGUN BY)# \$INSERT)# AND)# \$DELETE)# CONTROL CARDS.)P THE)# \$DUPLICATE)# CONTROL CARD IS NOT ALLOWED IN THE SAME #EDITOR CONTROL CARD GROUP WITH)# \$MERGE)# OR)# \$JOIN)# CONTROL CARDS.)LLL#W4 \$NO DOCUMENT)#P #THIS CONTROL CARD PREVENTS PRODUCTION OF THE EDITED DOCUMENT, WHICH OTHERWISE FOLLOWS A SUCCESSFUL EDIT RUN. ANY LISTING, PUNCHING, OVERRIDING, LOCATING, AND DICTIONARY FUNCTIONS THAT MAY HAVE BEEN REQUESTED CONCERNING THE LATEST TAPE INPUT DATASET ARE UNAFFECTED.)LLL#W4 \$OVERRIDE FIRST CONTROL CARD GROUP)#P THE FOLLOWING CARDS (UP TO THE NEXT #EDIT OR OR)# "GO")# CONTROL CARD) ARE CONTROL CARDS WHICH WILL OVERRIDE THE FIRST CONTROL CARD GROUP ON THE TAPE INPUT DATASET WHEN IT IS USED TO PRODUCE THE DOCUMENT. NO)# "TITLE")# OR)# "FOOTER")# CONTROL CARD MAY OVERRIDE. NO PHYSICAL CHANGE IS MADE TO EITHER MASTER.)# FORMAT)# SAVES THE OVERRIDING CONTROL CARD GROUP, AND USES IT AS PART OF THE FIRST GROUP READ FROM THE JUST-COMPLETED NEW MASTER AT THE START OF THE DOCUMENT PHASE. THE OVERRIDING CONTROL CARDS WILL BE INSERTED JUST BEFORE THE FIRST)# "TITLE", "FOOTER",)# OR "#GEO" CONTROL CARD IN THE OVERRIDDEN GROUP.)LLL#W4 \$PUNCH)#P #AT THE CONCLUSION OF THE SUCCESSFUL EDIT RUN AND AFTER THE EDITED DOCUMENT IS PRODUCED OR BYPASSED, THIS CONTROL CARD RESULTS IN THE LATEST TAPE INPUT DATASET BEING COPIED ONTO THE #SYSTEM #PUNCH DATASET.)LLL#W4 \$LIST)#P #THIS CONTROL CARD FORCES A LISTING OF THE NEW MASTER (IF ANY) AT THE CONCLUSION OF AN EDIT RUN, SUCCESSFUL OR NOT. FOLLOWING AN UNSUCCESSFUL EDIT, THE LISTING IS IN UPPER CASE. IN THE LISTING, THE #COMMAND #OPERANDS THAT APPEAR ON EACH CARD IMAGE ARE REITERATED ALONGSIDE THE CARD IMAGES, IN THE RIGHT-HAND PORTION OF THE PAGE. THIS ALLOWS ONE TO FIND #COMMAND #WORDS RAPIDLY, AND TO LOCATE DESIRED AREAS OF THE INPUT TEXT. THOSE SYMBOLS FOR WHICH NO GRAPHICS ARE EXPECTED ARE PRINTED AS ASTERISKS IN THE LISTING.)LLL#W4 \$OMIT LISTING OF NEW MASTER)#P THE PRESENCE OF THIS CONTROL CARD IN THE GROUP OF #EDITOR CONTROL CARDS WILL SUPPRESS THE LISTING OF THE NEW MASTER FOLLOWING A SUCCESSFUL EDIT

DIT. THE DEFAULT ACTION IS TO PRODUCE THE LISTING.)P THE PRODUCTION OF A LISTING DEPENDS ON A NUMBER OF FACTORS. IF THE)EDITOR CONTROL CARD GROUP WAS BEGUN WITH THE)EDITOR)CONTROL CARD, THEN A LISTING WILL BE PRODUCED ONLY IF THE EDIT WAS SUCCESSFUL (IN THE ABSENCE OF A)\$LIST)EDITOR CONTROL CARD). IF THE)EDITOR CONTROL CARD GROUP WAS BEGUN WITH THE)TAPE INPUT)CONTROL CARD, THEN A LISTING IS PROVIDED ONLY IF THE)\$LIST)EDITOR CONTROL CARD IS INCLUDED IN THE)EDITOR CONTROL CARD GROUP. THE)\$OMIT LISTING)EDITOR CONTROL CARD ALWAYS DELETES THE LISTING.)P)FORMAT)WILL USUALLY DIAGNOSE EDITING ERRORS

SO THAT THE CAUSE OF THE ERROR CAN BE IDENTIFIED READILY. IF ERRORS ARE EXPECTED, IT IS SOMETIMES HELPFUL TO INCLUDE THE)\$LIST)EDITOR CONTROL CARD IN THE)EDITOR CONTROL CARD GROUP; THE LISTING CAN THEN BE SCANNED TO SEE WHAT ACTIONS WERE TAKEN BY)FORMAT)IN HANDLING THE ERRORS.)LL\$W4 \$LOCATE THE FOLLOWING WORDS/PHRASES/STRINGS)P THIS FACILITY IS INTENDED PRIMARILY TO ASSIST IN THE TASK OF INDEX PRODUCTION; SEE THE)"DICTIONARY")CONTROL CARD ALSO. THE FOLLOWING CARDS (UP TO THE NEXT)EDITOR OR)"GEO" CONTROL CARD) CONTAIN ARGUMENTS TO BE LOCATED (BY CARD IMAGE NUMBER) IN THE LATEST INPUT STREAM, ACCORDING TO THE FOLLOWING RULES:)I4LLW1H3 !30-ONE SEARCH ARGUMENT PER CARD)HLH3 !30-NON-ALPHAMERICS NOT B+~/*\$ (B = BLANK) ARE IGNORED BOTH IN SEARCH ARGUMENTS AND IN THE TEXT STREAM)HLH3 !30-BLANKS ARE WORD DELIMITERS ONLY, BOTH IN SEARCH ARGUMENTS AND IN THE TEXT STREAM)HLH3 !30-A FINAL NON-BLANK CHARACTER OF "+" IN A SEARCH ARGUMENT MEANS THAT ALL STRINGS CONSISTING OF THE PRECEDING CHARACTERS ARE TO BE LOCATED)HLH3 !30-ALL BLANK SEARCH ARGUMENTS, DUPLICATE SEARCH ARGUMENTS, AND ARGUMENTS CONSISTING OF A SINGLE "+" ARE IGNORED; A "+" IN A SEARCH ARGUMENT IS IGNORED IF THE PRECEDING STRING CONSISTS SOLELY OF A SINGLE CHARACTER)HLH3 !30-SEARCH ARGUMENTS MAY BE IN ANY ORDER)HLH3 !30-A NON-TRIVIAL BLANK IN THE INPUT STREAM IS TREATED AS AN ORDINARY BLANK, BUT A NON-TRIVIAL BLANK IN A SEARCH ARGUMENT IS NOT CHANGED; THUS, NO STRINGS CAN BE LOCATED THAT MATCH A SEARCH ARGUMENT CONTAINING A NON-TRIVIAL BLANK)HLH3 !30-ONLY ORDINARY AND "AS-IS" TEXT ARE SEARCHED ON THE INPUT STREAM;)COMMAND)WORDS, CONTROL CARDS, AND TITLES ARE NOT)HLH3 !30-COMPARISONS ARE MADE ON AN UPPER CASE BASIS; IF TEXT OR SEARCH ARGUMENTS CONTAIN LOWER CASE LETTERS, THEY ARE CONVERTED TO UPPER CASE FOR THE COMPARISON)HLH3 !30-DATASET REFERENCE NUMBER 3 IS REQUIRED (SEE)SECTION)I\$X))HIP)FORMAT)SCANS THE INPUT TEXT FOR WORDS AND STRINGS THAT MATCH A SEARCH ARGUMENT, AND ACCUMULATES AS MUCH DATA AS IT CAN HOLD BEFORE WRITING ANY OUTPUT. WHEN ITS TABLES ARE FULL (OR WHEN ALL THE INPUT TEXT HAS BEEN SCANNED), THE PROGRAM WRITES THE RESULTS ON THE)SYSTEM)OUTPUT DATASET IN ALPHABETIC ORDER, FOR THAT SECTION OF THE INPUT TEXT, WITH THE LOCATIONS OF THE MATCHING STRINGS IN ASCENDING ORDER OF INPUT CARD IMAGE NUMBER. THE SCAN OF THE INPUT TEXT THEN BEGINS AGAIN, IF NECESSARY.)P A SEARCH ARGUMENT WITH A NON-LETTER AS ONE OF THE FIRST TWO CHARACTERS IS POSITIONED AT THE BEGINNING OF THE ENTRIES FOR THE LETTER OF THE FIRST TWO CHARACTERS. THUS, THE LOCATED STRINGS WHICH MATCH "A*" AND "*A" WOULD BOTH BE FOUND AT THE START OF THE LIST OF SEARCH ARGUMENTS BEGINNING WITH THE LETTER "A". A SEARCH ARGUMENT WHICH CANNOT BE FOUND IS SO ANNOTATED.)P IF THE NUMBER OF)\$LOCATE)ARGUMENTS IS TOO LARGE,)FORMAT)WILL PRINT A MESSAGE ON THE)SYSTEM)OUTPUT DATASET, GIVING THE NUMBER OF THE)\$LOCATE)ARGUMENT WHICH CAUSED THE TABLE OVERFLOW. IT AND THE REMAINING ARGUMENTS CAN THEN BE LOCATED IN A SUBSEQUENT COMPUTER RUN.)SU)EXAMPLES OF)P EDITOR CONTROL CARD GROUPS)UPL

L\$A

EDITOR

```

~$DELETE 10 5 ) (DELETE A SINGLE TEXT OR TITLE WORD) )
          !43THE COMPUTER ) (INSERTED TEXT) )
~$INSERT 15 ) (INSERT BEFORE CONTROL CARD) )
          LINES/PAGE = 70 ) (INSERTED CONTROL CARD) )
~$DE 16 0 18 6 ) (DELETE CONTROL CARDS AND TEXT) )
      GO ) (INSERTED CONTROL CARD) )
      )P !43THE DATA ) (INSERTED TEXT) )
)FF $END CHANGES
)FF $PUNCH
)FF $LIST ) (FORCE LISTING OF NEW MASTER) )
GO

```

90

```
)# (THIS GROUP WILL PRODUCE ONLY A DICTIONARY) )#
EDITOR
)FF $DUPLICATE OLD MASTER
)FF $NO DOCUMENT
)FF $OVERRIDE AND PROVIDE A
    DICTIONARY )# (OVERRIDE IS JUST THIS ONE CARD) )#
GO
```

```
TAPE INPUT DATASET
)FF $OVERRIDE CONTROL CARDS
    COLUMNS/PAGE = 2
    SPECIAL PRINT TRAIN
    LIST THE TAPE
)FF $LOCATE THE FOLLOWING:
----TAPE+ )# -----(LOCATES "TAPE", "TAPESTRY", ETC.)
)# ----CONTROL CARD+
    OLD MASTER
    CARD IMAGE+
    COLUMN-LINE
    RIGHT JUSTIFICATION
    TEXT PROCESSING PROG+
GO
```

```
)# (THIS GROUP COMBINES 3 TAPES AND PUNCHES THE RESULT) )#
EDITOR
)FF $PUNCH
)FF $OMIT THE LISTING OF THE NEW MASTER
)FF $JOIN TAPE DATASETS 4, 10, AND 9
GO )#
```

```
)# VII. )# )U #RULES FOR #USING )# FORMAT )#ULLL #A. )U #GENERAL: )P 1.-#EACH )
# FORMAT )# JOB MUST BEGIN WITH A CONTROL CARD GROUP (THE MINIMUM CONTROL CARD G
ROUP CONSISTS OF THE "#G#O" CONTROL CARD). )P 2.-#TITLE, FOOTER, AND TEXT INPUT
MUST APPEAR IN THE FIELD SPECIFIED ON THE )# "CARD FIELD" )# CONTROL CARD, OR, I
F NOT USED, IN THE DEFAULT CARD FIELD (CARD COLUMNS 1 THROUGH 80). )P 3.-#A )# F
ORMAT )# JOB IS ENDED BY THE APPEARANCE OF THE "#E" #COMMAND #OPERAND. MULTIPLE
)# FORMAT )# JOBS MAY BE STACKED ONE BEHIND THE OTHER. IF MIXED CARD AND TAPE RE
SIDENT JOBS ARE TO BE RUN, THE CARD JOBS MUST PRECEDE THE TAPE JOBS (INCLUDING A
N EDIT JOB); THE CARD JOBS MUST NOT USE DATASET REFERENCE NUMBER 2, HOWEVER. )P
4.-#COMMAND #WORDS MAY APPEAR FREELY INTERSPERSED THROUGHOUT TEXT AND TITLES. AL
THOUGH NO TEXT OR TITLE WORD MAY BEGIN WITH A ")" CHARACTER !16NOR THE APPROPRIA
TE "INN" CONFIGURATION FOR ")"!17, THE ")" MAY BE USED TEXTUALLY WHEN FOLLOWED B
Y A BLANK. )LLL #B. )U #TITLES AND #FOOTERS: )U )P 1.-#THE CARD IMAGES CONTAININ
G THE TITLE MUST IMMEDIATELY FOLLOW THE )# "TITLE STARTS ON" )# CONTROL CARD AND
MUST BE IMMEDIATELY FOLLOWED BY EITHER THE )# "FOOTER" )# OR THE "#G#O" CONTROL
CARD. )P 2.-#THE CARD IMAGES CONTAINING THE FOOTING TITLE MUST IMMEDIATELY FOLL
OW THE )# "FOOTER" )# CONTROL CARD AND MUST BE IMMEDIATELY FOLLOWED BY EITHER TH
E )# "TITLE" )# OR "#G#O" CONTROL CARD. )P 3.-#THE "#L", "#F", "#Z", AND "#E" #CO
MMAND #OPERANDS MAY BE USED IN TITLES. )P 4.-#THE TEXT OF A TITLE OR FOOTER MUST
BE ENDED BY THE #COMMAND #OPERAND "#E". )P 5.-#THE "#L" #COMMAND #OPERAND ALWAY
S ACTS AS IF SINGLE SPACING WERE IN EFFECT, REGARDLESS OF THE OPERAND FIELD ON T
HE )# "SPACING OF TEXT LINES" )# CONTROL CARD (OR ITS DEFAULT). THE "#E" #COMMAN
D #OPERAND, IN ADDITION TO ENDING THE TITLE, ALSO SINGLE SPACES. THUS: )LLA
)#          TITLE LINE 1 )LL TITLE LINE 2          )#
)
)#L RESULTS IN EXACTLY ONE BLANK LINE BETWEEN TITLE LINES, WHILE )LLA
)#          LAST TITLE LINE )LLE          )#
)
```

)L RESULTS IN A MINIMUM OF TWO BLANK LINES SEPARATING THE LAST TITLE LINE FROM THE BODY OF THE DOCUMENT.)P 6.-EACH TITLE LINE BEGINS IN THE PRINT POSITION SPECIFIED (OR THE DEFAULT, POSITION 1) AND ENDS WHEN A COMMAND WORD CONTAINING EITHER THE "EL" OR "EE" COMMAND OPERAND IS ENCOUNTERED, OR ELSE WHEN THE TITLE LINE ATTEMPTS TO EXCEED THE LAST PRINTER POSITION ALLOWED TO THE LINE.)P 7.-NO RIGHT-JUSTIFICATION IS ACCORDED TO TITLES, SINCE NO RIGHT-MOST TITLE LIMIT IS DEFINED.)P 8.-ALL HYPHENS APPEARING IN TITLES ARE PRINTED. EXCESS BLANKS ARE IGNORED. SPECIAL SPACING MAY BE ACHIEVED WITH NON-TRIVIAL BLANKS.)P 9.-THE SPECIAL OPERANDS MAY BE USED IN TITLES.)LLL EC.)U BODY OF THE DOCUMENT:)UP 1.-INPUT BLANKS BETWEEN WORDS SERVE ONLY AS WORD DELIMITERS (UNLESS OPERATING IN THE "AS IS" MODE). WORDS ARE SEPARATED BY A SINGLE BLANK, PLUS THE NUMBER OF BLANKS REQUIRED TO ACCOMPLISH RIGHT-JUSTIFICATION, IF IN EFFECT (SEE THE)E "JUSTIFICATION")E CONTROL CARD FOR DETAILS).)P 2.-HYPHENS ARE NOT AUTOMATICALLY INTRODUCED BY)E FORMAT.)E A HYPHEN IN THE INPUT STREAM IS PRINTED, AND MAY BE SELECTED TO BE THE LAST CHARACTER ON A COLUMN-LINE.)SV
TABS ARE AT 10 AND 50
GO

)E VIII.)EU SUMMARY OF)E FORMAT)E CONTROL CARDS AND COMMAND OPERANDS)ULP THE CONTROL CARDS ARE GROUPED BELOW BY THE OPTIONS TO WHICH THEY REFER. THUS, THE)E "JUSTIFICATION")E AND)E "NO JUSTIFICATION")E CONTROL CARDS ARE PAIRED BECAUSE EACH REFERS TO THE RIGHT-JUSTIFICATION OPTION. WITHIN EACH GROUP CERTAIN DEFAULT VALUES WILL BE ASSUMED IF NO CONTROL CARD FROM THAT GROUP IS USED.)L LLLL)TU CONTROL CARDS)UTU IF OMITTED)ULL)LL BACKSPACE CHARACTER IS SPECIAL CHARACTER)EU NN)UD NO BACKSPACES)LL BETWEEN COLUMNS LEAVE)EU X)U BLANKS)ED X=2)LL)E CAPITALIZE AUTOMATICALLY)L NO CAPITALIZATION AUTOMATICALLY)ED ASSUMED)LL CONTROL CARDS END IN COLUMN)EU X)UD X=80)LE CARD FIELD IS)U X)U THRU)U Y)UD X=1, Y=80)LE CARD FIELD EXTENDS THRU)U Y)ULL CENTER TEXT ON LINE)U X)DU X=5)LE START TEXT ON LINE)U X)U IN PRINT POSITION)U Y)UL TEXT STARTS ON LINE)EU X)U IN PRINT POSITION)EU Y)ULL COLUMNS PER PAGE =)EU X)UD X=1)LL COPIES =)U X)UD X=1)LE DARK PRINT EACH PAGE)EU X)EU TIMES)ED X=1)LE OUTPUT MEDIUM IS TAPE)L PRINT OUTPUT TAPE)LL CREATE A TAPE FROM CARD INPUT)L LIST THE INPUT DATASET)L PUNCH THE INPUT DATASET)LL CYCLE THE PAGE NUMBER)L LEFT TOP POSITION FOR PAGE NUMBER)L PAGE NUMBER STARTING AT)EU X)UD X=1)LE RIGHT TOP POSITION FOR PAGE NUMBER)DE ASSUMED)ELL DICTIONARY OF WORDS USED)LL DROP CHARACTER FOR 'D' COMMAND IS)EU X)UD X=75 (DOTS))ELL TAPE INPUT DATASET)L EDITOR)L --\$INSERT)L --\$DELETE)L --\$END CHANGES)L --\$MERGE TAPES)L --\$JOIN TAPES)L --\$DUPLICATE OLD MASTER)L --\$NO DOCUMENT)L --\$OMIT LISTING OF NEW MASTER)L --\$OVERRIDE)L --\$PUNCH)L --\$LIST)L --\$LOCATE)LLW2 FOOTER ON LINE)EU X)EU POS'N)EU Y)EU AFTER)EU Z)EU BLANK LINES)L TITLE STARTS ON LINE)EU X)EU IN PRINT POSITION)EU Y)ULL GO)DD ERROR)ELL IN DENT COLUMN)EU (X1,Y1),...,(X7,Y7))EU POSITIONS)DE X'S,Y'S = 0)ELL JUSTIFICATION)ED ASSUMED)EL NO JUSTIFICATION)LL LINES PER PAGE ARE)EU X)UD X=59)ELL NONTRIVIAL BLANK REP'D BY SPECIAL CHAR)EU NN)UD NN=0)EL NULL CHARACTER SWITCH SET TO)EU X)UD X=1)ELL PARAGRAPH INDENT IS)EU X)U PRINT POSITIONS)DE X=5)ELL REPEAT TITLE ON EVERY PAGE)L STOP PRINTING TITLE)ED ASSUMED)ELL SENTENCES SEPARATED BY AT LEAST)EU X)EU SPACES)ED X=1)ELL SEPARATION LINES BETWEEN PARAGRAPHS ARE)U X)UD X=1)LL SIDE BY SIDE COPIES)LL SPACING OF TEXT LINES IS)EU X)UD X=1)ELL SPECIAL KEYPUNCH)L SPECIAL KEYPUNCH IS A 2741)LL SPECIAL PRINTER TRAIN)LL TABS ARE SET AT)EU X1,...,X14)UD TABS SET TO 0)ELL UNDERLINE SWITCH SET TO)EU X)UD X=0)ELL WIDTH OF COLUMNS IS)EU X)U PRINT POSITIONS)ED X=64)ELL 026 KEYPUNCH)L 029 KEYPUNCH)ED ASSUMED)LLL)SM COMMAND OPERANDS)16FORMAT OF COMMAND WORDS IS ")EX...EY"!17)MLLT EA -- ENTER "AS IS" MODE)LT EC -- BEGIN A NEW COLUMN)LT ED -- TAB TO NEXT TAB STOP, DROPPING DOTS)LT EDN-- TAB TO N-TH TAB STOP, DROPPING DOTS)LT EE -- END THE TITLE OR THE FOOTER, OR END THE JOB)LT EF -- CAPITALIZE FIRST LETTERS OF WORDS / STOP)LT EHN -- INDENT (DELAYED) COLUMN USING NTH PAIR / RESTORE)LT EIN-- INDENT (NOW) COLUMN USING NTH PAIR / RESTORE)LT EJ -- ALWAYS BEGIN A NEW COLUMN-LINE)LT EK -- KEEP THE ENCLOSED TEXT IN ONE TEXT COLUMN)LT EL -- BEGIN A NEW COLUMN-LINE WHEN NOT AT TOP OF COLUMN)LT EM -- CENTER TEXT WITHIN THE COLUMN-LINE / STOP)LT EP -- BEGIN A NEW PARAGRAPH)LT ES -- BEGIN A NEW PAGE)LT ET -- TAB TO NEXT TAB STOP

P)LT #TN-- TAB TO N-TH TAB STOP)LT #U -- UNDERLINE / STOP UNDERLINING)LT #V -
 - READ IN THE NEXT GROUP OF CONTROL CARDS)LT #WN-- KEEP THE NEXT N LINES IN THE
 SAME COLUMN)LT # -- PRINT IN UPPER CASE ONLY / STOP)LLLL)M ! #SPECIAL #OPERA
 ND #VALUES !16#FORMAT IS "#NN"17)L (#T#N #PRINT #TRAIN GRAPHICS SHOWN))LL # 1
 1111111122222222223333333333444444444455)L # 012345678901234567890123456789012
 345678901)LL # !10!11!12!13!14!15!16!17!18!19!20!21!22!23!24!25!26!27!28!29!30!
 31!32!33!34!35!36!37!38!39!40!41!42!43!44!45!46!47!48!49!50!51)LLLL #THE)# EBC
 DIC)# CARD CODE FOR "# IS: 12-8-2)L #THE)# EBCDIC)# CARD CODE FOR "!" IS: 1
 1-8-2)MS# IX.)#U #DATASETS #USED BY)# FORMAT)#ULLP #THE CORRESPONDENCE BETWE
 EN DATASET REFERENCE NUMBERS (#D#R#N) AND SYSTEM DATASET NAMES (WHICH ARE USED I
 N THE NAME FIELD OF SYSTEM CONTROL CARDS) IS AS FOLLOWS:)LLV

TABS SET TO 24 33

CO

)#TU -DRN)T -OS/360)LLU)T #1)T FT01F001)L)T #2)T FT02F001)L)T #3)T
 FT03F001)L)T #4)T FT04F001)LT #5)T FT05F001)LT #6)T FT06F001)LT #7)
 T FT07F001)L)T #8)T FT08F001)L)T# ABOVE)T #FTXX#F001)LP #DATASET REFERE
 NCE NUMBERS 5, 6, AND 7 ARE ASSUMED TO APPLY RESPECTIVELY TO THE #SYSTEM #INPUT
 DATASET, THE #SYSTEM #OUTPUT DATASET, AND THE #SYSTEM #PUNCH DATASET.)P THE USE
 R MUST VERIFY THAT THE DATASET REFERENCE NUMBERS HE USES ARE IN FACT AVAILABLE;
 I.E., THAT THEY HAVE BEEN GENERATED INTO THE OPERATING SYSTEM IN USE AT HIS INST
 ALLATION.)P ALL DATASETS CREATED AND USED BY)# FORMAT)# ARE FORMATTED, SEQUEN
 TIAL, AND FIXED LENGTH, AND MAY BE DEFINED AS BLOCKED, IF OPERATING UNDER #RELEA
 SE 18 OR LATER RELEASES. THIS RESTRICTION IN EARLIER RELEASES IS DUE TO #DATA #M
 ANAGEMENT'S INABILITY TO BACKSPACE A BLOCKED DATASET AND NOT TO THE LOGIC OF THI
 S PROGRAM. IF BLOCKED DATASETS ARE USED WITH RELEASES PRIOR TO 18 THE RESULTS WI
 LL BE UNPREDICTABLE.)P LABELED TAPES CAN BE USED BY THE PROGRAM PROVIDING THAT
 THEY ARE ACCEPTABLE TO THE OPERATING SYSTEM USED. LABELING OF TAPES, IF DESIRED,
 IS THE RESPONSIBILITY OF THE USER.)P BEFORE A FILE-PROTECTED TAPE CAN BE READ
 BY #OS/360, THE MESSAGE "XX #IE#C103#D #F" IS TYPED ON THE CONSOLE. THE OPERAT
 OR MUST RESPOND WITH "REPLY XX, #U", WHERE XX IS THE ON-LINE MESSAGE NUMBER.)P
 UNDER #OS/360 THE NUMBER OF #I/#O BUFFERS MAY BE 1 OR 2. THE HIGHER NUMBER IS
 ALWAYS PREFERABLE UNLESS THERE IS DIFFICULTY FITTING THE PROGRAM INTO MEMORY, IN
 WHICH CASE THE NUMBER 1 SHOULD BE SPECIFIED WHERE NECESSARY; HOWEVER, PERFORMAN
 CE MAY BE SOMEWHAT DEGRADED.)P ALL DATASETS CREATED BY)# FORMAT)# ARE ENDED B
 Y AN "END-OF-FILE" MARK.)P THE FOLLOWING DESCRIBES THE DATASETS CREATED AND USE
 D BY)# FORMAT:)#)LLLW4 #DATASET #REFERENCE #NUMBER 1:)P #THIS DATASET RECORD
 S CONTROL CARDS, USER ERRORS, AND OTHER INFORMATION, AND IS ALWAYS REQUIRED. IT
 MAY BE DIRECT ACCESS DEVICE OR TAPE RESIDENT. ITS RECORD LENGTH IS 97 BYTES.)LL
 W4 #DATASET #REFERENCE #NUMBER 2:)P #THIS DATASET IS REQUIRED ONLY IF ONE OR B
 OTH OF THE FOLLOWING APPLY:)LLI3#H 1.-"EDITOR", "TAPE INPUT", "CREATE A TAPE",
 "DICTIONARY", "LIST",)# AND/OR)# "PUNCH")# HAS BEEN SPECIFIED)HLH 2.-#THE #C
 OMMAND #OPERAND "#K" HAS BEEN USED)IHP #THIS DATASET IS A CARD IMAGE SET WHICH
 MAY BE RESIDENT EITHER ON TAPE OR ON A DIRECT ACCESS DEVICE. IF THE #EDITOR FACI
 LITY IS BEING USED, TAPE IS PREFERABLE SINCE THE USER MAY WISH TO KEEP THIS DATA
 SET, FILE-PROTECT IT, AND USE IT AGAIN AS AN INPUT DATASET MASTER.)LLLW4 #DATAS
 ET #REFERENCE #NUMBER 3:)P THIS DATASET IS REQUIRED ONLY IF)# "DICTIONARY")#
 OR)# "\$LOCATE")# HAS BEEN SPECIFIED. IT CONTAINS 80 BYTES PER RECORD AND MAY B
 E TAPE RESIDENT OR (PREFERABLY) ON A DIRECT ACCESS DEVICE.)LLLW4 #DATASET #REFE
 RENCE #NUMBER 4:)P #THIS DATASET IS ONLY REQUIRED WHEN PRODUCING A "NEW MASTER"
 INPUT DATASET (OR A DUPLICATE OF THE "OLD MASTER") IN AN EDIT RUN. IT MAY ALSO
 BE (BUT NOT IN THE SAME RUN) AN INPUT DATASET TO BE #M#E#R#G#E#D OR #J#O#I#N#E#D
 IN AN EDIT RUN. ITS SPECIFICATIONS ARE IDENTICAL TO THOSE FOR DATASET REFERENCE
 NUMBER 2.)LLLW4 #DATASET #REFERENCE #NUMBER 5:)P #THIS IS THE #SYSTEM #INPUT D
 ATASET AND IS ALWAYS REQUIRED BY THE PROGRAM. ITS RECORD LENGTH IS ALWAYS 80 BYT
 ES.)LLLW4 #DATASET #REFERENCE #NUMBER 6:)P #THIS IS THE #SYSTEM #OUTPUT DATASE
 T AND IS ALWAYS REQUIRED BY THE PROGRAM. ITS RECORD LENGTH IS 133 BYTES, AND #A#
 S#A STANDARD CONTROL CHARACTERS ARE USED.)LLLW4 #DATASET #REFERENCE #NUMBER 7:
)P #THIS IS THE #SYSTEM #PUNCH DATASET, AND IS ONLY REQUIRED IF PUNCHED OUTPUT H
 AS BEEN REQUESTED. ITS RECORD LENGTH IS ALWAYS 80 BYTES.)LLLW4 #DATASET #REFERE
 NCE #NUMBER 8:)P #THIS DATASET IS ONLY REQUIRED BY)# FORMAT)# IF ANY OF THE F

NSISTS OF THREE FILES WRITTEN AT A RECORDING DENSITY OF 800 #B#P#I ON A 9-TRACK TAPE, WITH NO LABELS. ALL LOGICAL RECORDS ARE 80 BYTES LONG, AND EACH PHYSICAL RECORD IS 1600 BYTES LONG. THE FIRST FILE CONTAINS THE OBJECT DECK (INCLUDING #LINKAGE #EDITOR CONTROL STATEMENTS); THE SECOND FILE CONTAINS THE)# FORMAT)# JOB WHICH PRODUCES THIS MANUAL; THE THIRD FILE CONTAINS THE #FORTRAN SOURCE STATEMENTS FROM WHICH THE OBJECT DECK WAS PRODUCED.)S #X#I.)U #HINTS AND #SUGGESTIONS.)U)LLLLF #A.)U DOCUMENT PHASE)FULLL 1. THE)# TITLE)# AND)# FOOTER)# CONTROL CARDS, ALONG WITH THEIR FOLLOWING TITLE AND FOOTER TEXTS, MUST BE THE LAST CONTROL CARDS TO APPEAR IN A CONTROL CARD GROUP BEFORE THE #G#O CONTROL CARD.)LL L 2. IF THE TEXT FOR A TITLE OR FOOTER IS NOT ENDED WITH THE ")#E" #COMMAND #OPERAND,)# FORMAT)# WILL SEARCH FOR IT BY INCLUDING AS MUCH OF THE FOLLOWING MATERIAL AS POSSIBLE INTO THE "TITLE". THIS NATURALLY LEADS TO A DOCUMENT OF UNUSUAL PROPORTIONS.)LLL 3. WHEN ENDING AN "AS-IS" REGION (INITIATED BY THE "#A" #COMMAND AND #OPERAND), THE CARD CONTAINING THE ")#~" IN THE INITIAL COLUMNS SHOULD CONTAIN NO OTHER TEXT.)LLL 4. IF AN ERRONEOUS CONTROL CARD IS FOUND, IT IS TREATED BY)# FORMAT)# AS A "#G#O" CARD. THIS MEANS THAT ANY FOLLOWING CONTROL CARDS WILL BE READ IN TEXT MODE; IN PARTICULAR, IF A)# "TITLE")# CARD FOLLOWS THE BAD CONTROL CARD, THE "!40#E" THAT ENDS THE TITLE (OR FOOTER, OF COURSE) WILL APPEAR TO BE THE "!40#E" THAT ENDS THE TEXT INPUT.)LLL 5. WHEN SETTING UP TAB STOPS AND COLUMN INDENTS, REMEMBER THAT A TAB STOP IN (SAY) COLUMN 10 IS EQUIVALENT TO AN INDENT OF 9 SPACES -- THAT IS, THE LINE POSITION WHERE THE TEXT WILL BEGIN AFTER INDENTING IS 1 LARGER THAN THE NUMBER OF SPACES INDENTED.)LLLF #B.)U EDIT PHASE)FULLL 1. A SUCCESSFUL EDIT DOES NOT IMPLY A SUCCESSFUL DOCUMENT, SINCE CONFLECTING INFORMATION MAY HAVE BEEN EDITED INTO THE NEW MASTER.)LLL 2. CONTROL CARDS WRITTEN ONTO THE NEW MASTER ARE UNDER CONTROL OF THE)#LLM CONTROL CARD ENDS IN COLUMN)#U NN)UMLL CARD CURRENTLY IN EFFECT.)LLL 3. DURING AN EDIT, THE METHOD USED TO SEARCH FOR #EDITOR CONTROL CARDS CAN OCCASIONALLY CAUSE A NON-CONTROL CARD TO BE MISTAKEN FOR AN #EDITOR CONTROL CARD. (#DURING AN EDIT, EACH CARD IN THE #EDITOR CONTROL CARD GROUP MUST BE CHECKED TO SEE IF IT IS AN #EDITOR CONTROL CARD, OR TEXT TO BE INSERTED INTO THE NEW MASTER.) THE VALID #EDITOR CONTROL CARD CHARACTERS ARE SHOWN IN THE LEFTMOST COLUMN OF THE TABLE BELOW; THE INVALID COMBINATIONS THAT WILL BE MISTAKEN FOR THE VALID COMBINATIONS ARE SHOWN IN THE RIGHT COLUMNS.)LLW16V

TABS AT 22. 32

GO

)PUT VALID)UTU INVALID)ULLT# \$ME)T \$J5 \$KV \$LN)LT \$OV)T \$N5 \$PN \$QE)LT \$DE)T \$A5 \$CN \$BV)LT \$IN)T \$G5 \$HV)LT \$EN)T \$C5 \$FE \$DV)LT \$DU)T \$C4 \$EM \$PD)LT \$PU)T \$O4 \$QM \$RD)LT \$NO)T \$L6 \$NW \$OE)LT \$LI)T \$KR \$JZ)LT \$LO)T \$KW \$J6)LT \$JO)T \$KF)LT \$OM)T \$M4 \$NU \$PD)#LLM EDITOR CONTROL CARD EQUIVALENCE S)FLLM #TO AVOID SUCH ERRORS, (1) ARRANGE THE TEXT TO BE INSERTED SO THAT THE FIRST NONBLANK CHARACTER ON THE INPUT CARD IS NOT A "\$", OR (2) BE SURE THAT THE FIRST THREE CHARACTERS ARE NOT ONE OF THE INVALID COMBINATIONS.)LLL 4. THE)# DICTIONARY)# FEATURE REQUIRES THAT THE)U SOURCE)U TEXT BE IN UPPER CASE; TEXT ENTERED WITH THE)# "SPECIAL KEYPUNCH")# CONTROL CARD IN EFFECT MAY NOT BE PROCESSABLE BY THIS FACILITY.)SV

TAB SET AT 15

GO

)# XII.)#U #ERROR #HANDLING AND #DIAGNOSTIC #MESSAGES)ULLP #WITH THE EXCEPTION OF ERRORS MADE DURING AN EDIT RUN, USER ERRORS DO NOT ABROGATE THE DOCUMENT. WHEN A USER ERROR IS FOUND, THE PROGRAM NOTES THE ERROR, ASSUMES APPROPRIATE VALUES FOR THE ERRONEOUS DATA, AND CONTINUES. THE #EDITOR DOES NOT ALLOW A DOCUMENT TO BE PRODUCED UNLESS THE EDIT WAS ERROR-FREE; HOWEVER, THE EDIT ITSELF CONTINUES TO COMPLETION REGARDLESS OF USER ERRORS.)P THE ERROR DIAGNOSTICS (IF ANY) ARE WRITTEN ONTO THE #SYSTEM #OUTPUT DATASET AT THE CONCLUSION OF EACH JOB. EACH DIAGNOSTIC CONSISTS OF A TEXTUAL DESCRIPTION OF THE ERROR AND THE PAGE NUMBER, COLUMN NUMBER, AND LINE NUMBER BEING PRODUCED WHEN IT OCCURRED. IF THE ERROR WAS IN THE INPUT TEXT, THE CHARACTER NUMBER WITHIN THE LINE WHERE THE ERROR OCCURRED IS GIVEN; IF THE ERROR OCCURRED WITHIN A CONTROL CARD GROUP, THEN THE GROUP NUMBER IS GIVEN; AND IF THE ERROR CAN BE LOCALIZED TO A PARTICULAR CONTROL CARD OR #EDITOR INSERTION CARD, THEN THE CARD NUMBER IS GIVEN. ALSO LISTED FOR EACH ERROR IS

A CODE NUMBER THAT REFERS TO A PARAGRAPH BELOW, WHICH GIVES ADDITIONAL INFORMATION ABOUT THE ERROR AND DESCRIBES ACTION TAKEN BY THE PROGRAM WHEN IT OCCURS.)LLW3H3 212!29)% CONTROL CARD OPERAND IN ERROR)%L %AN OPERAND ON THE CONTROL CARD SPECIFIED IS OUTSIDE THE LEGAL RANGE OR IS OTHERWISE IN ERROR. IF THE ERROR OCCURS ON AN %EDITOR CONTROL CARD, THE CONTROL CARD IS IGNORED. OTHERWISE, THE PREVIOUS VALUE OF THE PARAMETERS INVOLVED OR, IF NONE, THE DEFAULT VALUES ARE USED.)HLLW3H3 218!29 UNRECOGNIZED CONTROL CARD)%L %THE SPECIFIED CONTROL CARD IS UNRECOGNIZABLE. IT IS TREATED AS IF IT WERE THE)% "SPECIAL PRINTER")% AND THE)% "GO")% CONTROL CARDS. IF THE INPUT STREAM IS NOT ON THE %SYSTEM %INPUT DATASET, IT IS BACKSPACED AND THE UNRECOGNIZABLE CONTROL CARD IS REREAD AS TEXT.)HLLW3H3 219!29 NUMBER OF PRINT POSITIONS REQUIRED NOT AVAILABLE)%L %THE NUMBER OF PRINT POSITIONS REQUIRED BY THIS CONTROL CARD GROUP EXCEEDS THE NUMBER AVAILABLE. THE DOCUMENT IS FORCED LEFTWARD, THE WIDTH OF THE TEXT COLUMNS MAY BE REDEFINED TO BE THE LARGEST VALUE POSSIBLE, AND THE NUMBER OF PRINT POSITIONS BETWEEN COLUMNS MAY BE SET TO 2.)HLLW3H3 220!29 TITLE/FOOTER TOO LONG)%L %THE TITLE OR FOOTER IS NOT ENDED AFTER THE LAST LINE ALLOTTED TO THE PAGE IS FILLED. THE TITLE OR FOOTER IS ENDED AND THE PROGRAM LOOKS FOR A CONTROL CARD. IF ISSUED FOR A TITLE, THE)% "STOP TITLE")% CONTROL CARD IS SIMULATED.)HLLW3H3 237!29 TABS NOT IN ASCENDING ORDER)%L %THE TABS SET IN THE SPECIFIED CONTROL CARD GROUP ARE NOT IN ASCENDING ORDER. STARTING WITH THE FIRST TAB SET OUT OF ORDER, THE TABS ARE SET TO THE LAST POSITION ON THE COLUMN-LINE.)HLLW3H3 249!29 CONTROL CARD NOT FIRST, OR ON DATASET OTHER THAN 5)%L %THE)% "EDITOR")% OR)% "TAPE INPUT DATASET")% CONTROL CARD SPECIFIED IS EITHER NOT THE FIRST CARD OF THE JOB, OR ELSE IT HAS BEEN READ FROM A DATASET WHICH IS NOT THE %SYSTEM %INPUT DATASET (DATASET REFERENCE NUMBER 5). IT IS IGNORED, AND ANY FOLLOWING %EDITOR CONTROL CARDS OR INSERTIONS WILL NOT BE PROPERLY INTERPRETED.)HLLW3H3 267!29 TAB IMPROPERLY SET)%L %IN THE SPECIFIED CONTROL CARD GROUP A TAB IS SET AT A POSITION BEYOND THE END OF THE COLUMN-LINE. THE ERRONEOUSLY SET TAB AND THE TABS WHICH FOLLOW IT ARE SET TO THE LAST POSITION ON THE COLUMN-LINE.)HLLW3H3 269!29 IMPROPER STARTING LINE FOR DOCUMENT TEXT)%L %THE BODY OF THE DOCUMENT IS POSITIONED IMPROPERLY BY THE SPECIFIED CONTROL CARD GROUP. THE CORRECTIVE ACTION TAKEN IS TO BEGIN THE TEXT IMMEDIATELY FOLLOWING THE TITLE (BUT NOT ABOVE LINE 5), AND THE TEXT IS EXTENDED THROUGH THE LAST LINE ON THE PAGE.)HLLW3H3 289!29 IMPROPER CONTROL CARD ORDER)%L %THE REFERENCED CONTROL CARD IS NEITHER THE)% "TITLE", "FOOTER",)% NOR "%GEO" CONTROL CARD. AN ATTEMPT IS MADE TO ALLOW THE PRESENT CONTROL CARD ORDER.)HLLW3H3 300!29 INDENTS TOO LARGE)%L %THE CUMULATIVE INDENTS IN EFFECT HAVE REDUCED THE EFFECTIVE COLUMN-LINE WIDTH TO ZERO OR LESS. ALL COLUMN INDENTS ARE TURNED OFF AT THE INDICATED CHARACTER POSITION.)HLLW3H3 304!29 CHARACTER STRING LENGTH EXCEEDS COLUMN WIDTH)%L %A STRING OF NON-BLANK, UNHYPHENATED CHARACTERS AT THE INDICATED CHARACTER POSITION IS LONGER THAN THE COLUMN-LINE. IT IS PRINTED WITHOUT HYPHENATION OVER AS MANY LINES AS ARE REQUIRED TO CONTAIN IT.)HLLW3H3 327!29 TAB COMMAND OPERAND IMPROPERLY USED)%L %THE %COMMAND %OPERAND "%T" OR "%D" AT THE INDICATED CHARACTER POSITION IS BEYOND THE POSITION OF ANY TAB SET, OR IS NOT TO THE RIGHT OF THE CURRENT CHARACTER POSITION, OR IS IN AN INDENTED PORTION OF THE COLUMN-LINE, OR AN UNSET TAB HAS BEEN USED. THE %COMMAND %OPERAND IS IGNORED.)HLLW3H3 513!29 NUMBER OF UNDERLINE SEGMENTS ON PAGE EXCEEDS 99)%L %AT THE INDICATED CHARACTER POSITION MORE THAN 99 COLUMN-LINES, OR PORTIONS OF COLUMN-LINES, HAVE BEEN UNDERLINED ON THIS DOCUMENT PAGE. THOSE IN EXCESS OF 99 ARE IGNORED.)HLLW3H3 700!29 UNDEFINED COMMAND OPERAND)%L %A %COMMAND %WORD AT THE INDICATED CHARACTER POSITION (BEFORE THE LINE IS JUSTIFIED) CONTAINS AN UNDEFINED %COMMAND %OPERAND. IT, AND THE REST OF THE %COMMAND %WORD, ARE TREATED AS TEXT. THE ")" IS ALSO PRINTED IF THE UNDEFINED %COMMAND %OPERAND IS THE FIRST IN THE %COMMAND %WORD. IF THE ERROR WAS DETECTED DURING AN EDIT, THE NUMBER GIVEN FOR THE ERRONEOUS CONTROL CARD WILL BE THAT OF THE LAST ONE READ BEFORE THE ERROR WAS DETECTED. IF THE INVALID %COMMAND %OPERAND IS A ")", THEN)% FORMAT)% WILL TREAT IT AS THE START OF A NEW %COMMAND %WORD IF IT IS NOT FOLLOWED BY A BLANK.)HLLW3H3 800!29 UNEXPECTED END OF INPUT)%L %AN UNEXPECTED END OF THE INPUT STREAM HAS OCCURRED, CAUSED BY AN OMITTED "%GEO" OR)% "\$END CHANGES")% CONTROL CARD, OR BY AN OMITTED "%E" %COMMAND %OPERAND. SOME OUTPUT MAY BE LOST.)HLLW3H3 802!29 NEW MASTER ALREADY FINISHED)%L %AN ATTEMPT HAS BEEN MADE BY THE

SPECIFIED CONTROL CARD TO CONTINUE THE NEW MASTER AFTER THE END OF THE OLD MASTER HAS BEEN REACHED. FOR EXAMPLE, A)% "\$DELETE")% CONTROL CARD MAY HAVE OCCURRED AFTER A)% "\$DUPLICATE")% CONTROL CARD. THE EDIT CONTINUES.)HLLW3H3% 804!29 EDIT FAILED BECAUSE OF ABOVE ERROR(S) OR BECAUSE NEW MASTER NOT FINISHED)%L %E ERRORS ALREADY NOTED HAVE OCCURRED DURING THE EDIT ENDED BY THE REFERENCED CARD, OR ELSE THE NEW MASTER HAS NOT BEEN ENDED BECAUSE THE END OF THE OLD MASTER HAS NOT BEEN REACHED OR REFERENCED. THE JOB IS TERMINATED.)HLLW3H3% 805!29 REFERENCED WORD NOT LOCATED)%L %THE WORD REFERENCED ON THE SPECIFIED)% "\$INSERT")% CONTROL CARD, OR THE FIRST WORD REFERENCED ON THE SPECIFIED)% "\$DELETE")% CONTROL CARD CANNOT BE LOCATED. THE EDIT CONTINUES.)HLLW3H3% 806!29 INPUT/OPERAND MODE ERROR)%L %THE MODE OF THE OPERAND ON THE SPECIFIED)% "\$INSERT")% OR)% "\$DELETE")% CONTROL CARD DIFFERS FROM THE PRESENT MODE OF THE NEW INPUT DATASET (NEW MASTER). THAT IS, A WORD NUMBER IS SPECIFIED AND THE NEW MASTER IS IN AN "AS IS" OR CONTROL CARD REGION (CARD MODE), OR NO WORD NUMBER IS SPECIFIED AND THE NEW MASTER IS IN A TEXT REGION (WORD MODE). THE EDIT CONTINUES.)HLLW3H3% 807!29 END OF \$DELETE FIELD NOT FOUND)%L %THE END OF THE FIELD TO BE DELETED, REFERENCED ON THE SPECIFIED)% "\$DELETE")% CONTROL CARD, CANNOT BE LOCATED. THE EDIT CONTINUES.)HLLW3H3% 814!29 NON-TEXT MODE NOT ENDED)%L %AN "AS IS" OR CONTROL CARD REGION EDITED INTO THE MIDDLE OF A TEXT CARD IMAGE HAS NOT BEEN ENDED BEFORE THE SPECIFIED CONTROL CARD. THE EDIT CONTINUES IN "TEXT" (WORD) MODE.)HLLW3H3% 847!29 \$INSERT/\$DELETE/\$DUPLICATE AND \$MERGE/\$JOIN NOT ALLOWED IN SAME RUN)%L %EDITOR CONTROL CARDS)% "\$INSERT", "\$DELETE",)% AND)% "\$DUPLICATE")% MAY NOT APPEAR IN THE SAME RUN WITH)% "\$MERGE")% AND)% "\$JOIN")% CONTROL CARDS. THAT IS, MERGES MUST BE ACCOMPLISHED IN A SEPARATE RUN FROM CHANGES AND DUPLICATION. THE INDICATED CONTROL CARD IS IN VIOLATION OF THIS RULE. THE EDIT CONTINUES.)HLLW3H3% 857!29 NOT ALLOWED)%L %THE)% "TITLE")% OR)% "FOOTER")% CONTROL CARD INDICATED IS NOT PERMITTED AS AN OVERRIDING CONTROL CARD. THE EDIT CONTINUES.)HLLW3H3% 922!29 NO TEXT AFTER TAB(S))%L %THE LAST TAB ON THE INDICATED LINE IS NOT FOLLOWED BY TEXT.)HLLW3H3% 997!29 TOO MANY BACKSPACES ON ONE PAGE)%L %TOO MANY BACKSPACES HAVE BEEN SPECIFIED ON THE CURRENT PAGE. THE FIRST 99 HAVE BEEN HANDLED, BUT ANY AFTER THE 100TH WILL BE TREATED AS NORMAL TEXT CHARACTERS.)S% XIII.)%U %APPENDIX)ULLP %THE FOLLOWING PAGES WERE PRODUCED AT THE CONCLUSION OF THE COMPUTER RUN PRODUCING THIS MANUAL. THE)% "COLUMNS PER PAGE = 9")% CONTROL CARD, THE FIRST CONTROL CARD IN THE FIRST CONTROL CARD GROUP, IS INTENTIONALLY FAULTY, AND PRODUCES THE FIRST DIAGNOSTIC. THE FIFTH EXAMPLE OF %COMMAND %WORDS (AT THE END OF %SECTION %I%V) PRODUCES THE SECOND.)SV

COLUMNS/PAGE = 2

WIDTH = 30

BETWEEN COLUMNS = 4

PAGE NUMBER IS NULL

NO CAPITALIZATION AUTOMATICALLY

NO JUSTIFICATION

STOP TITLE

TITLE ON LINE 5

)% INDEX)%E

GO

)LLM %A)HL %A%\$A, 52)L %ASTERISK, 12, 41)L "%AS IS", 24, 4,)H6 7, 12, 18, 25, 28, 31, 37, 42, 45, 55, 60, SEE ALSO %COMMAND %OPERAND "%A")HL)H6 "%AS IS" TEXT MODE, 4, 7, 24, 31, 60)HL)LLM %B)HL% "BACKSPACE CHARACTER",)%H6 9, 10, 23, 27, 46, 60)HL %BACKSPACING OF FILES, 50, 57)L)% "BETWEEN COLUMNS",)% 10, 21, 46)L %BLANK,)H6 1, 2, 4-12, 16, 27, 41, 42, 45, SEE ALSO %NON-TRIVIAL BLANK)HL %BLANK LINES,)H6 14, 16, 19, 26, 45)HL %BLOCKED DATASETS, 50)L)LLM %C)HL %CAPITALIZATION,)H6 SEE %SPECIAL %OPERAND "%", %COMMAND %OPERANDS "%" AND "%F", AND)% "CAPITALIZE AUTOMATICALLY")%HL)%H6 "CAPITALIZE AUTOMATICALLY",)% 11, 26, 27, 46)HL)H6% "CARD FIELD",)% 11, 12, 37, 38, 44, 46)HLH6 %CENTERING, 7, 17, 20, SEE ALSO %COMMAND %OPERAND "%M" AND)% "CENTER TEXT ON")%H6L)% "CENTER TEXT ON",)% 11, 46)L)H6% "COLUMNS PER PAGE",)% 11, 21, 46, 61)HL %COMMAND %OPERANDS,)H6 2, 4-8, 12, 24, 28, 29, 41, 49)HL)I7 "%A", 24, 28, 55)L "%C", 24, 28)L "%D", 24, 14, 27, 28, 59)C)LH6 "%E", 25, 7, 21, 28, 39, 44, 45, 55, 59)HL "%F", 25-28, 44)L "%H", 25, 6, 7, 28)L "%I", 25, 6, 28)L "%J", 2

5, 28)L "K", 26, 10, 27, 28, 51)L "L", 26, 5, 25, 28, 44, 45)L "M", 26, 7, 17, 28)L "P", 26, 5, 11, 25, 27, 28)L "S", 26, 11, 25, 27, 28)L "T", 25-28, 59)L "U", 27, 17, 26, 28)L "V", 27, 5, 25, 28, 39)L "W", 27, 28)L "Z", 27, 28, 31, 44)L #COMMAND #WORD,)H6 24, 2, 4, 7, 22, 41, 42, 44)H)L #CONCATENATION, 39)L #CONDENSING, 12, 18, 35)L #CONSOLE MESSAGE, 50)L)#H6 "CONTROL CARDS END IN",)# 11, 38, 46, 55)HL #CONTROL CARD GROUP,)H6 4, 6, 9, 15, 27, 35-41, 43, 44, 49, 55-57, 61)HL #CONTROL CARD MODE, 4, 5, 7)L #CONTROL VARIABLES, 4, 5, 7)L)# "COPIES",)# 12, 17, 18, 46, 52)L)# "CREATE A TAPE",)#H6 12, 16, 18, 35, 46, 51)HL)#H6 "CYCLE THE PAGE NUMBER",)# 13, 46)HL)LLM #D)HL)# "\$DELETE",)H6# 38, 36, 39, 40, 47, 59, 60)HL)H6# "\$DUPLICATE",)# 40, 38, 39, 47, 59, 60)HL# "DARK PRINT",)# 13, 23, 46)L #DATASET REFERENCE NUMBER 1, 51)L #DATASET REFERENCE NUMBER 2,)H6 51, 12, 14, 20, 37, 40, 44)HLH6 #DATASET REFERENCE NUMBER 3, 51, 14, 42)HLH6 #DATASET REFERENCE NUMBER 4, 51, 37, 39, 40)HLH6 #DATASET REFERENCE NUMBER 5, 52, 58, SEE ALSO #SYSTEM #INPUT #DATASET)HLH6 #DATASET REFERENCE NUMBER 6, 52, SEE ALSO #SYSTEM #OUTPUT #DATASET)HLH6 #DATASET REFERENCE NUMBER 7, 52, SEE ALSO #SYSTEM #PUNCH #DATASET)HLH6 #DATASET REFERENCE NUMBER 8, 52, 17)HLH6 #DIAGNOSTIC MESSAGES, 57, 4, 7, 18, 27, 29, 37, 61)HL)# "DICTIONARY OF WORDS USED",)#H6 13, 40, 41, 47, 51, 56)HL #DOCUMENT PHASE, 4, 55)L #DOTS, 24)L)# "DROP CHARACTER",)#H6 14, 24, 47)HL)LLM #E)HL)# "\$END CHANGES",)# 38, 36, 47, 59)L)# EBCDIC,)# 12, 14, 19, 32, 49)L #EDIT PHASE, 4, 14, 55)L #EDITOR,)H6 35, 14, 13, 20, 56, 57)HL# "EDITOR",)#H6 14, 20, 41, 47, 51, 58)HLH6 #ESCAPE CHARACTER ") ", 4, 5, 12, 22, 24, 31, 44, 59)HL)LLM #F)ML)# "FOOTER ON LINE",)#H6 14, 40, 44, 47, 55, 58, 60)HL #FORTRAN, 2, 53, 54)L #FREE-FORM, 1, 4, 5, 9)L)# FT01F001, 50, 53, 54)L FT02F001, 50, 53)L FT03F001, 50, 53)L FT04F001, 50, 53)L FT05F001, 50, 54)L FT06F001, 50, 53, 54)L FT07F001, 50, 53)L FT08F001, 50, 54)L)C)LLM #G)HL)# "GO",)#H6 15, 4, 6, 9, 35, 36, 40, 44, 47, 55, 57, 58, 59)HL)LLM #H)HL #CHANGING INDENT,)H6 SEE #COMMAND #OPERAND #H)HL #HYPHEN, 45)L #HYPHENATION, 2, 15, 58)L)LLM #I)ML)# "\$INSERT",)H6# 37, 36, 38-41, 47, 59, 60)HL #IMMEDIATE INDENT,)H6 SEE #COMMAND #OPERAND #I)HL)#H6 "INDENTATION OF THE COLUMN", 15, 6, 25, 47)HL# #INDEX, 1, 14, 41)L #I/#O BUFFERS, 50, 53)L)LLM #J)ML)#H6 "\$JOIN", 39, 38, 47, 51, 52, 60)HL#)# "JUSTIFICATION",)# 15, 45, 57)L)LLM #K)ML #KEEP, SEE #COMMAND #OPERAND #K)L #KEYPUNCH,)H6 19, 22, 24, 28, 31, 33, 34, 47, 56)HL)LLM #L)ML)# "\$LIST",)# 41, 47)L)#H6 "\$LOCATE", 41, 8, 14, 42, 47, 51)H6#L)#H6 "LEFT TOP POSITION FOR PAGE NUMBER", 15, 46)HL#)C)#H6 "LINES PER PAGE", 16, 21, 47)HL#)#H6 "LIST THE INPUT", 16, 35, 46, 51)HL#)LLM #M)HL)#H6 "\$MERGE", 39, 38, 47, 51, 52, 60)HL# #MEMORY, 50, 53)L #MODE, SEE #AS-IS)H6 TEXT MODE, #NORMAL TEXT MODE, #CONTROL CARD MODE, #OUTPUT MODE, AND #EDITOR)HL #MULTI-PUNCHING, 31, 33)L)LLM #N)HL)# "\$NO DOCUMENT",)H6 40, 35, 47)HL "NONTRIVIAL BLANK",)H6 16, 33, 47)HL# #NON-TRIVIAL BLANK,)H6 33, 16-18, 42, 45)HL)#H6 "NO CAPITALIZATION AUTOMATICALLY", 16, 11, 46)HL#)#H6 "NO JUSTIFICATION", 16, 15, 18, 23, 47)HLH6 #NORMAL TEXT MODE, 4, 5, 7, 15)HL#H6 "NULL CHARACTER SWITCH", 17, 27, 33, 47)HL#)LLM #O)HL)#H6 "\$OVERRIDE", 40, 47)HL#)# "\$OMIT LISTING", 41, 47)L #OBJECT DECK, 53, 54)L #OPERATOR, 50)L #O#S/360, 2, 50, 53)L)# "OUTPUT MEDIUM IS TAPE",)H6 12, 46, 52)HL# #OUTPUT MODE, 2)L #OVERLAY, 53)HLH6 #OVERPRINTING, SEE)# "BACKSPACE CHARACTER")# AND)# "DARK PRINTING")HL#)C)LLM #P)ML)# "\$PUNCH", 40, 46, 47)L)#H6 "PAGE NUMBER STARTING AT", 17, 46)HL# #PARAGRAPH,)H6 3, 5, 11, 16, 17, 19, 26, 29, 30, SEE ALSO #COMMAND #OPERAND #P)HL)#H6 "PARAGRAPH INDENT", 17, 26, 47)HL# #PARENTHESES,)H6 SEE #ESCAPE #CHARACTER)HL #PERFORMANCE, 50)L #PERIOD, 14, SEE ALSO #DOTS)L #PHASE,)H6 SEE #EDIT #PHASE AND #DOCUMENT #PHASE)HL #PLUS SIGN, 41, 42)L #PRINTER, 19)L)#H6 "PRINT OUTPUT TAPE", 17, 46, 52)HL# #PRINTER TRAIN,)H6 SEE)# "SPECIAL PRINTER TRAIN")# AND #TEN #PRINT #TRAIN)HL#H6 "PUNCH THE INPUT", 18, 35, 51)HL#)LLM #R)HL #RECORD LENGTH, 12, 54)L)# "REPEAT TITLE", 18, 20, 47)L #REREAD, 57)L #RESTRICTION, 50)L)#H6 "RIGHT TOP POSITION FOR PAGE NUMBER", 18, 46)HL#)LLM #S)ML #SEARCH, 41)L #SEARCH ARGUMENT, 41, 42)L #SEGMENTS, 14, 59)L)#H6 "SENTENCES SEPARATED BY", 11, 18, 47)HL#)#H6 "SEPARATION LINES BETWEEN PARAGRAPHS", 19, 26)HL# #SEQUENTIAL, 50)L #SETUP, 53)L)C)#H6 "SIDE BY SIDE COPIES", 19, 21, 47)HL#)#H6 "SPACING OF TEXT LINES", 19, 26, 44, 47)HL# #SPECIAL CHARACTERS,)H6 1, 2, 9, 14, 16, 31)HL)H6# "SPECIAL KEYPUNC

H", 19, 14, 31, 47, 56)H=L #SPECIAL #OPERANDS,)H6 31, 4, 6, 8, 24, 34, 49)HL
)I7 "#", 31, 6, 19, 33, 34)L "!", 31-34)IL)# "SPECIAL PRINTER TRAIN",)H6 19,
 12, 48, 57)H)L #SPELLING, 14)L #STACKED,)H6 7, 25, 44)HL)#H6 "START TEXT
 ON",)# SEE)# "TEXT STARTS ON")HL)#H6 "STOP PRINTING TITLE", 20, 47, 57)#H
 L #STRING,)H6 24, 31, 35, 41, 42, 58)HL #SUBSCRIPTS, 2)L #SUPERSSCRIPTS,)H6 1
 9, 28, 31)HL)H6 #SYSTEM #INPUT #DATASET, 2, 12, 14, 20, 36, 37, 50, 52, 57, 58
)HL)H6 #SYSTEM #OUTPUT #DATASET, 2, 9, 13, 17, 42, 50, 52, 57)HL)H6 #SYSTEM
 #PUNCH #DATASET, 2, 18, 40, 50, 52)HL)LLM #T)ML)#H6 "TABS ARE SET AT", 20, 2
 3, 48)H=L)H6 #TABULATION, SEE #COMMAND #OPERANDS "#D" AND "#T" AND)# "TABS AR
 E SET AT")#HL)# "TAPE INPUT DATASET", 20,)H6 35, 36, 41, 47, 51, 58)#HL #TAP
 E LABELS, 50, 54)L)#H6 "TEXT STARTS ON", 20, 21, 23)#HL)#H6 "TITLE STARTS ON
 ", 20, 23, 40, 44, 46, 47, 55, 58, 60)#HLH6 #T#N #PRINT #TRAIN, 2, 32, 49)HL #
 TRANSLATION, 19, 31)L)C)LLM #U)ML)H6 #UNDERLINING, 17, 21, 27, SEE ALSO #CO
 MMAND #OPERAND "#U")HL)LLM #W)ML #WIDOWS,)H6 SEE #COMMAND #OPERAND "#W")HL
)#H6 "WIDTH OF COLUMNS", 21, 48)H=L)LLL)#H6 "026 KEYPUNCH", 22, 12, 38, 48)H
 #L)#H6 "029 KEYPUNCH", 22, 12, 38, 48)H=L)E

```

C*****01001000
C01002000
C    FORMAT RELEASE 501003000
C01004000
C    VERSION OF JUNE 24, 197101005000
C01006000
C*****01007000
C    IMPLICIT INTEGER*4 (A - Z)01008000
C    INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS,01009000
C    * PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR01010000
C    INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN01011000
C    COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE,01012000
C    * REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT,01013000
C    * CIINC,DICT,NEXT,HIT,01014000
C    * ARRAY1(3),01015000
C    * LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC,01016000
C    * HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR01017000
C    COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,MYPAGE,RIVER,HYPTRX,WPT,SUND,TWO01018000
C    * ,TWOUP,I,CPSW,ISPT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTX,LWI,N,01019000
C    * LINSIZ,NSYM,SPACNG,XTLINE,K,LSTBL,AUTO,PFLN,NOGO,NAME,SWK028,IIU,01020000
C    * CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTOTB,SWX,01021000
C    * JUNK,ID,PDUM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN,01022000
C    * LN2,REPTTL,IC,CWIDTH,USEWS1,LTITLE,CU(8),TAB2(7),INDP(4),01023000
C    * INDARR(8),FLN,ICINC,PARA,TPLN,USTART,FCM,SWWPT,ENDL,EWX,PIVOT,01024000
C    * INDEX,TABSEQ,CHAR,ENDF,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX,01025000
C    * WANT,GAPS,WANTIN,WSEPD,L,LSIDE,RSIDE,SWEW(68),CARD(40),01026000
C    * PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99)01027000
C*****01028000
C*****01029000
C    EXEC ROUTINE01030000
C*****01031000
C*****01032000
100    CALL READY01033000
      IF (NODOC.EQ. 0) GO TO 10101034000
103    CALL ENDJOB01035000
101    CALL VRDR01036000
      IF (NODOC) 100,105,10301037000
105    IF (K.GT. 0) GO TO 10201038000
      CALL EDITOR01039000
      IF (NODOC) 101,101,10001040000
102    CALL DRDR01041000
      IF (INDEX.EQ. 6) GO TO 10101042000
      GO TO 10001043000
      END01044000
C01045000
C01046000
C    SUBROUTINE ERR (/POS/,/CODE/)01047000
C    IMPLICIT INTEGER*4 (A - Z)01048000
C    DIMENSION CARRAY(3)01049000
C    COMMON IOUTPG,COL,LN,ERRCNT01050000
C    EQUIVALENCE (CARRAY(1),IOUTPG)01051000
C    WRITE (1,1000) CODE,CARRAY,POS01052000
1000    FORMAT ('E',76X,5A4)01053000
      ERRCNT = ERRCNT + 101054000
      RETURN01055000
      END01056000
C01057000
C01058000
C    BLOCK DATA01059000

```

```

IMPLICIT INTEGER*4 (A - Z)
INTEGER*2 LIST, COPIES, SPCHAR, BLANK, NUM, SCWORD, HYPHEN, LOWCAS,
* PERIOD, KEY028, LOCATE, CARDIC, OMLIST, DOLLAR
INTEGER*2 OVRDE, F1, F2, CWORD, TITLEX, BUFPT
COMMON /A/ POSN, IREAD, IWRITE, CCGCNT, PUNCH, NODOC, MERGE,
* REMNNT, INSWRD, FINISH, DELETE, COVEA, INVALID, BFOUND, CICNT1, CICNT,
* CIINC, DICT, NEXT, HIT,
* OVRDE, F1, F2, CWORD, TITLEX, BUFPT,
* LIST, COPIES, SPCHAR(42), BLANK, NUM(10), SCWORD, CARDIC,
* HYPHEN, LOWCAS, PERIOD, KEY028, LOCATE, OMLIST, DOLLAR
DATA SPCHAR, BLANK, NUM, HYPHEN, SCWORD,
C.... Z0140
* LOWCAS, CARDIC, POSN, IREAD, LIST, COPIES, IWRITE, CCGCNT
C.... 0 5 0 1 6 0
* , PUNCH, NODOC, PERIOD, REMNNT, INSWRD, FINISH, DELETE, COVEA,
C.... 0 0 0 10000 0 0 10
* INVALID, BFOUND, KEY028, CICNT, CIINC, MERGE, OVRDE, F1, F2,
C.... 0 10 0 0 0 0 1 80
* CWORD, TITLEX, BUFPT, DICT
C.... 0 0 1 0
* /Z8D40, Z9D40, Z8E40, ZA040, Z8B40, Z9B40, ZAD40, ZBD40, Z8C40, ZAE40,
* Z9E40, ZBE40, ZAB40, ZBB40, ZAC40, ZBC40, Z8F40, ZBF40, Z9C40, Z9F40,
* ZAF40, ZA140, Z5040, Z4F40, Z5F40, Z4C40, Z7E40, Z6E40, Z4E40, Z4D40,
* Z5D40, Z7F40, Z7D40, Z4A40, Z7B40, Z6C40, Z7C40, Z6D40, Z5E40, Z7A40,
* Z6F40, Z5A40,
* ' ', 'A', 'E', 'F', 'L', 'V', 'Z', '0', '1', '5', '9',
* '- ', Z0140, Z4000, ' ', '0,5,0,1,6,0,0,0, ' ', '0,10000,0,0,10,0,
* 10, ZE040, 0,0,0,0,1,80,0,0,1,0/
DATA CICNT1/1/, NEXT/677/, LOCATE/0/, HIT/0/
DATA OMLIST/0/, DOLLAR/'$ '/
END

C
C
SUBROUTINE READY
IMPLICIT INTEGER*4 (A - Z)
DIMENSION LINE3(5), CARD3(4), INPUTI(20)
INTEGER*2 CODES(25), SRTTXD(25), SRTTXT(26), TEXT(410), E, A, RECORD
INTEGER*2 LIST, COPIES, SPCHAR, BLANK, NUM, SCWORD, HYPHEN, LOWCAS,
* PERIOD, KEY028, LOCATE, CARDIC, OMLIST, DOLLAR
INTEGER*2 MASK1, EXCNT, CPAREN, EOSCHR, SAVMSK
INTEGER*2 PAGDUM, SRT, COLBEG, CHRFIN
COMMON IOUTPG, COL, LN, ERRCNT
COMMON /A/ POSN, IREAD, IWRITE, CCGCNT, PUNCH, NODOC, MERGE,
* REMNNT, INSWRD, FINISH, DELETE, COVEA, INVALID, BFOUND, CICNT1, CICNT,
* CIINC, DICT, NEXT, HIT,
* ARRAY1(3),
* LIST, COPIES, SPCHAR(42), BLANK, NUM(10), SCWORD, CARDIC,
* HYPHEN, LOWCAS, PERIOD, KEY028, LOCATE, OMLIST, DOLLAR
COMMON /B/ FIELD1, FIELD2, FIELD3, SPOP, CP, LB, PER, KEEPSV(2),
* UPPER, UP1, CAP, FIRST, MASK1, EXCNT, CPAREN, EOSCHR, SAVMSK
COMMON /C/ ALT, BLNKLN, ITEXT, LINPAG, MYPAGE, RIVER, HYPTRX, WPT, SUND, TW01110000
* , TW0UP, I, CPSW, ISPT, ASIS, CCHAR, CHARCO, NEWH, SAVCCC, KEEP, WPTX, LWI, N, 01111000
* LINSIZ, NSYM, SPACNG, TXTLNE, K, LSTBL, AUTO, FPLN, NOGO, NAME, SWK028, IJU, 01112000
* CCCNT, COLFAG, IVALUE, LINSZ, PAGENO, START, UNDERL, J, CONST, AUTOTB, SWK, 01113000
* JUNK, ID, PDUM, REM, IER700, END, CENTER, CSEP, INDENT, TEXEND, TLLN, CLEAN, 01114000
* LN2, REPTTL, IC, CWIDTH, USEWS1, LTITLE, CU(8), TAB2(7), INDP(4), 01115000
* INDARR(8), PLN, ICINC, PARA, TPLN, USTART, FCM, SWWPT, ENDL, EWY, PIVOT, 01116000
* INDEX, TABSEQ, CHAR, ENDF, LINEX(67), WORDS, LNTW, CS, ENDSAV, ID1, LINEW, 01117000
* WANT, GAPS, WANTIN, WSEPD, LSIDE, RSIDE, SWEW(68), CARD(40), 01118000
* PAGDUM(7788), SRT(99), COLBEG(8), CHRFIN(99) 01119000

```



```

DATA CODES,SRTEXT                                01120000
*/212,218,219,237,249,267,269,289,300,304,922,327,513,700,800,802, 01121000
* 804,805,806,807,847,814,857,220,997,          01122000
* 1,16,29,53,67,92,101,121,135,144,166,176,194,218,231,243,257,293,01123000
* 307,319,334,366,378,384,395,411/             01124000
INTEGER*2 TEXT1(100), TEXT2(100), TEXT3(100), TEXT4(10) 01125000
EQUIVALENCE (TEXT(101),TEXT1(1)), (TEXT(201),TEXT2(1)), 01126000
* (TEXT(301),TEXT3(1)), (TEXT(401),TEXT4(1))      01127000
DATA TEXT /'CONTROL CARD OPERAND IN ERROR UNRECOGNIZED CONTROL CAR01128000
*D NUMBER OF PRINT POSITIONS REQUIRED NOT AVAILABLE TABS NOT IN ASC01129000
*NDING ORDER CONTROL CARD NOT FIRST, OR ON DATASET OTHER THAN STAB 01130000
*IMPROPERLY SET'/                                01131000
DATA TEXT1                                       01132000
* / 'IMPROPER STARTING LINE FOR DOCUMENT TEXTIMPROPER CONO1133000
*TROL CARD ORDER INDENTS TOO LARGE CHARACTER STRING LENGTH EXCEEDS 01134000
*COLUMN WIDTH NO TEXT AFTER TAB(S) TAB COMMAND OPERAND IMPROPERLY USE01135000
*D NUMBER OF UNDE'/                             01136000
DATA TEXT2                                       01137000
* / 'RLINE SEGMENTS ON PAGE EXCEEDS 99 UNDEFINED COMMAND01138000
*D OPERAND UNEXPECTED END OF INPUT NEW MASTER ALREADY FINISHED EDIT01139000
* FAILED BECAUSE OF ABOVE ERROR(S) OR BECAUSE NEW MASTER NOT FINISH01140000
*EDREFERENCED WORD '/                          01141000
DATA TEXT3                                       01142000
* / 'NOT LOCATED INPUT/OPERAND MODE ERROR END OF $DELETE01143000
*TE FIELD NOT FOUND$INSERT/DELETE/DUPLICATE AND $MERGE/JOIN NOT ALLO1144000
*OWED IN SAME RUNNON-TEXT MODE NOT ENDED NOT ALLOWED TITLE/FOOTER TO1145000
*OO LONG TOO MANY BAC'/                        01146000
DATA TEXT4                                       01147000
* /'KSPACES ON ONE PAGE '/                    01148000
EQUIVALENCE (LINE3(1),LINEX(1)), (E,NUM(2)), (CARD3(1),LINEX(41)), 01149000
* (A,NUM(1)), (SRTEXT(1),SRTEXT(2)), (INPUT1(1),CARD(2)) 01150000
COMMON /EHRMAN/ DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF, 01151000
* BACKWD, BAKPOS, BACHAR, BACKST, BACKND, NULLSW, 01152000
* CCWIDT, NOTRIV, MASK2, EDCCWI, UNDRSW, EDCOL1 01153000
INTEGER*2 DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF, NULLSW, 01154000
* BACKWD(68), BAKPOS(100), BACKST(8), BACKND(8), CCWIDT, NOTRIV, 01155000
* MASK2, EDCCWI, UNDRSW, EDCOL1 01156000
LOGICAL*1 BACHAR(100)                          01157000
INTEGER*4 BAKZRO(119)                          01158000
EQUIVALENCE (BACKCH,BAKZRO(1))                 01159000
IF (DICT .LT. 0) STOP 1                        01160000
IF (CCGCNT .EQ. 0) GO TO 733                   01161000
REWIND 1                                        01162000
I = 1                                           01163000
730 WRITE (6,1001)                             01164000
1001 FORMAT ('1',51X,30HFORMAT RELEASE 5 CONTROL CARDS///15X,'GROUP PA01165000
*GE COLUMN LINE NO. ')                       01166000
JUNK = 0                                       01167000
J = 4                                          01168000
DO 725 I = 1, CCCNT                          01169000
IF (J .GT. 55) GO TO 730                     01170000
716 READ (1,1006) RECORD,CARD3,INPUT1       01171000
1006 FORMAT (A1,24A4)                        01172000
IF (RECORD .NE. A) GO TO 716                 01173000
IF (JUNK .EQ. CARD3(1)) GO TO 729            01174000
JUNK = CARD3(1)                              01175000
WRITE (6,1002) CARD3,I,INPUT1               01176000
1002 FORMAT (/11X,4I7,I6,7X,20A4)           01177000
J = J + 2                                    01178000
GO TO 725                                    01179000

```

[illegible]

LINSIZ = 132	01240000
LN = 0	01241000
LN2 = 2	01242000
LSIDE = 132	01243000
LTITLE = 0	01244000
SAVMSK = MASK1	01245000
C.....FOR USE IN NUMBERING WORDS IN PROPER CASE IN LISTER ROUTINE	01246000
MASK1 = LOWCAS	01247000
MASK2 = LOWCAS	01248000
MYPAGE = 1	01249000
NEWH = 0	01250000
NOTRIV = 0	01251000
NSYM = 3	01252000
NULLSW = 1	01253000
PAGENO = 1	01254000
PARA = 0	01255000
REPTTL = 0	01256000
RIVER = 10	01257000
RSIDE = 0	01258000
SPACNG = 1	01259000
START = 0	01260000
SWK028 = 0	01261000
TEXEND = 7788	01262000
TPLN = 2	01263000
TWOUP = 1	01264000
TXTLNE = 3	01265000
UNDERL = 0	01266000
UNDRSW = 1	01267000
USTART = 0	01268000
WPTX = 0	01269000
REWIND 1	01270000
DO 1051 I = 1, 119	01271000
C.....THIS LOOP ZEROS ALL THE BACKSPACE FLAGS AND ARRAYS	01272000
1051 BAKZRO(I) = 0	01273000
DO 102 I = 1, TEXEND	01274000
102 PAGDUM(I) = BLANK	01275000
DO 105 I = 1, 27	01276000
C.....THIS LOOP ZEROES CU(8), TAB2(7), INDP(4), AND INDARR(8) ARRAYS	01277000
105 CU(I) = 0	01278000
RETURN	01279000
END	01280000
C	01281000
C	01282000
SUBROUTINE COND (INPUT, LINE)	01283000
IMPLICIT INTEGER*4 (A-Z)	01284000
LOGICAL*1 INPUT(80), CRDC	01285000
INTEGER*2 LINE(133), F, Z, A, ONE, ZERO, FIVE, NINE, QUOTES, CENTS, EXPT,	01286000
* QMARK	01287000
INTEGER*2 LIST, COPIES, SPCHAR, BLANK, NUM, SCWORD, HYPHEN, LOWCAS,	01288000
* PERIOD, KEY028, LOCATE, CARDIC, OMLIST, DOLLAR	01289000
INTEGER*2 MASK1, EXCNT, CPAREN, EOSCHR, SAVMSK	01290000
COMMON /A/ POSN, IREAD, IWRITE, CCGCNT, PUNCH, NODOC, MERGE,	01291000
* REMNNT, INSWRD, FINISH, DELETE, COVEA, INVALID, BFOUND, CICNT1, CICNT,	01292000
* CIINC, DICT, NEXT, HIT,	01293000
* ARRAY1(3),	01294000
* LIST, COPIES, SPCHAR(42), BLANK, NUM(10), SCWORD, CARDIC,	01295000
* HYPHEN, LOWCAS, PERIOD, KEY028, LOCATE, OMLIST, DOLLAR	01296000
COMMON /B/ FIELD1, FIELD2, FIELD3, SPOP, CP, LB, PER, KEEPSV(2),	01297000
* UPPER, UP1, CAP, FIRST, MASK1, EXCNT, CPAREN, EOSCHR, SAVMSK	01298000
COMMON /EHRMAN/ DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF,	01299000

```

*          BACKWD, BAKPOS, BACHAR, BACKST, BACKND, NULLSW, 01300000
* CCWIDT, NOTRIV, MASK2, EDCCWI, UNDRSW, EDCOL1 01301000
  INTEGER*2 DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF, NULLSW, 01302000
* BACKWD(68), BAKPOS(100), BACKST(8), BACKND(8), CCWIDT, NOTRIV, 01303000
* MASK2, EDCCWI, UNDRSW, EDCOL1 01304000
  LOGICAL*1 BACHAR(100) 01305000
  EQUIVALENCE (F,NUM(3)), (Z,NUM(6)), (A,NUM(1)), (ONE,NUM(8)), (ZERO, 01306000
* NUM(7)), (QUOTES,SPCHAR(32)), (FIVE,NUM(9)), (NINE,NUM(10)) 01307000
*, (CENTS,SPCHAR(34)), (EXPT,SPCHAR(42)), (QMARK,SPCHAR(41)) 01308000
*, (CRDC,CARDIC) 01309000
  FIELD3 = 0 01310000
  IF (SPOP .EQ. 0) ASSIGN 917 TO SPOP 01311000
  DO 907 I = FIELD1, FIELD2 01312000
  CRDC = INPUT(I) 01313000
  IF (CARDIC .EQ. NOTRIV) CARDIC = REI028 01314000
C.....ALLOW NONTRIVIAL BLANKS IN TITLES AND FOOTERS 01315000
  GO TO SPOP, (917,900,901,911) 01316000
C.....SP. OP. IS NONE CTS EXPT CO 01317000
900  ASSIGN 917 TO SPOP 01318000
  IF (CARDIC .GE. A .AND. CARDIC .LE. NINE) GO TO 925 01319000
C.....THE ABOVE IS CODE DEPENDENT 01320000
  EXCNT = CENTS 01321000
  GO TO 930 01322000
925  IF (CARDIC .GT. Z) CARDIC = CARDIC - MASK2 01323000
  GO TO 927 01324000
901  IF (EXCNT .NE. BLANK) GO TO 913 01325000
  EXCNT = CARDIC 01326000
  IF (CARDIC .GE. ONE .AND. CARDIC .LE. FIVE) GO TO 907 01327000
  EXCNT = EXPT 01328000
  ASSIGN 917 TO SPOP 01329000
  PER = 10 01330000
  GO TO 930 01331000
913  ASSIGN 917 TO SPOP 01332000
  IF (CARDIC .LT. ZERO .OR. CARDIC .GT. NINE) GO TO 915 01333000
  N = (10 * EXCNT + CARDIC + 42048) / 256 01334000
C.....THE ABOVE IS CODE DEPENDENT 01335000
  IF (N .LE. 42) GO TO 919 01336000
915  FIELD3 = FIELD3 + 1 01337000
  LINE(FIELD3) = EXPT 01338000
930  FIELD3 = FIELD3 + 1 01339000
  LINE(FIELD3) = EXCNT 01340000
  EXCNT = BLANK 01341000
  GO TO 914 01342000
919  CARDIC = SPCHAR(N) 01343000
  EXCNT = BLANK 01344000
  IF (N .GT. 40) PER = 10 01345000
  GO TO 927 01346000
904  N = 32000 01347000
  LB = 10 01348000
922  UP1 = 0 01349000
  IF (PER + CAP * FIRST .GT. 10) UP1 = 10 01350000
  IF (PER .EQ. 0) IF (N) 907,928,908 01351000
  GO TO 908 01352000
906  PER = 10 01353000
  LB = 0 01354000
927  UP1 = 0 01355000
  GO TO 908 01356000
911  IF (CARDIC .NE. BLANK) GO TO 910 01357000
  ASSIGN 917 TO SPOP 01358000
  N = CP - 1 01359000

```

	CP = 0	01360000
	GO TO 922	01361000
910	CP = CP - 2	01362000
	IF (CARDIC .NE. CENTS) GO TO 923	01363000
	UPPER = 10 - UPPER	01364000
	GO TO 907	01365000
923	IF (CARDIC .NE. F) GO TO 926	01366000
	FIRST = 20 - FIRST	01367000
	GO TO 907	01368000
926	IF (CP .GT. 10000) GO TO 908	01369000
	CP = 32000	01370000
928	FIELD3 = FIELD3 + 1	01371000
	LINE(FIELD3) = SCWORD	01372000
	GO TO 908	01373000
902	ASSIGN 900 TO SPOP	01374000
	GO TO 916	01375000
905	CP = 1	01376000
	ASSIGN 911 TO SPOP	01377000
	GO TO 907	01378000
903	ASSIGN 901 TO SPOP	01379000
916	PER = 0	01380000
	LB = 0	01381000
	GO TO 907	01382000
917	IF (CARDIC .LT. 0) GO TO 918	01383000
	C.....THE ABOVE IS CODE-DEPENDENT	01384000
914	IF (CARDIC .EQ. BLANK) GO TO 904	01385000
	IF (CARDIC .EQ. CPAREN .AND. LB .NE. 0) GO TO 905	01386000
	IF (CARDIC .EQ. CENTS) GO TO 902	01387000
	IF (CARDIC .EQ. EXPT) GO TO 903	01388000
	IF (CARDIC .EQ. PERIOD .OR. CARDIC .EQ. QMARK) GO TO 906	01389000
	IF (CARDIC .EQ. CPAREN .OR. CARDIC .EQ. QUOTES) GO TO 920	01390000
918	PER = 0	01391000
920	LB = 0	01392000
	IF (-UPPER .NE. UP1) GO TO 927	01393000
	IF (CARDIC .LE. Z .AND. CARDIC .NE. KEY028 .AND. CARDIC .GE. A)	01394000
	* CARDIC = CARDIC - MASK1	01395000
908	FIELD3 = FIELD3 + 1	01396000
	LINE(FIELD3) = CARDIC	01397000
907	CONTINUE	01398000
	RETURN	01399000
	END	01400000
C		01401000
C		01402000
	SUBROUTINE VRDR	01403000
	IMPLICIT INTEGER*4 (A - Z)	01404000
	DIMENSION LINE1(14),CARD1(54),CABRAY(3),PMT0(3),PMTI(2),PMTIN(22)	01405000
	*,INPUTI(20)	01406000
	LOGICAL*1 INPUT(80),CRDC	01407000
	INTEGER*2 VV(47),OVPT,LINE2(80),LINE(133),BUFCHR,L,E,TAB(14)	01408000
	*,MINE,ZERO,OVRC(7787),OVRDE,F1,F2,CWORD,TITLEX,BUFPT,INDARH(14)	01409000
	*,TAB1(13)	01410000
	INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS,	01411000
	* PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR	01412000
	INTEGER*2 MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK	01413000
	INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN	01414000
	COMMON IOUTPG,COL,LN,ERRCNT	01415000
	COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE,	01416000
	* REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT,	01417000
	* CIINC,DICT,NEXT,HIT,	01418000
	* OVRDE,F1,F2,CWORD,TITLEX,BUFPT,	01419000

237	CONTINUE	01480000
245	WRITE (1,3001) CCGCNT,CARRAY,INPUTI	01481000
3001	FORMAT ('A',24A4)	01482000
	CCCNT = CCCNT + 1	01483000
	DO 281 I = 41, 54	01484000
281	CARD1(I) = 0	01485000
	K = 0	01486000
	NAME = 0	01487000
	DO 282 I = 1, CCWIDT	01488000
	CRDC = INPUT(I)	01489000
	IF (CARDIC .EQ. BLANK) GO TO 282	01490000
	K = K + 1	01491000
	NAME = 16 * NAME + (CARDIC - 3648) / 256	01492000
C.....	THE ABOVE IS CODE DEPENDENT	01493000
	IF (K .EQ. 3) GO TO 285	01494000
282	CONTINUE	01495000
285	K = 41	01496000
280	ASSIGN 284 TO HIT1	01497000
284	IF (I .GE. CCWIDT) GO TO 289	01498000
	I = I + 1	01499000
	CRDC = INPUT(I)	01500000
	IF (CARDIC .LE. NINE .AND. CARDIC .GE. ZERO) GO TO 288	01501000
	GO TO HIT1, (284,291)	01502000
291	IF (K .EQ. 54) GO TO 289	01503000
	K = K + 1	01504000
	GO TO 280	01505000
288	CARD1(K) = 10 * CARD1(K) + (CARDIC + 4032) / 256	01506000
C.....	THE ABOVE IS CODE DEPENDENT	01507000
	ASSIGN 291 TO HIT1	01508000
	GO TO 284	01509000
289	IF (NAME .NE. VV(15) .AND. NAME .NE. VV(12) .AND. NAME .NE. VV(38)	01510000
C.....	GO TITLE FOOTER	01511000
	*) GO TO 250	01512000
	IF (OVRDE .EQ. 0) GO TO 251	01513000
	OVRDE = -OVRDE	01514000
	BACKSPACE IREAD	01515000
	CICNT = CICNT - CIINC	01516000
	BACKSPACE 1	01517000
	CCCNT = CCCNT - 1	01518000
	GO TO 230	01519000
250	IF (TITLE .NE. 0) CALL ERR (CCCNT,289)	01520000
251	TITLE = 0	01521000
	DO 241 K = 1, 47	01522000
	IF (NAME .EQ. VV(K)) GO TO (201,202,203,204,205,206,207,208,	01523000
	* 209,210,211,212,213,215,219,216,217,265,226,227,228,229,	01524000
	* 271,273,274,275,276,265,214,253,243,258,249,257,253,252,298,	01525000
	* 283,256,254,	01526000
	* 998,999,997,996,995,994,993	01527000
	*) , K	01528000
241	CONTINUE	01529000
	CALL ERR (CCCNT,218)	01530000
	MASK1 = 0	01531000
	IF (IREAD .NE. 5) BACKSPACE IREAD	01532000
	GO TO 219	01533000
C.....	WIDTH OF COLUMNS IS XX POSITIONS	01534000
201	IF (CARD5 .LT. 1) GO TO 234	01535000
	CHARCO = CARD5	01536000
	GO TO 235	01537000
C.....	COLUMNS PER PAGE ARE XX	01538000
202	IF (CARD5 .LT. 1 .OR. CARD5 .GT. 8) GO TO 234	01539000

COLPAG = CARD5	01540000
GO TO 235	01541000
C.....LINES PER PAGE = XX	01542000
203 IF (CARD5 .LT. 5 .OR. CARD5 .GT. 1000) GO TO 234	01543000
LINPAG = CARD5	01544000
TLLN = CARD5	01545000
LINSIZ = 7788 / LINPAG	01546000
IF (LINSIZ .GT. 132) LINSIZ = 132	01547000
GO TO 235	01548000
C.....PAGE NUMBER = XX	01549000
204 PAGENO = CARD5	01550000
GO TO 235	01551000
C.....BETWEEN COLUMNS LEAVE XX POSITIONS	01552000
205 CSEP = CARD5	01553000
GO TO 235	01554000
234 CALL ERR (CCCNT,212)	01555000
GO TO 235	01556000
C.....PARAGRAPH INDENT = XX	01557000
206 INDENT = CARD5	01558000
GO TO 235	01559000
C.....SEPARATION BETWEEN PARAGRAPHS = XX LINES	01560000
207 BLNKLN = CARD5	01561000
GO TO 235	01562000
C.....JUSTIFICATION	01563000
208 AUTO = 1	01564000
GO TO 235	01565000
C.....NO JUSTIFICATION	01566000
209 AUTO = 0	01567000
GO TO 235	01568000
C.....REPEAT TITLE ON EVERY PAGE	01569000
210 REPTTL = 10	01570000
GO TO 235	01571000
C.....STOP REPEATING TITLE	01572000
211 REPTIL = 0	01573000
GO TO 235	01574000
C.....SPACING = XX	01575000
213 IF (CARD5 .LT. 1) GO TO 234	01576000
SPACNG = CARD5	01577000
GO TO 235	01578000
C.....CARD FIELD IS XX THRU YY, OR THRU YY	01579000
215 IF (CARD6 .NE. 0) GO TO 240	01580000
CARD6 = CARD5	01581000
CARD5 = 1	01582000
240 IF (CARD5 .LT. 1 .OR. CARD6 .GT. 80 .OR. (CARD5 + 2) .GT. CARD6)	01583000
* GO TO 234	01584000
FIELD1 = CARD5	01585000
FIELD2 = CARD6	01586000
GO TO 235	01587000
C.....TABS ARE SET AT XX1,...,XX14	01588000
216 DO 220 I = 1, 14	01589000
TAB(I) = LINE1(I)	01590000
220 CONTINUE	01591000
GO TO 235	01592000
C.....SIDE BY SIDE	01593000
217 TWOUP = 2	01594000
GO TO 235	01595000
C.....CENTER TEXT ON LINE X	01596000
256 START = 0	01597000
GO TO 266	01598000
C.....TEXT STARTS ON LINE XX, POSITION YY	01599000

C.....START TEXT ON LINE XX, POSITION YY	01600000
265 IF (CARD6 .NE. 0) START = CARD6	01601000
266 IF (CARD5 .NE. 0) FLN = CARD5	01602000
GO TO 235	01603000
C.....LEFT TOP POSITION FOR PAGE NUMBER	01604000
227 ALT = 0	01605000
GO TO 235	01606000
C.....RIGHT TOP POSITION FOR PAGE NUMBER	01607000
228 ALT = 1	01608000
GO TO 235	01609000
C.....CYCLE PAGE NUMBER	01610000
229 IVALUE = 1	01611000
GO TO 235	01612000
C.....026 KEYPUNCH	01613000
271 CPAREN = SPCHAR(26)	01614000
GO TO 235	01615000
C.....029 KEYPUNCH	01616000
273 CPAREN = SPCHAR(31)	01617000
GO TO 235	01618000
C.....CAPITALIZE AUTOMATICALLY	01619000
274 CAP = 10	01620000
GO TO 235	01621000
C.....NO CAPITALIZATION AUTOMATICALLY	01622000
275 CAP = 0	01623000
GO TO 235	01624000
C.....SPECIAL PRINT TRAIN	01625000
C.....SPECIAL KEYPUNCH	01626000
276 MASK1 = 0	01627000
MASK2 = 0	01628000
C.....CHECK FOR SUPERSCRIPTS DESIRED FROM SPECIAL KEYPUNCH	01629000
IF (CARD5 .EQ. 2741) MASK2 = 16384	01630000
GO TO 235	01631000
C.....INDENT THE COLUMN (L1,R1),...., (L7,R7) PRINT POSITIONS	01632000
214 DO 272 I = 1, 14	01633000
272 INDARH(I) = LINE1(I)	01634000
GO TO 235	01635000
C.....DICTIONARY OF INPUT WORDS	01636000
254 DICT = 10	01637000
GO TO 255	01638000
C.....PUNCH	01639000
252 PUNCH = 10	01640000
C.....CREATE INPUT TAPE	01641000
C.....LIST	01642000
253 LIST = 10	01643000
255 CIINC = 1	01644000
IF (IREAD .EQ. 5) GO TO 244	01645000
IF (CICNT .EQ. 0) CICNT = 1	01646000
GO TO 235	01647000
244 CCGCNT = CCGCNT - 1	01648000
K = -10	01649000
F1 = FIELD1	01650000
F2 = FIELD2	01651000
C.....REQ'D BY FMTIV (EXEC) TO CALL EDITOR ROUTINE TO COPY SYSIN ONTO FT20	01652000
259 IREAD = 2	01653000
REWIND 2	01654000
FIELD1 = 1	01655000
FIELD2 = 80	01656000
GO TO 293	01657000
C.....COPIES = XX	01658000
243 COPIES = CARD5	01659000

GO TO 235	01660000
C.....PRINT MOUNTED OUTPUT TAPE	01661000
226 NODOC = 10	01662000
C.....OUTPUT IS ONTO TAPE	01663000
257 IWRITE = 8	01664000
GO TO 235	01665000
C.....TAPE INPUT	01666000
258 IF (IREAD - CCCNT - 4) 200,218,200	01667000
C.....EDITOR	01668000
249 IF ((IREAD - CCCNT) .EQ. 4) GO TO 236	01669000
C.....ABOVE IS EQUIVALENT TO: IF (CCCNT .EQ. 1 .AND. IREAD .EQ. 5) GO 236	01670000
200 CALL ERR (CCCNT,249)	01671000
GO TO 235	01672000
236 LIST = 10	01673000
218 CIINC = 1	01674000
K = 0	01675000
C.....REQUIRED BY FMTIV (EXEC) TO CALL EDITOR	01676000
GO TO 259	01677000
C.....SENTENCES SEPARATED BY 2 BLANKS MINIMUM	01678000
298 EOSCHR = 10	01679000
IF (CARD5 .EQ. 2) GO TO 235	01680000
EOSCHR = 0	01681000
GO TO 235	01682000
C.....DARK PRINT N TIMES	01683000
998 DARKER=CARD5	01684000
IF (CARD5 .GT. 3) DARKER = 3	01685000
GO TO 235	01686000
C.....DROP CHARACTER FOR 'D' COMMAND IS ...	01687000
999 IF (CARD5 .NE. 0) GO TO 9991	01688000
DROPCH = PERIOD	01689000
GO TO 235	01690000
9991 IF (CARD5 .GE. 10 .AND. CARD5 .LE. 51) GO TO 9992	01691000
DROPCH = 256*CARD5 + 64	01692000
C.....MAKE UP A FAKE EBCDIC CHARACTER FOR DROPPING ON 'D' COMMAND	01693000
C.....CODE DEPENDENT	01694000
GO TO 235	01695000
9992 DROPCH = SPCHAR(CARD5-9)	01696000
GO TO 235	01697000
C.....BACKSPACE CHARACTER IS XX	01698000
997 BACKCH = 0	01699000
IF (CARD5 .LT. 10 .OR. CARD5 .GT. 50) GO TO 235	01700000
IF (CARD5 .EQ. 43) GO TO 235	01701000
C.....NO CENTS SIGNS OR EXCLAMATION POINTS FOR BACKSPACING MARKER	01702000
BACKCH = SPCHAR(CARD5-9)	01703000
GO TO 235	01704000
C.....NULL SWITCH IS XX	01705000
996 NULLSW = 1	01706000
C.....NULLSW=1 IS NORMAL, =2 MEANS CENTER AND UNDERLINE 028'S	01707000
IF (CARD5 .EQ. 2) NULLSW = CARD5	01708000
GO TO 235	01709000
C.....CONTROL CARD ENDS IN COLUMN XX	01710000
995 CCWIDT = CARD5	01711000
IF (CARD5 .LE. 6 .OR. CARD5 .GT. 80) CCWIDT = 80	01712000
GO TO 235	01713000
C.....NONTRIVIAL BLANK IS REPRESENTED BY SPECIAL CHARACTER NN	01714000
994 NOTRIV = 0	01715000
IF (CARD5 .GE. 10 .AND. CARD5 .LE. 51) NOTRIV = SPCHAR(CARD5-9)	01716000
GO TO 235	01717000
C.....UNDERLINE SWITCH SET TO 0 (UNDERLINES EVERYTHING)	01718000
993 UNDRSW = 1	01719000

IF (CARD5 .EQ. 0) UNDRSW = 0	01720000
GO TO 235	01721000
C....FOOTER ON LINE XX, PRINT POSITION YY, PRECEDED BY ZZ BLANK LINES	01722000
283 J = LINPAG	01723000
GO TO 263	01724000
248 J = TEXEND	01725000
247 IF (J .EQ. 7788) GO TO 239	01726000
J = J + 1	01727000
PAGDUM(J) = BLANK	01728000
GO TO 247	01729000
239 FFLN = LINPAG	01730000
IF (CARD5 .NE. 0) FFLN = CARD5	01731000
JUNK = 2	01732000
C....2 LINES BETWEEN THE TEXT AND THE FOOTER IS THE DEFAULT	01733000
IF (CARD7 .NE. 0) JUNK = CARD7	01734000
TLLN = FFLN - JUNK - 1	01735000
C....TLLN IS LAST LINE AVAILABLE TO TEXT; TEXEND = TLLN * LINSIZ	01736000
TITLNK = FFLN	01737000
ASSIGN 235 TO DUMMY	01738000
C...."TITLE" ROUTINE	01739000
2001 SPOP = 0	01740000
CP = 0	01741000
EXCNT = BLANK	01742000
FIRST = 0	01743000
LB = 10	01744000
PER = 0	01745000
UPPER = 0	01746000
UP1 = 0	01747000
ASSIGN 2215 TO BR2	01748000
GO TO 2216	01749000
2215 K = 0	01750000
CPSW = 0	01751000
2239 XX = (TITLNK - 1) * LINSIZ	01752000
IF (TITLNK .LE. LINPAG) GO TO 2219	01753000
CALL ERR (CCGCNT,220)	01754000
GO TO 2294	01755000
2219 DO 2221 I = 1, LINSIZ	01756000
2221 PAGDUM(I + XX) = BLANK	01757000
LSTBL = 10	01758000
I = TITLE	01759000
2214 IF (CPSW .NE. 0) GO TO 2202	01760000
2201 CARDIC = LINE(IC)	01761000
J = 0	01762000
IF (CARDIC .EQ. BLANK) GO TO 2200	01763000
IF (CARDIC .EQ. SCWORD) GO TO 2205	01764000
2204 LSTBL = 0	01765000
IF (CARDIC .EQ. KEY028) GO TO 2202	01766000
K = 0	01767000
PAGDUM(I + XX) = CARDIC	01768000
IF (TWOUP .GT. 1) PAGDUM(I + XX + LINSZ) = CARDIC	01769000
2202 IC = IC + 1	01770000
ASSIGN 2217 TO BR2	01771000
2218 IF (IC .LE. FIELD3) GO TO 2206	01772000
2216 IC = 1	01773000
READ (IREAD,FMTIN,END=710)	01774000
CICNT = CICNT + CIINC	01775000
C.... CALL COND FOR FOOTER AND TITLE	01776000
CALL COND (INPUT,LINE)	01777000
GO TO 2218	01778000
2206 GO TO BR2, (2217,2215)	01779000

112

2217	IF (CPSW .EQ. 0) GO TO 2207	01780000
	CARDIC = LINE(IC)	01781000
	IF (CARDIC .EQ. BLANK) GO TO 2208	01782000
	IF (CARDIC .EQ. E) GO TO 2209	01783000
	CPSW = -10	01784000
	IF (CARDIC .EQ. L) GO TO 2210	01785000
	J = LN	01786000
	LN = TITLNK	01787000
	CALL ERR (I - TITLE + 1,700)	01788000
	LN = J	01789000
	GO TO 2202	01790000
2208	IF (CPSW .GT. 0) GO TO 2211	01791000
	CPSW = 0	01792000
2203	J = 10	01793000
	GO TO 2202	01794000
2211	IC = IC - 1	01795000
	CPSW = 0	01796000
	CARDIC = CPAREN	01797000
	GO TO 2204	01798000
2210	IF (K .EQ. 0) GO TO 2212	01799000
	K = 0	01800000
	GO TO 2202	01801000
2200	IF (LSTBL .NE. 0) GO TO 2203	01802000
	LSTBL = 10	01803000
	GO TO 2202	01804000
2205	CPSW = 10	01805000
	GO TO 2202	01806000
2207	IF (J .NE. 0) GO TO 2201	01807000
	I = I + 1	01808000
	IF (I .LE. LINSZ) GO TO 2214	01809000
	K = 10	01810000
2212	TITLNK = TITLNK + 1	01811000
	IF (RSIDE .LT. I - 1) RSIDE = I - 1	01812000
	GO TO 2239	01813000
2209	IF (K .EQ. 0) TITLNK = TITLNK + 1	01814000
2294	GO TO DUMMY, (235,221)	01815000
C....	TITLE STARTS ON LINE XX, POSITION YY	01816000
212	J = 2	01817000
263	TITLE = 10	01818000
	GO TO 295	01819000
296	J = (CARD5 - 1) * LINSIZ	01820000
	DO 260 I = 1, J	01821000
260	PAGDUM(I) = BLANK	01822000
	IF (CARD5 .NE. 0) TFLN = CARD5	01823000
	TITLNK = TFLN	01824000
	ASSIGN 221 TO DUMMY	01825000
	GO TO 2001	01826000
221	LTITLE = TITLNK - 1	01827000
	LN2 = TITLNK	01828000
	IF (LN2 .EQ. LINPAG) REPTTL = 0	01829000
	GO TO 235	01830000
C....GO		01831000
219	IF (NODOC .NE. 0) GO TO 293	01832000
	CP = 0	01833000
	CPSW = 0	01834000
	END = 1	01835000
	EXCNT = BLANK	01836000
	FIRST = 0	01837000
	IC = 80	01838000
	KEEP = -IABS(KEEP)	01839000

	PER = 0	01840000
	SPOP = 0	01841000
	UPPER = 0	01842000
	UP1 = CAP	01843000
	IF (COPIES + IWRITE .GT. 7) IWRITE = 8	01844000
	IF (PAGENO .EQ. 0) GO TO 277	01845000
	MYPAGE = PAGENO	01846000
	CONST = 1	01847000
	JUNK = PAGENO	01848000
	DO 278 ID1 = 1, PAGENO	01849000
	JUNK = JUNK / 10	01850000
	CONST = 10 * CONST	01851000
	IF (JUNK .EQ. 0) GO TO 279	01852000
278	CONTINUE	01853000
279	ID = ID1 - 1	01854000
277	IOUTPG = MYPAGE	01855000
295	LINSZ = LINSIZ / TWOUP	01856000
264	CWIDTH = START - 1 + (COLPAG * CHARCO) + CSEP * (COLPAG - 1)	01857000
	IF (START .NE. 0) GO TO 261	01858000
	START = LINSZ/2 - (CWIDTH + 1)/2 + 1	01859000
	GO TO 264	01860000
261	IF (CWIDTH .LE. LINSZ) GO TO 238	01861000
	CALL ERR (CCGCNT,219)	01862000
	START = LINSZ - CWIDTH + START	01863000
	IF (START .GT. 0) GO TO 238	01864000
	START = 1	01865000
	CSEP = 2	01866000
	CHARCO = (LINSZ - CSEP * (COLPAG - 1)) / COLPAG	01867000
238	IF (LSIDE .GT. START) LSIDE = START	01868000
	IF (TITLE .EQ. 0) GO TO 292	01869000
	IF (CARD5 .LE. LIMPAG .AND. CARD6 .LE. LINSZ) GO TO 262	01870000
	CALL ERR (CCCNT,212)	01871000
	CARD5 = J	01872000
	CARD6 = 0	01873000
262	TITLE = START	01874000
	IF (CARD6 .GT. 0) TITLE = CARD6	01875000
	IF (LSIDE .GT. TITLE) LSIDE = TITLE	01876000
	IF (J - 2) 248,296,248	01877000
292	SWEW(1) = FMTO(1)	01878000
	SWEW(2) = FMTO(2)	01879000
	SWEW(35) = FMTO(3)	01880000
	DO 225 I = 3, 34	01881000
225	SWEW(I) = BLBL	01882000
	IF (ITEXT .LT. LN2) GO TO 267	01883000
	JUNK = (LN2 - 1) * LINSIZ + 1	01884000
	J = ITEXT * LINSIZ	01885000
	DO 224 I = JUNK, J	01886000
224	PAGDUM(I) = BLANK	01887000
267	DO 231 I = 1, 14	01888000
	IF (TAB(I) .EQ. 0) GO TO 223	01889000
	IF (TAB(I) .LE. CHARCO) GO TO 232	01890000
	CALL ERR (CCGCNT,267)	01891000
233	TAB(I) = CHARCO	01892000
232	IF (I .EQ. 1 .OR. TAB(I) .GT. TAB1(I) .OR. TAB(I) .EQ. CHARCO) GO	01893000
	1 TO 231	01894000
	CALL ERR (CCGCNT,237)	01895000
	GO TO 233	01896000
231	CONTINUE	01897000
223	JUNK = START	01898000
	DO 268 I = 1, COLPAG	01899000

114

```

COLBEG(I) = JUNK                                01900000
JUNK = JUNK + CHARCO + CSEP                      01901000
268 CONTINUE                                    01902000
ISPOT = JUNK - CSEP - 1                          01903000
IF (RSIDE .LT. ISPOT) RSIDE = ISPOT              01904000
IF (TWOUP .GT. 1) RSIDE = LINSIZ                 01905000
IF (FLN .LE. TLLN) GO TO 269                     01906000
CALL ERR (CCGCNT,269)                            01907000
FLN = 5                                           01908000
TLLN = LINPAG                                    01909000
269 IF (FLN .GT. LTITLE) LN2 = FLN               01910000
ITEXT = LN2 - 1                                  01911000
IF (TXTLNE .GT. 1) LN = LN2                     01912000
TEXEND = TLLN * LINSIZ                           01913000
K = 10                                            01914000
C.....K, IF USED, SHOULD NOT BE LEFT .LE. 0    01915000
C                                                  01916000
293 RETURN                                       01917000
C                                                  01918000
C.....EOF ON INPUT DATASET                    01919000
710 IF (TITLE .NE. 0) CALL ERR (CCGCNT,800)      01920000
NODOC = 10 - TITLE * 80                         01921000
GO TO 293                                        01922000
END                                              01923000
C                                                  01924000
C                                                  01925000
C                                                  01926000
SUBROUTINE DRDR                                01927000
IMPLICIT INTEGER*4 (A - Z)                      01928000
DIMENSION CICNTX(2),INDP1(3),INDAR(7),LINE2(20),FMTOUT(35),
* FMTIN(22)                                     01929000
LOGICAL*1 PAGOV(15576),PAGOV1(15576),UCHAR,PULLN(132),CC,PLUS,BL,
* CCDUM(2),CRDC,INPUT(80)                      01930000
INTEGER*2 LINE(133),CUSTRT(8),CUEND(8),SW(66),EW(66),SW1,EW1,
* SAVE(4),SAVE1,SAVE2,SWEW2(4)                 01931000
* A,ONE,ZERO,FIVE,NINE,QUOTES,CENTS,EXPT,QMARK,DICTRY(10),TABCHR,
* TAB(14),RIIND,LIIND,RHIND,LHIND,INDP2(8).Z    01932000
INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS,
* PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR     01933000
INTEGER*2 MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK      01934000
INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN              01935000
COMMON IOUTPG,COL,LN,ERRCNT                     01936000
COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE,
* REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALD,BFOUND,CICNT1,CICNT,
* CIINC,DICT,NEXT,HIT,                          01937000
* ARRAY1(3),                                    01938000
* LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC,
* HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR
COMMON /B/ FIELD1,FIELD2,FIELD3,SPOP,CP,LB,PER,KEEPSV(2),
* UPPER,UP1,CAP,FIRST,MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK
COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,MYPAGE,RIVER,HYPTRX,WPT,SUND,TWO
* TWOUP,I,CPSW,ISPOT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTX,LWI,N,
* LINSIZ,NSYM,SPACNG,TXTLNE,K,LSTBL,AUTO,FPLN,NOGO,NAME,SWK028,IIU,
* CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTOTB,SWX,
* JUNK,ID,PDUM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN,
* LN2,REPTTL,IC,CWIDTH,USEWS1,LTITLE,CU(8),TAB2(7),INDP(4),
* INDARR(8),FLN,ICINC,PARA,TPLN,USTART,FCM,SWWPT,ENDL,EWX,PIVOT,
* INDEX,TABSEQ,CHAR,ENDF,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX,
* WANT,GAPS,WANTIN,WSEPD,LSIDE,RSIDE,SWEW(68),CARD(40),
* PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99)    01950000
01951000
01952000
01953000
01954000
01955000
01956000
01957000
01958000
01959000

```

```

DATA DICTRY/'P L S M A V C E J K W I U H D T Z F '/,PLUS/'+'/' 01960000
EQUIVALENCE (CUSTRT(1),CU(1)),(CUEND(1),CU(5)),(UCHAR,SPCHAR(38)), 01961000
* (LINE(1),LINEX(1)),(CICNTX(1),CICNT1),(INDP1(1),INDP(2)), 01962000
* (INDAR(1)), 01963000
* INDARR(2)),(LRI,INDP(2)),(LRH,INDP(4)),(SAVE1,SAVE(1)),(SAVE2, 01964000
* SAVE(2)),(SAVE(1),KEEPSV(1)),(KEEPS2,KEEPSV(2)),(LINE2(1), 01965000
* LINEX(1)),(FMTOUT(1),SWEW(1),SWEW2(1)),(CCDUM(1),SWEW(2)),(CC, 01966000
* CCDUM(2)),(SW(1),SW1,FULLN(1),SWEW2(4)),(BL,BLANK),(PMTIN(2), 01967000
* INPUT(1),CARD(2)),(CRDC,CARDIC),(EW(1),EW1,SWEW(36)),(PAGOV(1), 01968000
* PAGOV1(2),PAGDUM(1)) 01969000
*,(Z,NUM(6)),(A,NUM(1)),(ONE,NUM(8)),(ZERO,NUM(7)),(QUOTES, 01970000
* SPCHAR(32)),(FIVE,NUM(9)),(NINE,NUM(10)),(CENTS,SPCHAR(34)), 01971000
* (EXPT,SPCHAR(42)),(QMARK,SPCHAR(41)),(TAB(1),TAB2(1)),(LIIND, 01972000
* INDP2(3)),(RIIND,INDP2(4)),(LHIND,INDP2(7)),(RHIND,INDP2(8)), 01973000
* (INDP2(1),INDP(1)) 01974000
COMMON /EHRMAN/ DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF, 01975000
* BACKWD, BAKPOS, BACHAR, BACKST, BACKND, NULLSW, 01976000
* CCWIDT, NOTRIV, MASK2, EDCCWI, UNDRSW, EDCOL1 01977000
INTEGER*2 DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF, NULLSW, 01978000
* BACKWD(68), BAKPOS(100), BACKST(8), BACKND(8), CCWIDT, NOTRIV, 01979000
* MASK2, EDCCWI, UNDRSW, EDCOL1 01980000
LOGICAL*1 BACHAR(100) 01981000
INTEGER*2 BACKJA, BACKJB, BACKJC, JNKHLP 01982000
LOGICAL*1 JNKDUM(2) 01983000
EQUIVALENCE (JNKDUM(1),JNKHLP) 01984000
INTEGER*2 CHPUNC(10) /' ' ( " . , : ; ) ! ? ' / 01985000
C....PUNCTUATION CHARACTERS NOT TO BE UNDERLINED IF LAST ON UNDERGROUP 01986000
C***** 01987000
C***** 01988000
C ROUTINE TO READ AND NORMALIZE INPUT 01989000
C***** 01990000
C***** 01991000
IF (SPOP.EQ. 0) ASSIGN 917 TO SPOP 01992000
300 AUTOTB = AUTO * (1 - CENTER) 01993000
PDUM = LIIND + LHIND + PARA 01994000
LINEW = CHARCO - PDUM - RIIND - RHIND 01995000
IF (LINEW.GT. 0) GO TO 319 01996000
CALL ERR (1,300) 01997000
LRI = 0 01998000
LRH = 0 01999000
PDUM = 0 02000000
LINEW = CHARCO 02001000
319 IF (ASIS.EQ. 0) GO TO 936 02002000
ASSIGN 917 TO SPOP 02003000
AUTOTB = 0 02004000
END = 1 02005000
ENDF = 300 02006000
IC = FIELD2 02007000
PER = 0 02008000
UP1 = 0 02009000
GO TO 924 02010000
936 ENDF = LINEW + 1 02011000
TABSEQ = 0 02012000
924 WPT = WPTX 02013000
LNTW = WPT 02014000
TW = -1 02015000
HYPTRX = 0 02016000
ENDL = END - 1 02017000
ENDSAV = ENDL 02018000
IF (WPT.GT. 0) GO TO 909 02019000

```

SUND = UNDERL	02020000
LB = 10	02021000
909 IC = IC + ICINC	02022000
IF (IC .LE. FIELD2) GO TO 931	02023000
932 READ (IREAD,PMTIN,END=708)	02024000
CICNT = CICNT + CIINC	02025000
IC = FIELD1	02026000
931 DO 907 IC = IC, FIELD2	02027000
CRDC = INPUT(IC)	02028000
IF (CARDIC .EQ. NOTRIV) CARDIC = KEY028	02029000
GO TO SPOP, (917,900,901,311,935)	02030000
C.....NOW HANDLING TXT CTS XPT COP 2CH	02031000
900 ASSIGN 917 TO SPOP	02032000
IF (CARDIC .GE. A .AND. CARDIC .LE. NINE) GO TO 9001	02033000
C.....THE ABOVE IS CODE DEPENDENT	02034000
9002 BAXPTF = 0	02035000
BACKFL = 0	02036000
GO TO 917	02037000
C.....NONTRIVIAL BLANK LOOKS LIKE A NORMAL ALPHANUMERIC	02038000
9001 IF (CARDIC .EQ. KEY028) GO TO 9002	02039000
END = ENDSAV	02040000
IF (CARDIC .GT. Z) CARDIC = CARDIC - MASK2	02041000
GO TO 929	02042000
901 IF (EXCNT .NE. BLANK) GO TO 913	02043000
IF (CARDIC .LE. FIVE .AND. CARDIC .GE. ONE) GO TO 9011	02044000
BAXPTF = 0	02045000
BACKFL = 0	02046000
GO TO 914	02047000
9011 EXCNT = CARDIC	02048000
GO TO 907	02049000
913 ASSIGN 917 TO SPOP	02050000
IF (CARDIC .GT. NINE .OR. CARDIC .LT. ZERO) GO TO 915	02051000
N = (10 * EXCNT + CARDIC + 42048) / 256	02052000
C.....THE ABOVE IS CODE DEPENDENT	02053000
IF (N .LE. 42) GO TO 919	02054000
915 JUNK = EXCNT	02055000
EXCNT = BLANK	02056000
BAXPTF = 0	02057000
BACKFL = 0	02058000
930 LINE(END) = JUNK	02059000
ENDL = END	02060000
END = END + 1	02061000
IF (JUNK .NE. HYPHEN) GO TO 934	02062000
IF (END .LE. ENDF) GO TO 947	02063000
HYPTRX = 1	02064000
948 ASSIGN 935 TO SPOP	02065000
ICINC = 0	02066000
GO TO 955	02067000
947 EW(WPT) = -END	02068000
934 IF (END - ENDF) 914,914,948	02069000
919 CARDIC = SPCHAR(N)	02070000
EXCNT = BLANK	02071000
END = ENDSAV	02072000
IF (N .LE. 40) PER = 0	02073000
GO TO 929	02074000
311 ICINC = 1	02075000
IF (CP .EQ. 2) GO TO 339	02076000
940 IF (CARDIC .EQ. BLANK) GO TO 911	02077000
J = 1 + ASIS	02078000
C.....SINCE ASIS IS 0 OR 16, J IS 1 OR 17	02079000

DO 303 J = J, 18	02080000
IF (CARDIC .NE. DICTRY(J)) GO TO 303	02081000
INDEX = J	02082000
CP = 0	02083000
I = J - 10	02084000
IF (I .GT. 0) GO TO (338,338,325,338,324,332,910,923), I	02085000
C..... W I U H D T Z F	02086000
GO TO 307	02087000
303 CONTINUE	02088000
IF (ASIS .EQ. 0) CALL ERR (END + PDUM, 700)	02089000
CPSW = 0	02090000
IF (CP .EQ. 0) GO TO 914	02091000
937 LB = 0	02092000
UP1 = 0	02093000
WPT = WPT + 1	02094000
SW(WPT) = END + SUND	02095000
EW(WPT) = 0	02096000
ENDL = END	02097000
JUNK = CPAREN	02098000
GO TO 930	02099000
324 TABCHR = DROPCH	02100000
C.....TAB CHARACTER SET TO PRESET DROP CHARACTER	02101000
GO TO 338	02102000
332 TABCHR = BLANK	02103000
338 WANT = 0	02104000
CP = 2	02105000
GO TO 907	02106000
339 IF (CARDIC .GT. NINE .OR. CARDIC .LT. ZERO) GO TO 340	02107000
WANT = 10 * WANT + (CARDIC + 4032) / 256	02108000
C.....THE ABOVE IS CODE DEPENDENT	02109000
GO TO 907	02110000
340 CP = 0	02111000
IF (INDEX .EQ. 16 .OR. INDEX .EQ. 15) GO TO 322	02112000
IF (INDEX .NE. 14) GO TO 318	02113000
C.....INDEX = 14 IS H	02114000
IF (NEWH * WANT * (WANT - WANTIN) * (WANT + WANTIN - 1) .EQ. 0)	02115000
* NEWH = 1 - NEWH	02116000
WANTIN = WANT	02117000
GO TO 940	02118000
325 IF (LNTW .GT. 0) EW(LNTW) = EW(LNTW) + UNDERL - SUND	02119000
UNDERL = 16384 - UNDERL	02120000
SUND = UNDERL	02121000
GO TO 907	02122000
322 CHAR = END - 1	02123000
AUTOTB = 0	02124000
ICINC = 0	02125000
IF (WANT .EQ. 0) GO TO 3223	02126000
IF (WANT .GT. 14) GO TO 333	02127000
JUNK = TAB(WANT)	02128000
J = 14	02129000
GO TO 3224	02130000
C.....FORMERLY DO 327 J = 1, 14	02131000
3223 J = 1	02132000
3225 JUNK = TAB(J)	02133000
3224 IF (JUNK .GT. (CHAR + PDUM + TABSEQ)) GO TO 329	02134000
IF (JUNK .EQ. 0) GO TO 333	02135000
J = J + 1	02136000
327 IF (J .LE. 14) GO TO 3225	02137000
333 CALL ERR (CHAR,327)	02138000
GO TO 909	02139000

118

329	IF (JUNK .GT. LINEW + PDUM) GO TO 333	02140000
	TW = WPT	02141000
	TABSEQ = 1	02142000
	J = JUNK - END - PDUM	02143000
	IF (J .EQ. 0) GO TO 909	02144000
	WPT = WPT + 1	02145000
	TW = WPT	02146000
	SW(WPT) = END	02147000
	EW(WPT) = CHAR + J	02148000
330	IF (J .EQ. 0) GO TO 909	02149000
	J = J - 1	02150000
	LINE(END) = TABCHR	02151000
	END = END + 1	02152000
	GO TO 330	02153000
911	ASSIGN 917 TO SPOP	02154000
C.....	LB = 10 FROM THE BLANK PRECEDING THE COMMAND WORD	02155000
	CPSW = 0	02156000
C.....	REQUIRED BY JK MECHANISM	02157000
	UP1 = 0	02158000
	IF (PER + CAP + FIRST .GT. 10) UP1 = 10	02159000
	IF (CP .EQ. 0) GO TO 907	02160000
	IF (ASIS .EQ. 0 .OR. END .NE. 1) GO TO 937	02161000
	ASIS = 0	02162000
	NSYM = 1	02163000
	GO TO 300	02164000
910	UPPER = 10 - UPPER	02165000
	GO TO 907	02166000
923	FIRST = 20 - FIRST	02167000
	GO TO 907	02168000
905	CP = 1	02169000
	CPSW = 10	02170000
C.....	REQUIRED BY JK MECHANISM	02171000
	ASSIGN 311 TO SPOP	02172000
	GO TO 907	02173000
902	ASSIGN 900 TO SPOP	02174000
	PER = 0	02175000
	GO TO 916	02176000
903	ASSIGN 901 TO SPOP	02177000
	PER = 10	02178000
916	ENDSAV = END	02179000
	BACKPTF = BACKFL	02180000
C.....	WORRY ABOUT HELDUP CHARACTERS ONLY IF BACKING UP	02181000
	GO TO 920	02182000
904	JUNK = SWK028	02183000
	SWK028 = 0	02184000
	IF (BACKFL .EQ. 0) GO TO 9042	02185000
C.....	BACKSPACE FOLLOWED BY BLANK WILL BE IGNORED	02186000
	BACKFL = 0	02187000
9041	JUNK = BACKCH	02188000
	GO TO 930	02189000
9042	CONTINUE	02190000
	IF (LB .GT. 0) IF (JUNK + ASIS) 907,907,921	02191000
	TABSEQ = 0	02192000
C.....	END OF A WORD.	02193000
	EW(WPT) = ENDL	02194000
	SUND = 0	02195000
	LNTW = WPT	02196000
	LB = 10	02197000
	IF (PER + CAP + FIRST .GT. 10) UP1 = 10	02198000
	IF (PER .EQ. 0) GO TO 921	02199000

```

IF (EOSCHR + ASIS .EQ. 10 .AND. END .LT. LINEW) END = END + 1 02200000
GO TO 921 02201000
906 PER = 10 02202000
GO TO 920 02203000
935 ICINC = 1 02204000
914 ASSIGN 917 TO SPOP 02205000
C..... BACKSPACE CHARACTER RECOGNIZED ONLY IN TEXT, TREATED AS NORMAL 02206000
C..... SOURCE IF PRECEDED OR FOLLOWED BY BLANK. 02207000
917 IF (0 .NE. LB) GO TO 9171 02208000
IF (0 .EQ. BACKCH) GO TO 9171 02209000
IF (CARDIC .EQ. BACKCH) GO TO 7000 02210000
9171 IF (CARDIC .LT. 0) GO TO 939 02211000
C.....THE ABOVE IS CODE DEPENDENT 02212000
IF (CARDIC .EQ. BLANK) GO TO 904 02213000
IF (CARDIC .EQ. PERIOD) GO TO 906 02214000
IF (CARDIC .EQ. CPAREN .AND. LB .NE. 0) GO TO 905 02215000
IF (CARDIC .EQ. CENTS) GO TO 902 02216000
IF (CARDIC .EQ. EXPT) GO TO 903 02217000
IF (CARDIC .NE. HYPHEN) GO TO 944 02218000
IF (LB .NE. 0) GO TO 918 02219000
C.....DO NOT BREAK A LINE ON A HYPHEN WHICH BEGINS A WORD. 02220000
IF (END .LT. ENDF) GO TO 943 02221000
HYPTRX = 1 02222000
GO TO 918 02223000
7000 BACKFL = 1 02224000
GO TO 907 02225000
943 EW(WPT) = -END 02226000
SWK028 = 0 02227000
GO TO 918 02228000
939 IF (CARDIC .NE. KEY028) GO TO 918 02229000
CARDIC = BLANK 02230000
PER = 0 02231000
UP1 = 0 02232000
IF (SWK028 .EQ. 0) SWK028 = END 02233000
GO TO (908, 929), NULLSW 02234000
927 UP1 = 0 02235000
GO TO 929 02236000
944 IF (CARDIC .EQ. CPAREN .OR. CARDIC .EQ. QUOTES) GO TO 920 02237000
IF (CARDIC .EQ. QMARK) GO TO 906 02238000
918 PER = 0 02239000
920 IF (LB .EQ. 0) GO TO 946 02240000
C.....START OF A WORD. 02241000
LB = 0 02242000
WPT = WPT + 1 02243000
SW(WPT) = END + SUND 02244000
EW(WPT) = 0 02245000
946 IF (UP1 .NE. (-UPPER)) GO TO 927 02246000
IF (CARDIC .LE. Z .AND. CARDIC .GE. A) CARDIC = CARDIC - MASK1 02247000
929 ENDL = END 02248000
908 IF (BACKCH .EQ. 0) GO TO 9084 02249000
IF (LB .NE. 0) GO TO 9084 02250000
IF (CARDIC .EQ. BACKCH) GO TO 7000 02251000
9084 LINE(END) = CARDIC 02252000
IF (BACKFL .EQ. 0) GO TO 921 02253000
IF (CARDIC .EQ. BLANK) GO TO 921 02254000
IF (BACKCT .LT. 99) GO TO 9081 02255000
BACKFL = 0 02256000
IF (BACKCT .EQ. 16384) GO TO 9041 02257000
BACKCT = 16384 02258000
CALL ERR(WPT, 997) 02259000

```

	GO TO 9041	02260000
9081	IF (BAXPTF .GT. 1) GO TO 9086	02261000
	BACKCT = BACKCT + 1	02262000
9086	JUNK = SW(WPT)	02263000
	BAKPOS(BACKCT) = END - MOD(JUNK, 16384)	02264000
	IF (BACKWD(WPT) .EQ. 0) GO TO 9082	02265000
	BACKWD(WPT) = BACKWD(WPT) + 1 - (BAXPTF/2)	02266000
	GO TO 9083	02267000
9082	BACKWD(WPT) = BACKCT * 256	02268000
9083	IF (BACKST(COL) .EQ. 0) BACKST(COL) = BACKCT	02269000
	BACKND(COL) = BACKCT	02270000
	BACHAR(BACKCT) = CRDC	02271000
	LINE(END) = BLANK	02272000
	IF (BAXPTF .NE. 1) GO TO 9085	02273000
	BAXPTF = 2	02274000
	C....BACKFL = 0 MEANS NO BACKSPACE	02275000
	C....BACKFL = 1 MEANS BACKSPACE IN PROCESS OF BEING RESOLVED	02276000
	C....BAXPTF = 0 MEANS NORMAL BACKSPACING IN PROCESS	03377000
	C....BAXPTF = 1 MEANS CENTS OR EXCLAMATION IS TO BE PUT IN LINE, NOW	02278000
	C....WAITING TO SEE IF FOLLOWING CHARACTERS ARE PART OF BACKSPACE.	02279000
	C....BAXPTF = 2 MEANS NEXT CHARACTER WILL BE PUT IN AS A BACKSPACE, ON	02280000
	C....TOP OF THE SAVED CENTS OR BANG ALREADY THERE.	02281000
	GO TO 9087	02282000
9085	BAXPTF = 0	02283000
	BACKFL = 0	02284000
9087	END = END - 1	02285000
	ENDL = END	02286000
921	END = END + 1	02287000
	IF (END .GT. ENDF) GO TO 955	02288000
907	CONTINUE	02289000
	IF (ASIS .EQ. 0) GO TO 932	02290000
	C....GUARANTEED EXIT FOR 'AS IS' REGION TEXT LINE.	02291000
C	IF (CARDIC .EQ. BLANK) GO TO 925 USED TO BE HERE	02292000
	IF (INPUT(FIELD2) .EQ. BL) GO TO 925	02293000
C	CARD(FIELD2) = BLANK USED TO BE HERE, OVERWRITES END OF FORMAT	02294000
	INPUT(FIELD2) = BL	02295000
	IC = FIELD2	02296000
	GO TO 931	02297000
318	ICINC = 0	02298000
307	LWI = WPT + 1	02299000
	SW(LWI) = 16384	02300000
	IF (END .LT. ENDF) AUTOTB = 0	02301000
	END = 1	02302000
	RIVER = -10	02303000
	WPTX = 0	02304000
	J = WPT	02305000
	GO TO 922	02306000
925	SWK028 = 0	02307000
928	IF (WPT .EQ. 0) GO TO 600	02308000
	IF (EW(WPT) .LE. LINEW) GO TO 955	02309000
	IF (SW(WPT) .LE. LINEW) GO TO 926	02310000
	WPT = WPT - 1	02311000
	GO TO 928	02312000
926	EW(WPT) = LINEW	02313000
	C....MAKE LWI POINT TO THE LAST WORD TO BE INSERTED INTO PAGDUM NOW.	02314000
955	LWI = WPT	02315000
	J = EW(WPT)	02316000
	SWWPT = SW(WPT)	02317000
	SWWPT = MOD(SWWPT, 16384)	02318000
	WPTX = 1	02319000

```

      PCM = SWK028                                02320000
      IF (J) 950,951,952                          02321000
C.....LAST WORD BEGUN IS INCOMPLETE.            02322000
951  IF (LWI .EQ. 1) GO TO 956                    02323000
      LWI = LWI - 1                               02324000
      PCM = SWWPT                                02325000
      GO TO 953                                   02326000
956  CALL ERR (CHARCO,304)                        02327000
      SWK028 = 0                                  02328000
      J = -LINEW                                  02329000
C.....LAST BREAK CHARACTER WAS A HYPHEN.         02330000
950  EW (LWI) = -J                                02331000
      PCM = 1 - J                                02332000
      SWWPT = PCM                                02333000
      SUND = 0                                    02334000
      GO TO 953                                   02335000
C.....LAST WORD BEGUN IS COMPLETE (CARDIC = BLANK) OR IS PREFACED BY 023S02336000
952  WPTX = 0                                     02337000
953  J = LWI                                       02338000
      IF (SWK028 .GT. 0 .AND. SWK028 .LT. PCM) PCM = SWK028 02339000
922  CHAR = EW (J)                                02340000
      EW (J) = CHAR + UNDERL - SUND              02341000
      CHAR = MOD (CHAR,16384)                    02342000
      IF (J .EQ. TW) CALL ERR (CHARCO,922)        02343000
      IF (AUTOTB .EQ. 0 .OR. J .EQ. 1) GO TO 912  02344000
      GAPS = J - 1                               02345000
      WSEPD1 = (LINEW - CHAR) / GAPS              02346000
      REM = LINEW - CHAR - (WSEPD1 * GAPS)        02347000
      USEWS1 = GAPS - REM + 1                    02348000
912  CONTINUE                                     02349000
C*****02350000
C*****02351000
C  ROUTINE TO INSERT NORMALIZED TEXT INTO THE PAGE ARRAY 02352000
C*****02353000
C*****02354000
      IF (KEEP .EQ. 1) GO TO 209                  02355000
500  IER700 = (LN - 1) * LINSIZ                   02356000
      PDUM = PDUM + CENTER * (LINEW - CHAR) / 2  02357000
      IIU = IER700 + PDUM + COLBEG (COL)          02358000
      JUNK = SW1                                  02359000
      PIVOT = IIU + 16383 - 16384 * AUTOTB + MOD (JUNK,16384) 02360000
      PARA = 0                                    02361000
      RIVER = -RIVER                             02362000
      IF (NEWH .EQ. 0) GO TO 501                  02363000
      NEWH = 0                                    02364000
      I = 3                                       02365000
      GO TO 715                                   02366000
501  EWX = 0                                      02367000
      DO 502 K = 1, LWI                          02368000
      SWX = SW (K)                               02369000
      IIU = IIU + MOD (SWX,16384) - EWX - 1      02370000
      IF (IIU .LE. PIVOT) GO TO 506              02371000
      IIU = IIU + WSEPD1                          02372000
      IF (RIVER .GT. 0) GO TO 508                02373000
      IF (GAPS - USEWS1) 506,514,514            02374000
508  IF (GAPS .GT. REM) GO TO 506                02375000
514  IIU = IIU + 1                               02376000
506  IF (SWX .LT. 16384) GO TO 504              02377000
      IF (SWX .GT. 16384) GO TO 503             02378000
C.....SW (K) .GT. 2**14 MEANS START UNDERLINE HERE. 02379000

```

122

```

      IF (KEEP .NE. 1 .AND. K .EQ. 1) GO TO 700
      CCHAR = 10
      GO TO 600
503   SWX = SWX - 16384
      IF (USTART .LT. 99) GO TO 510
C....NOTE DEPENDENCE ON SIZE OF CHRFIN AND SRT
      CALL ERR (IIU - IER700 + 1 - COLBEG(COL), 513)
      USTART = 16384
      GO TO 504
510   USTART = USTART + 1
      IF (CUSTRT(COL) .EQ. 0) CUSTRT(COL) = USTART
      CUEND(COL) = USTART
      SRT(USTART) = IIU - IER700 + (LN * 256)
504   NSYM = 0
      JUNK = BACKWD(K)
      IF (JUNK .EQ. 0) GO TO 5049
      BACKJA = JUNK / 256
      JUNK = BACKJA + MOD(JUNK, 256)
      BACKJB = IIU - IER700 - 1 + (LN * 256)
      DO 5041 BACKJC = BACKJA, JUNK
5041  BACKPOS(BACKJC) = BACKPOS(BACKJC) + BACKJB
      BACKWD(K) = 0
5049  EWX = EW(K)
      EWX = MOD(EWX, 16384)
      DO 509 SWX = SWX, EWX
      CARDIC = LINE(SWX)
      PAGDUM(IIU) = CARDIC
      IF (TWOUP .GT. 1) PAGDUM(IIU + LINSZ) = CARDIC
      IIU = IIU + 1
509   CONTINUE
      GAPS = K
      IF (EW(K) .GT. 16384 .AND.
*      USTART .LE. 99) CHRFIN(USTART) = IIU + IIU
502   CONTINUE
C....END OF TEXT INSERTION. RE-INITIALIZE LINE AND ITS PARAMETERS.
      END = 1
      IF (WPTX + SWK028 .EQ. 0) GO TO 600
      SW1 = 1 + UNDERL + SWWPT - PCM
      DO 505 I = PCM, ENDF
      LINE(END) = LINE(I)
505   END = END + 1
      BACKWD(1) = BACKWD(LWI + 1)
      BACKWD(LWI + 1) = 0
      EW1 = HYPTRX * (1 - END)
C*****
C*****
C   PRINTER ROUTINE
C*****
C*****
600   TXTLNE = 1
      LN = LN + SPACNG
      IF (LN .LE. TLLN) GO TO 626
612   IF (KEEP .EQ. 20) GO TO 202
629   COL = COL + 1
      TXTLNE = 2
      IF (COL .LE. COLPAG) GO TO 601
622   TXTLNE = 3
      IF (PAGENO .EQ. 0) GO TO 609
      JUNK = PAGENO
      IF (CONST .NE. JUNK) GO TO 614

```

```

02380000
02381000
02382000
02383000
02384000
02385000
02386000
02387000
02388000
02389000
02390000
02391000
02392000
02393000
02394000
02395000
02396000
02397000
02398000
02399000
02400000
02401000
02402000
02403000
02404000
02405000
02406000
02407000
02408000
02409000
02410000
02411000
02412000
02413000
02414000
02415000
02416000
02417000
02418000
02419000
02420000
02421000
02422000
02423000
02424000
02425000
02426000
02427000
02428000
02429000
02430000
02431000
02432000
02433000
02434000
02435000
02436000
02437000
02438000
02439000

```

```

CONST = 10 * CONST                                02440000
ID = ID1                                           02441000
ID1 = ID1 + 1                                     02442000
614 I = COLBEG(1) + ID                            02443000
    POSN = I - ALT * (I - ISPOT)                  02444000
    CLEAN = IVALUE * (ISPOT + I - POSN - POSN) + POSN 02445000
C....ISPOT IS THE LAST PRINT POSITION OF THE LAST COLUMN (SET IN VRDR) 02446000
    ALT = IABS(IVALUE - ALT)                      02447000
    DO 621 I = 1, ID1                             02448000
    K = JUNK / 10                                  02449000
    PAGDUM(CLEAN) = BLANK                          02450000
    PAGDUM(POSN) = 256 * (JUNK - 10 * K) - 4032    02451000
C....THE ABOVE IS CODE DEPENDENT                  02452000
    IF (TWOUP .EQ. 1) GO TO 617                    02453000
    PAGDUM(CLEAN + LINSZ) = BLANK                  02454000
    PAGDUM(POSN + LINSZ) = PAGDUM(POSN)           02455000
617 JUNK = K                                        02456000
    POSN = POSN - 1                                02457000
    CLEAN = CLEAN - 1                              02458000
621 CONTINUE                                       02459000
    PAGENO = PAGENO + 1                            02460000
609 MYPAGE = MYPAGE + 1                            02461000
    IF (REPTTL .EQ. 0) LTITLE = 0                 02462000
    IF (LTITLE .EQ. 0) LN2 = 1                     02463000
    IF (CICNT .NE. 0) GO TO 619                     02464000
    WRITE (IWRITE,1001)                            02465000
1001 FORMAT ('1',132X)                             02466000
    GO TO 623                                       02467000
619 WRITE (IWRITE,1000) CICNTX                     02468000
1000 FORMAT ('1',120X,2I6)                         02469000
    CICNT1 = CICNT                                02470000
623 KEEPS2 = SW1                                   02471000
    BR1 = 1                                         02472000
    IF (USTART .NE. 0) BR1 = BR1 + 1               02473000
    IF (BACKCT .NE. 0) BR1 = BR1 + 2               02474000
    DO 610 I = 1, 66                               02475000
    IF (SW(I) .EQ. BLANK) GO TO 613                 02476000
    SW(I) = BLANK                                  02477000
610 CONTINUE                                       02478000
613 USTART = 0                                     02479000
    BACKCT = 0                                     02480000
    K = 0                                           02481000
    DO 602 LN = 1, LINPAG                          02482000
    DO 605 I = LSIDE,RSIDE                         02483000
605 FULLN(I) = PAGOV1(K + I + I)                  02484000
    WRITE (IWRITE,FMTOUT)                          02485000
    ASSIGN 6051 TO DRKRET                          02486000
    GO TO 800                                       02487000
6051 GO TO (608,606,6042,606), BR1                02488000
C*****02489000
C    A LITTLE RINKYDINK BOLDFACE DARK PRINTING SUBROUTINE 02490000
C*****02491000
800 IF (DARKER .LE. 1) GO TO 802                   02492000
    CC = PLUS                                       02493000
    DO 801 JUNK = 2, DARKER                        02494000
801 WRITE (IWRITE, FMTOUT)                         02495000
802 CC = BL                                         02496000
    GO TO DRKRET, (6051, 6071, 6049)              02497000
C*****02498000
C*****02499000

```

124

606	I = 0	02500000
	DO 604 COL = 1, COLPAG	02501000
616	CS = CUSTRT(COL)	02502000
	IF (CS .EQ. 0) GO TO 604	02503000
	JUNK = SRT(CS)	02504000
	IF (JUNK / 256 .NE. LN) GO TO 604	02505000
	JUNK = JUNK - LN * 256	02506000
	JUNK = JUNK + JUNK + K	02507000
	BACKJC = JUNK	02508000
	I = CHRFIN(CS) - 2	02509000
	JNKHLP = BLANK	02510000
	EWX = 0	02511000
	DO 615 JUNK = JUNK, I, 2	02512000
	IF (UNDRSW .EQ. 0) GO TO 6154	02513000
	IF (EWX .NE. 0) GO TO 6154	02514000
	JNKDUM(1) = PAGOV1(JUNK)	02515000
	IF (JNKHLP .EQ. CHPUNC(1)) GO TO 615	02516000
	IF (JNKHLP .EQ. CHPUNC(2)) GO TO 615	02517000
	IF (JNKHLP .EQ. CHPUNC(3)) GO TO 615	02518000
6154	EWX = JUNK	02519000
	PAGOV(JUNK) = UCHAR	02520000
	IF (TWOUP .GT. 1) PAGOV(JUNK + LINSZ + LINSZ) = UCHAR	02521000
615	CONTINUE	02522000
	IF (UNDRSW .EQ. 0 .OR. EWX .EQ. 0) GO TO 6153	02523000
6155	JNKDUM(1) = PAGOV1(EWX)	02524000
	DO 6151 JUNK = 1, 10	02525000
	IF (JNKHLP .EQ. CHPUNC(JUNK)) GO TO 6152	02526000
6151	CONTINUE	02527000
	GO TO 6153	02528000
6152	PAGOV(EWX) = BL	02529000
	IF (TWOUP .GT. 1) PAGOV(EWX + LINSZ + LINSZ) = BL	02530000
	EWX = EWX + 2	02531000
	IF (EWX .GE. BACKJC) GO TO 6155	02532000
6153	IF (CUEND(COL) .EQ. CS) CS = -1	02533000
	CUSTRT(COL) = CS + 1	02534000
	GO TO 616	02535000
604	CONTINUE	02536000
	IF (I .EQ. 0) GO TO 6042	02537000
	DO 607 I = LSIDE, RSIDE	02538000
607	PULLN(I) = PAGOV(K + I + I)	02539000
	CC = PLUS	02540000
	WRITE (IWRITE, FMTOUT)	02541000
	ASSIGN 6071 TO DRKRET	02542000
	GO TO 800	02543000
6071	CONTINUE	02544000
6042	IF (DR1 - 3) 608, 6043, 6041	02545000
6041	DO 6044 I = LSIDE, RSIDE	02546000
6044	PAGOV(K+I+I) = BL	02547000
6043	I = 0	02548000
	DO 6046 COL = 1, COLPAG	02549000
6045	CS = BACKST(COL)	02550000
	IF (CS .EQ. 0) GO TO 6046	02551000
	JUNK = BAKPOS(CS)	02552000
	IF (JUNK/256 .NE. LN) GO TO 6046	02553000
	I = JUNK - LN*256	02554000
	PAGOV(I + I + K) = BACHAR(CS)	02555000
	IF (TWOUP .GT. 1) PAGOV(I + I + K + LINSZ + LINSZ) = BACHAR(CS)	02556000
	IF (BACKND(COL) .EQ. CS) CS = -1	02557000
	BACKST(COL) = CS + 1	02558000
	GO TO 6045	02559000

6046	CONTINUE	02560000
	IF (I .EQ. 0) GO TO 6049	02561000
	DO 6047 I = LSIDE, RSIDE	02562000
6047	FULLN(I) = PAGOV(K+I+I)	02563000
	CC = PLUS	02564000
	WRITE (IWRITE, FMTOUT)	02565000
	ASSIGN 6049 TO DRKRET	02566000
	GO TO 800	02567000
6049	CONTINUE	02568000
608	K = K + LINSIZ + LINSIZ	02569000
602	CONTINUE	02570000
	SW1 = KEEPS2	02571000
	JUNK = (LN2 - 1) * LINSIZ + 1	02572000
	DO 618 I = JUNK, TEXEND	02573000
618	PAGDUM(I) = BLANK	02574000
	IF (LN2 .LT. FLN) LN2 = FLN	02575000
	ITEXT = LN2 - 1	02576000
	IOUTPG = MYPAGE	02577000
	COL = 1	02578000
601	LN = LN2	02579000
626	IF (CCHAR .NE. 0) GO TO 700	02580000
628	IF (INDEX .NE. 8) GO TO 300	02581000
611	RETURN	02582000
C*****		02583000
C*****		02584000
C	)N ROUTINE	02585000
C*****		02586000
C*****		02587000
C....	LN IS A CLEAN LINE ALWAYS	02588000
C....	NSYM = 0 IF HAVE ENDED A NON-EMPTY LINE	02589000
C....	= 1 IF LINE IS CLEAN, CONTAINS NO TEXT	02590000
C....	= 2 IF NEW COLUMN STARTED BY)C COMMAND OPERAND	02591000
C....	= 3 IF NEW PAGE STARTED BY)S COMMAND OPERAND	02592000
C....	TXTLNE = 1 IF CURRENT COLUMN-LINE IS SOMEWHERE IN MID-COLUMN	02593000
C....	= 2 IF AT TOP OF COLUMN WITH NORMAL TEXT RUNOVER	02594000
C....	= 3 IF AT TOP OF PAGE WITH NORMAL TEXT RUNOVER	02595000
700	CCHAR = 0	02596000
	JUNK = NSYM	02597000
	GO TO (705,702,705,730,731,611,704,701,706,709,707,711), INDEX	02598000
C....	P L S M A V C E J K W I	02599000
702	IF ((JUNK * TXTLNE) .EQ. 1) GO TO 600	02600000
7021	JUNK = 0	02601000
706	NSYM = 1	02602000
	IF (JUNK) 600,300,600	02603000
708	CALL ERR (1,800)	02604000
	NODOC = 10	02605000
	INDEX = 8	02606000
701	NSYM = 3	02607000
	LRI = 0	02608000
	LRH = 0	02609000
	INDP(1) = 0	02610000
	INDP(3) = 0	02611000
	KEEP = -IABS(KEEP)	02612000
	IF (TXTLNE .LE. 2 .OR. JUNK .GT. 2) GO TO 622	02613000
	GO TO 628	02614000
704	NSYM = 2	02615000
	KEEP = -IABS(KEEP)	02616000
	IF (TXTLNE .LE. 1 .OR. JUNK .GT. 1) GO TO 629	02617000
	GO TO 300	02618000
705	CENTER = 0	02619000

```

FIRST = 0
UNDERL = 0
UPPER = 0
PER = 10
IF (INDEX .EQ. 3) GO TO 701
PARA = INDENT
NSYM = 1
IF (TXTLNE .GT. 1) GO TO 300
C.....AT LEAST 2 LINES OF A NEW PARAGRAPH MUST APPEAR IN THE SAME COLUMN
C.....REQUIRES (BLNKLN+SPACNG) LINES TO FIT
JUNK = LN + BLNKLN + SPACNG
IF (JUNK .GT. TLLN) GO TO 612
LN = JUNK - SPACNG
GO TO 626
730 CENTER = 1 - CENTER
GO TO 7021
731 ASIS = 16
CPSW = 0
GO TO 300
707 IF (LN + WANT * SPACNG - SPACNG - TLLN) 300,300,612
709 IF (KEEP .EQ. 0) GO TO 710
KEEP = 0
GO TO 300
710 KEEP = -10
IF (TXTLNE .GT. 1) GO TO 300
C.....IGNORE ODD OCCURENCE OF ) K AT TOP OF COLUMN
C.....INITIALIZE KEEP ) K
KEEP = 20
CIINC = 1
SAVE1 = LN
SAVE2 = ERRCNT
KEEPS2 = CICNT
WRITE (1,1009) IC,CUSTRT(COL),USTART,UNDERL,CENTER,FIRST,SPOP,
* INDP,PER,UPPER,UP1,FMTIN
1009 FORMAT ('K',40X,14A4/'K',8X,22A4)
IF (IREAD .NE. 5) GO TO 300
IREAD = 2
REWIND 2
200 READ (5,1002,END=201) LINE2
1002 FORMAT (20A4)
WRITE (2,1002) LINE2
GO TO 200
201 ENDFILE 2
REWIND 2
GO TO 300
C.....KEEP = 20 AND THE LAST COLUMN-LINE HAS BEEN FILLED
202 IF (CPSW .GT. 0) IF (INDEX ~ 10) 203,629,203
C.....INDEX = 10 WHEN COMMAND OPERAND IS K
KEEP = 1
GO TO 300
209 IF (INDEX .EQ. 10 .AND. WPT .EQ. 0 .AND. CPSW .GT. 0) GO TO 500
C.....KEEP REQUIREMENT EXCEEDS AVAILABLE COLUMN SPACE
203 KEEP = 5
I = SAVE1
DO 204 I = I, TLLN
J = COLBEG(COL) + (I - 1) * LINSIZ
K = J + CHARCO - 1
DO 205 JUNK = J, K
PAGDUM(JUNK) = BLANK
IF (TWOUP .GT. 1) PAGDUM(JUNK + LINSZ) = BLANK

```

```

02620000
02621000
02622000
02623000
02624000
02625000
02626000
02627000
02628000
02629000
02630000
02631000
02632000
02633000
02634000
02635000
02636000
02637000
02638000
02639000
02640000
02641000
02642000
02643000
02644000
02645000
02646000
02647000
02648000
02649000
02650000
02651000
02652000
02653000
02654000
02655000
02656000
02657000
02658000
02659000
02660000
02661000
02662000
02663000
02664000
02665000
02666000
02667000
02668000
02669000
02670000
02671000
02672000
02673000
02674000
02675000
02676000
02677000
02678000
02679000

```

```

205 CONTINUE 02680000
204 CONTINUE 02681000
    I = ERRCNT - SAVE2 + 2 02682000
    DO 206 J = 1, I 02683000
    BACKSPACE 1 02684000
206 CONTINUE 02685000
    READ (1,1009) IC,CUSTRT(COL), USTART,UNDERL,CENTER,FIRST,SPOP,INDP, 02686000
    * PER,UPPER,UP1,FMTIN 02687000
    I = CICNT - KEEPS2 02688000
    CICNT = KEEPS2 02689000
207 IF (I .EQ. 0) GO TO 208 02690000
    I = I - 1 02691000
    BACKSPACE IREAD 02692000
    GO TO 207 02693000
208 ASIS = 0 02694000
    CCHAR = 0 02695000
    CP = 0 02696000
    CPSW = 10 02697000
    CUEND(COL) = USTART 02698000
    END = 1 02699000
    EXCNT = BLANK 02700000
    PARA = 0 02701000
    WPTX = 0 02702000
    GO TO 629 02703000
711 I = 1 02704000
    WANTIN = WANT 02705000
715 JUNK = INDP(I) 02706000
    IF (JUNK .EQ. 0) GO TO 714 02707000
    IF (WANTIN .EQ. JUNK .OR. WANTIN .EQ. 0) GO TO 712 02708000
714 IF (WANTIN .EQ. 0 .OR. WANTIN .GT. 7) WANTIN = 1 02709000
713 INDP1(I) = INDAR(WANTIN) 02710000
C.....INDAR(0) IS ZERO 02711000
    INDP(I) = WANTIN 02712000
    IF (I - 1) 300,300,501 02713000
712 WANTIN = 0 02714000
    GO TO 713 02715000
    END 02716000
C 02717000
C 02718000
C 02719000
    SUBROUTINE CONDSE (/EOF/) 02720000
    IMPLICIT INTEGER*4 (A - Z) 02721000
    DIMENSION STABLE(40),FMTIN(22) 02722000
    LOGICAL*1 BUFO(80),BL,BPCR 02723000
    INTEGER*2 BUFFER(80),A,V,E,PREN26,PREN29,VV(47),COPS(36), 02724000
    * BUFSAV(80),VAR(8),VAR1,VAR2,OVRRDE,F1,F2,CWORD,TITLEX,BUFPT 02725000
    *,BUFCHR 02726000
    INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS, 02727000
    * PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR 02728000
    INTEGER*2 MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK 02729000
    INTEGER*2 PAGDUM,GRT,COLBEG,CHRFIN 02730000
    COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE, 02731000
    * REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT, 02732000
    * CIINC,DICT,NEXT,HIT, 02733000
    * OVRRDE,F1,F2,CWORD,TITLEX,BUFPT, 02734000
    * LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC, 02735000
    * HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR 02736000
    COMMON /B/ FIELD1,FIELD2,FIELD3,SPOP,CP,LD,PER,KEEPSV(2), 02737000
    * UPPER,UP1,CAP,FIRST,MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK 02738000
    COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,HYPAGE,RIVER,HYPTRX,WPT,SUND,TWO 02739000

```

```

*,TWOUP,I,CPSW,ISPT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTY,LWI,N,02740000
* LINSIZ,NSYM,SPACNG,XTTLNE,K,LSTBL,AUTO,FPLN,NOGO,NAME,SWK028,IIU,02741000
* CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTOTB,SWX,02742000
* JUNK,ID,PDUM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN,02743000
* LN2,REPTTL,IC,CWIDTH,USEWS1,LTITLE,CU(8),TAB2(7),INDP(4),02744000
* INDARR(8),FLN,ICINC,PARA,TFLN,USTART,PCM,SWWPT,ENDL,ZWY,PIVOT,02745000
* INDEX,TABSEQ,CHAR,ENDP,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX,02746000
* WANT,GAPS,WANTIN,WSEPD,LSIDE,RSIDE,SWEW(68),CARD(40),02747000
* PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99)02748000
DATA VV02749000
*/ -1192,-20485,-8152,-8149,-12155,-19384,-20667,-20487,-20425,02750000
* -20155,-20151,-20121,-19941,-19883,-18650,-16344,-16332,-16265,02751000
* -16252,-15563,-15549,-15383,-15365,-15051,-14997,-14809,-14791,02752000
* -14743,-12489,-12487,-12442,-12316,-12295,-12221,-12217,-12215,02753000
* -12029,-12008,-11418,-20123,-20229,-19848,-20763,-15323,-20153,02754000
* -15545,-11738/02755000
EQUIVALENCE (VAR1,VAR(1)),(VAR2,VAR(2)),(A,NUM(1)),(V,NUM(5)),02756000
* (E,NUM(2)),(PREN29,SPCHAR(31)),(PREN26,SPCHAR(26))02757000
*,(BUFFER(1),LINEX(1)),(BUFO(1),CARD(2),FMTIN(2),STABLE(2)),(BL,02758000
* BLANK),(COPS(1),LINEX(41)),(BPCR,BUPCHR)02759000
COMMON /EHRMAN/ DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF,02760000
* BACKWD, BAKPOS, BACHAR, BACKST, BACKND, NULLSW,02761000
* CCWIDT, NOTRIV, MASK2, EDCCWI, UNDRSW, EDCOL102762000
INTEGER*2 DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF, NULLSW,02763000
* BACKWD(68), BAKPOS(100), BACKST(8), BACKND(8), CCWIDT, NOTRIV,02764000
* MASK2, EDCCWI, UNDRSW, EDCOL102765000
LOGICAL*1 BACHAR(100)02766000
EQUIVALENCE (CENTS,SPCHAR(34))02767000
INTEGER*2 CENTS02768000
C02769000
IF (EOF .NE. 0) GO TO 10502770000
WORDS = 002771000
COPPTR = 102772000
IF (DELETE .NE. 0) GO TO 10102773000
IF (COVEA - 5) 101,114,10202774000
C....COVEA (-5=END OF ASIS),(0=TEXT OR TITLE),(5=ASIS),(10=CONTROL CD)02775000
136 IF (F1.EQ. 1) GO TO 13902776000
J = 102777000
DO 137 I = F1, P202778000
BUFFER(J) = BUFFER(I)02779000
J = J + 102780000
137 CONTINUE02781000
139 J = F2 - F1 + 202782000
138 IF (J.GT. 80) GO TO 124102783000
BUFFER(J) = BLANK02784000
J = J + 102785000
GO TO 13802786000
114 IF (BUFFER(F1) .NE. CPAREN .OR. BUFFER(F1+1) .NE. BLANK) GO TO 13602787000
BUFFER(F1) = PREN2902788000
COVEA = -502789000
101 IBUFR = F102790000
IF (FINISH.GT. 0) IBUFR = IBUFSV02791000
DO 104 IBUFR = IBUFR, F202792000
IF (FINISH.EQ. 0) GO TO 14002793000
BUPCHR = BUFSV(IBUFR)02794000
GO TO 14102795000
140 BUPCHR = BUFFER(IBUFR)02796000
IF (COVEA.LT. 0) GO TO 12702797000
141 IF (CWORD.GT. DELETE) GO TO 10602798000
C....CWORD = 10 INSIDE COMMAND WORD02799000

```

126	IF (BUFCHR .EQ. CPAREN) GO TO 107	02800000
	IF (BUFCHR .EQ. BLANK) GO TO 108	02801000
130	IF (BFOUND .EQ. 0) GO TO 109	02802000
	WORDS = WORDS + 1	02803000
	IF (NODOC .EQ. (-30)) STABLE(WORDS) = IBUFR	02804000
	IF (WORDS .NE. INSWRD) GO TO 129	02805000
	REMNNT = 10	02806000
	IBUFRV = IBUFR	02807000
	IF (FINISH .NE. 0) GO TO 100	02808000
	DO 131 I = 1, 80	02809000
	BUFSAV(I) = BUFR(I)	02810000
131	CONTINUE	02811000
	GO TO 100	02812000
129	BFOUND = 0	02813000
109	IF (DELETE .NE. 0) GO TO 104	02814000
	IF (NODOC .LT. 0) GO TO 123	02815000
	BUFO(BUFR) = BFCR	02816000
	BUFR = BUFR + 1	02817000
	IF (BUFR .GT. 80) GO TO 110	02818000
	IF (COVEA .EQ. 0) GO TO 104	02819000
	DO 115 J = BUFR, 80	02820000
115	BUFO(J) = BL	02821000
110	WRITE (IREAD,FMTIN)	02822000
	BUFR = 1	02823000
123	IF (COVEA) 103,104,132	02824000
	C....FLAG BAD COMMAND OPERANDS DURING EDIT	02825000
106	IF (BUFCHR .LT. A) GO TO 1061	02826000
	IF (BUFCHR .LT. 0) GO TO 1062	02827000
	IF (BUFCHR .EQ. BLANK) GO TO 1062	02828000
	IF (BUFCHR .EQ. CENTS) GO TO 1062	02829000
	C....FLAG ERROR ONLY IF EDITING, SKIP IF 'CREATE' OR 'LIST'	02830000
1061	IF (K .GE. 0) CALL MSG(700)	02831000
1062	IF (NODOC .NE. (-30) .OR. COPPTR .GT. 36) GO TO 135	02832000
	COPS(COPPTR) = BUFCHR	02833000
	COPPTR = COPPTR + 1	02834000
135	IF (TITLX .NE. 0) GO TO 112	02835000
	C....TITLX = 10 INSIDE TITLE/FOOTER	02836000
	IF (BUFCHR .NE. A) GO TO 128	02837000
	COVEA = 5	02838000
	GO TO 127	02839000
128	IF (BUFCHR .EQ. V) GO TO 111	02840000
112	IF (BUFCHR .NE. E) GO TO 126	02841000
	IF (TITLX .NE. 0) GO TO 134	02842000
	IF (MERGE .EQ. -20) BUFCHR = V	02843000
	CPAREN = PREN29	02844000
	P1 = 1	02845000
	P2 = 80	02846000
134	TITLX = 0	02847000
111	COVEA = 10	02848000
	GO TO 127	02849000
107	BUFCHR = PREN29	02850000
	BUFR(IBUFR) = PREN29	02851000
	IF (BFOUND .GT. DELETE .AND. WORDS .LT. (INSWRD - 1)) CWORD = 10	02852000
	GO TO 130	02853000
108	IF (BFOUND .NE. 0) GO TO 104	02854000
127	BFOUND = 10	02855000
	CWORD = 0	02856000
	GO TO 109	02857000
103	COVEA = 0	02858000
104	CONTINUE	02859000

130

132	IF (FINISH .NE. 0) REMNNT = 0	02860000
100	RETURN	02861000
C....	COVEA = 10 ROUTINE	02862000
102	CALL CCRDR (VAR)	02863000
	DO 113 J = 1, 6	02864000
	IF (NAME .EQ. VV(J)) GO TO (116,117,118,119,120,120), J	02865000
C....	GO CAR 026 029 TIT POO	02866000
113	CONTINUE	02867000
	DO 133 J = 7, 47	02868000
	IF (NAME .EQ. VV(J)) GO TO 124	02869000
133	CONTINUE	02870000
	IF (NAME .LT. 0) NAME = 0	02871000
	GO TO 116	02872000
120	TITEX = 10	02873000
116	COVEA = 0	02874000
124	IF (J .NE. 45) GO TO 1241	02875000
C....	TAKE CARE OF CONTROL CARD WIDTH FOR CCRDR ROUTINE	02876000
	EDCCWI = VAR1	02877000
	IF (VAR1 .LE. 6 .OR. VAR1 .GT. 80) EDCCWI = 80	02878000
1241	IF (NODOC .LT. 0) GO TO 100	02879000
	WRITE (IREAD,2000) (BUFFER(JUNK), JUNK = 1, EDCCWI)	02880000
2000	FORMAT (80A1)	02881000
	GO TO 100	02882000
117	IF (VAR2 .NE. 0) GO TO 121	02883000
	VAR2 = VAR1	02884000
	VAR1 = 1	02885000
121	IF (VAR1 .LT. 1 .OR. VAR2 .GT. 80 .OR. (VAR1 + 2) .GT. VAR2 .OR.	02886000
	* IREAD .EQ. 4) GO TO 100	02887000
	P1 = VAR1	02888000
	P2 = VAR2	02889000
	GO TO 100	02890000
118	IF (IREAD .EQ. 2) CPAREN = PREN26	02891000
	GO TO 100	02892000
119	CPAREN = PREN29	02893000
	GO TO 100	02894000
C*****	ROUTINE FOR EOF ON INPUT VOLUME *****	02895000
105	IF (BUFPT .EQ. 1) GO TO 122	02896000
	WRITE (IREAD,PMTIN)	02897000
122	IF (MERGE .NE. 0) GO TO 100	02898000
	ENDFILE IREAD	02899000
	REWIND IREAD	02900000
	CPAREN = PREN29	02901000
	GO TO 100	02902000
	END	02903000
C		02904000
C		02905000
C		02906000
	SUBROUTINE CCRDR (VAR)	02907000
	IMPLICIT INTEGER*4 (A - Z)	02908000
	INTEGER*2 BUFFER(80), BUFCHR, NINE, ZERO, VAR(8)	02909000
	INTEGER*2 LIST, COPIES, SPCHAR, BLANK, NUM, SCWORD, HYPHEN, LOWCAS,	02910000
	* PERIOD, KEY028, LOCATE, CARDIC, ONLIST, DOLLAR	02911000
	INTEGER*2 PAGDUM, SRT, COLBEG, CHRFIN	02912000
	COMMON /A/ POSN, IREAD, IWRITE, CCGCNT, PUNCH, NODOC, MERGE,	02913000
	* REMNNT, INSWRD, FINISH, DELETE, COVEA, INVALID, BFOUND, CICNT1, CICNT,	02914000
	* CIINC, DICT, NEXT, HIT,	02915000
	* ARRAY1(3),	02916000
	* LIST, COPIES, SPCHAR(42), BLANK, NUM(10), SCWORD, CARDIC,	02917000
	* HYPHEN, LOWCAS, PERIOD, KEY028, LOCATE, ONLIST, DOLLAR	02918000
	COMMON /C/ ALT, BLNKLN, ITEXT, LINPAG, MYPAGE, RIVER, HYPTRX, WPT, SUND, TWO	02919000

```

* ,TWOUP,I,CPSW,ISPT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTX,LWI,N,02920000
* LINSIZ,NSYM,SPACNG,TXTLNE,K,LSTBL,AUTO,PFLN,NOGO,NAME,SWK028,IIU,02921000
* CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTOTB,SWX,02922000
* JUNK,ID,PDUM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN,02923000
* LN2,REPTTL,IC,CWIDTH,USEWS1,LITLE,CU(8),TAB2(7),INDP(4),02924000
* INDARR(8),FLN,ICINC,PARA,TPLN,USTART,FCH,SWWPT,ENDL,EWX,PIVOT,02925000
* INDEX,TABSEQ,CHAR,ENDF,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX,02926000
* WANT,GAPS,WANTIN,WSEPD,LSIDE,RSIDE,SWEW(68),CARD(40),02927000
* PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99)02928000
COMMON /EHRMAN/ DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF,02929000
* BACKWD, BAKPOS, BACHAR, BACKST, BACKND, NULLSW,02930000
* CCWIDT, NOTRIV, MASK2, EDCCWI, UNDRSW, EDCOL102931000
INTEGER*2 DARKER, DROPCH, BACKCH, BACKFL, BACKCT, BAXPTF, NULLSW,02932000
* BACKWD(68), BAKPOS(100), BACKST(8), BACKND(8), CCWIDT, NOTRIV,02933000
* MASK2, EDCCWI, UNDRSW, EDCOL102934000
LOGICAL*1 BACHAR(100)02935000
EQUIVALENCE (BUFFER(1),LINEX(1)),(NINE,NUM(10)),(ZERO,NUM(7))02936000
DO 200 I = 1, 802937000
200 VAR(I) = 002938000
K = 002939000
NAME = 002940000
DO 282 I = 1, EDCCWI02941000
BUFCHR = BUFFER(I)02942000
IF (BUFCHR .EQ. BLANK) GO TO 28202943000
IF (K .EQ. 0) EDCOL1 = BUFCHR02944000
C.....SAVE CHARACTER FOR DOLLAR TEST - AVOID HOMONYMS DURING EDIT02945000
K = K + 102946000
NAME = 16 * NAME + (BUFCHR - 3648) / 25602947000
C.....THE ABOVE IS CODE DEPENDENT02948000
IF (K .EQ. 3) GO TO 28502949000
282 CONTINUE02950000
285 K = 102951000
280 ASSIGN 284 TO HIT102952000
284 IF (I .GE. EDCCWI) GO TO 28902953000
I = I + 102954000
BUFCHR = BUFFER(I)02955000
IF (BUFCHR .LE. NINE .AND. BUFCHR .GE. ZERO) GO TO 28802956000
GO TO HIT1, (284,291)02957000
291 IF (K .EQ. 8) GO TO 28902958000
K = K + 102959000
GO TO 28002960000
288 VAR(K) = 10 * VAR(K) + (BUFCHR + 4032) / 25602961000
C.....THE ABOVE IS CODE DEPENDENT02962000
ASSIGN 291 TO HIT102963000
GO TO 28402964000
289 RETURN02965000
END02966000
C02967000
C02968000
C02969000
SUBROUTINE ENDJOB02970000
C***** ROUTINE FOR END OF JOB STREAM *****02971000
IMPLICIT INTEGER*4 (A - Z)02972000
INTEGER*4 LINE1(35), OUTFMT(3) /'(133H ) '/02973000
INTEGER*2 OVRNDE,F1,F2,CWORD,TITLEX,BUFFT02974000
INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS,02975000
* PERIOD,KEY028,LOCATE,CARDIC,ONLIST,DOLLAR02976000
INTEGER*2 MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK02977000
INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN02978000
COMMON /A/ POSN,IREAD,IWRITE,CCGNT,PUNCH,NODOC,MERGE,02979000

```

```

* REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT, 02980000
* CIINC,DICT,NEXT,HIT, 02981000
* OVRDE,F1,F2,CWORD,TITLEX,BUFPT, 02982000
* LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC, 02983000
* HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR 02984000
COMMON /B/ FIELD1,FIELD2,FIELD3,SPOP,CP,LB,PER,KEEPSV(2), 02985000
* UPPER,UP1,CAP,FIRST,MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK 02986000
COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,MYPAGE,RIVER,HYPTRY,WPT,SUND,TWO 02987000
*,TWOUP,I,CPSW,ISPT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTX,LWI,N, 02988000
* LINSIZ,NSYM,SPACNG,TXTLNE,K,LSTBL,AUTO,PFLN,NOGO,NAME,SWK028,IIU, 02989000
* CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTOTB,SWX, 02990000
* JUNK,ID,PDM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN, 02991000
* LN2,REPTTL,IC,CWIDTH,USEWS1,LITLE,CU(8),TAB2(7),INDP(4), 02992000
* INDARR(8),FLN,ICINC,PARA,TFLN,USTART,PCM,SWWPT,ENDL,EWX,PIVOT, 02993000
* INDEX,TABSEQ,CHAR,ENDF,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX, 02994000
* WANT,GAPS,WANTIN,WSEPD,LSIDE,RSIDE,SWEW(68),CARD(40), 02995000
* PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99) 02996000
IF (IREAD .NE. 5) REWIND IREAD 02997000
F1 = 1 02998000
F2 = 80 02999000
CWORD = 0 03000000
IF (IWRITE .EQ. 6) GO TO 700 03001000
IF (NODOC .EQ. 0) ENDFILE IWRITE 03002000
705 IF (COPIES .EQ. 0) GO TO 700 03003000
COPIES = COPIES - 1 03004000
LINE1(1) = OUTFMT(1) 03005000
LINE1(2) = OUTFMT(2) 03006000
LINE1(35) = OUTFMT(3) 03007000
REWIND IWRITE 03008000
701 READ (IWRITE,LINE1,END=705) 03009000
WRITE (6,LINE1) 03010000
GO TO 701 03011000
700 IF (LOCATE .EQ. 0) GO TO 702 03012000
C.....ALL VALUES IN /C/ WILL NOW BE DESTROYED. 03013000
CALL LOC1 03014000
CALL LOC2 03015000
CALL LOC3 03016000
702 IF (LIST .EQ. 0) GO TO 703 03017000
IF (OMLIST .NE. 0) GO TO 703 03018000
CALL LISTER 03019000
703 IF (DICT .EQ. 0) STOP 03020000
C.....ALL VALUES IN /C/ WILL NOW BE DESTROYED. 03021000
CALL PREVNT 03022000
CALL DICTNY 03023000
C.....IREAD = 3 NOW. 03024000
RETURN 03025000
END 03026000
C 03027000
C 03028000
C 03029000
SUBROUTINE LISTER 03030000
IMPLICIT INTEGER*4 (A - Z) 03031000
DIMENSION COPSD(18),STABLE(40),TEXT2(20),COPS(9),FMT0(2),FMT0U(22) 03032000
LOGICAL*1 TEXT1(116),DG1,DG2,BUFR(81),SVMSK 03033000
INTEGER*2 TITLX,BUFR(80),DIG1,DIG2,BUF(116),CSIGN,EXPT,ASTRSK, 03034000
* ZERO,NINE 03035000
INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS, 03036000
* PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR 03037000
INTEGER*2 MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK 03038000
INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN 03039000

```



```

COMMON /A/ POSN, IREAD, IWRITE, CCGCNT, PUNCH, NODOC, MERGE, 03040000
* REMNNT, INSWRD, FINISH, DELETE, COVEA, INVALID, BFOUND, CICNT1, CICNT, 03041000
* CIINC, DICT, NEXT, HIT, 03042000
* ARRAY1(3), 03043000
* LIST, COPIES, SPCHAR(42), BLANK, NUM(10), SCWORD, CARDIC, 03044000
* HYPHEN, LOWCAS, PERIOD, KEY028, LOCATE, ONLIST, DOLLAR 03045000
COMMON /B/ FIELD1, FIELD2, FIELD3, SPOP, CP, LB, PER, KEEPSV(2), 03046000
* UPPER, UP1, CAP, FIRST, MASK1, EXCNT, CPAREN, EOSCHR, SAVMSK 03047000
COMMON /C/ ALT, BLNKLN, ITEXT, LINPAG, HYPAGE, RIVER, HYPTRY, WPT, SUND, TWO 03048000
*, TWOUP, I, CPSW, ISPT, ASIS, CCHAR, CHARCO, NEWH, SAVCCC, KEEP, WPTX, LWI, N, 03049000
* LINSIZ, NSYM, SPACNG, TXTLNE, K, LSTBL, AUTO, PFLN, NOGO, NAME, SWK028, IIU, 03050000
* CCCNT, COLPAG, IVALUE, LINSZ, PAGENO, START, UNDERL, J, CONST, AUTOTB, SWX, 03051000
* JUNK, ID, PDUM, REM, IER700, END, CENTER, CSEP, INDENT, TEXEND, TLLN, CLEAN, 03052000
* LN2, REPTTL, IC, CWIDTH, USEWS1, LTITLE, CU(8), TAB2(7), INDP(4), 03053000
* INDARR(8), FLN, ICINC, PARA, TPLN, USTART, FCH, SWWPT, ENDL, EWX, PIVOT, 03054000
* INDEX, TABSEQ, CHAR, ENDF, LINEX(67), WORDS, LNTW, CS, ENDSAV, ID1, LINEW, 03055000
* WANT, GAPS, WANTIN, WSEPDL, LSIDE, RSIDE, SWEW(68), CARD(40), 03056000
* PAGDUM(7788), SRT(99), COLBEG(8), CHRFIN(99) 03057000
DATA BLBL/' ',ASTRSK/'* ',XIMNO/1/,XLNNO/62/,FMTO/'(82H ) '/03058000
EQUIVALENCE (LINEX(1),BUFFER(1),BUF(1),TEXT1(1),TEXT2(1),FMTOU(1))03059000
*,(LINEX(3),BUFFR(4)),(LINEX(21),COPS(1)),(LINEX(41),COPSD(1)), 03060000
*(DG1,DIG1),(DG2,DIG2),(SVMSK,SAVMSK),(CSIGN,SPCHAR(34)),(EXPT, 03061000
* SPCHAR(42)),(ZERO,NUM(7)),(STABLE(1),CARD(1)),(NINE,NUM(10)), 03062000
*(TITLEX,ARRAY1(3)) 03063000
DATA XLNPP/60/ 03064000
NODOC = -30 03065000
INSWRD = 10000 03066000
COVEA = 10 03067000
BFOUND = 10 03068000
TITLEX = 0 03069000
ASSIGN 713 TO BR3 03070000
IF (SAVMSK .EQ. 0) ASSIGN 715 TO BR3 03071000
PAGENO = 0 03072000
703 READ (IREAD,2002,END=702) BUFFER 03073000
2002 FORMAT (80A1) 03074000
IF (XLNNO .LE. XLNPP) GO TO 704 03075000
PAGENO = PAGENO + 1 03076000
WRITE (6,2001) PAGENO 03077000
2001 FORMAT ('1',32X,15HINPUT TO FORMAT,37X,'CARD IMAGE COMMAND OPERAN03078000
*DS',6X,'PAGE',I4/) 03079000
XLNNO = 3 03080000
704 IF (PUNCH .EQ. 0) GO TO 706 03081000
WRITE (7,2002) BUFFER 03082000
706 XLNNO = XLNNO + 1 03083000
DO 708 XI = 1, 18 03084000
708 COPSD(XI) = BLBL 03085000
CALL CONDSE (0) 03086000
DO 700 XI = 1, 116 03087000
DIG1 = BUF(XI) 03088000
GO TO BR3,(713,715) 03089000
715 IF (DIG1 .LT. 0) GO TO 713 03090000
IF (DIG1 .EQ. CSIGN) GO TO 714 03091000
IF (DIG1 - EXPT) 712,714,712 03092000
713 IF (DIG1 .NE. KEY028) GO TO 712 03093000
714 DIG1 = ASTRSK 03094000
712 TEXT1(XI) = DG1 03095000
700 CONTINUE 03096000
WRITE (6,2003) TEXT2,XIMNO,COPS 03097000
2003 FORMAT (' ',20A4,4X,I7,5X,9A4) 03098000
XIMNO = XIMNO + 1 03099000

```

```

IF (WORDS .EQ. 0) GO TO 703                                03100000
DO 707 XI = 2, 21                                           03101000
707  FMTOU(XI) = BLBL                                       03102000
    FMTOU(1) = FMTO(1)                                       03103000
    FMTOU(22) = FMTO(2)                                       03104000
C.....FROM THIS POINT TILL LABEL 709 THE PROGRAM IS HIGHLY CODE-DEPENDENT 03105000
    DIG1 = ZERO                                              03106000
    DIG2 = ZERO                                              03107000
    DO 709 XI = 1, WORDS                                     03108000
    XJ = STABLE(XI)                                          03109000
    DIG2 = DIG2 + 256                                        03110000
    IF (DIG2 .LE. NINE) GO TO 710                           03111000
    DIG1 = DIG1 + 256                                        03112000
    DIG2 = ZERO                                              03113000
710  IF (DIG1 .EQ. ZERO) GO TO 711                           03114000
    BUFR(XJ) = DG1 - SVMSK                                   03115000
    XJ = XJ + 1                                              03116000
711  BUFR(XJ) = DG2 - SVMSK                                   03117000
709  CONTINUE                                               03118000
    WRITE (6,FMTOU)                                         03119000
    XLNNO = XLNNO + 1                                       03120000
    GO TO 703                                                03121000
702  REWIND IREAD                                           03122000
    RETURN                                                  03123000
    END                                                    03124000
C                                                         03125000
C                                                         03126000
C                                                         03127000
    SUBROUTINE PREVNT                                       03128000
    IMPLICIT INTEGER*4 (A - Z)                             03129000
    DIMENSION WLX(338)                                       03130000
    INTEGER*2 PREVN1(517),PREVN2(516),PREVN3(518),LTR,R,LTR1, 03131000
    * LTR2,II,WL(8285),WLL(8284),WLC(8283)                 03132000
C.....WL, WLL, WLC MAY BE DIMENSIONED THE SAME AS IN DICTNY, IF NECESSARY 03133000
    INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS, 03134000
    * PERIOD,KEY028,LOCATE,CARDIC,ONLIST,DOLLAR             03135000
    INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN                     03136000
    COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE, 03137000
    * REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT, 03138000
    * CIINC,DICT,NEXT,HIT,                                  03139000
    * ABRAY1(3),                                           03140000
    * LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC, 03141000
    * HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,ONLIST,DOLLAR      03142000
    COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,HYPAGE,RIVER,HYPTRX,WPT,SUND,TW03143000
    *,TWOUP,I,CPSW,ISPOT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTX,LWI,N,03144000
    * LINSIZ,NSYM,SPACNG,TXTLNE,K,LSTBL,AUTO,PFLN,NOGO,NAME,SWK028,IIU,03145000
    * CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTOTB,SWK,03146000
    * JUNK,ID,PDUM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN,03147000
    * LN2,REPTTL,IC,CWIDTH,USEWS1,LTITLE,CU(8),TAB2(7),INDP(4), 03148000
    * INDARR(8),FLN,ICINC,PABA,TFLN,USTART,FCM,SWWPT,ENDL,EWX,PIVOT, 03149000
    * INDEX,TABSEQ,CHAR,ENDF,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX, 03150000
    * WANT,GAPS,WANTIN,WSEPDL,LSIDE,RSIDE,SNEW(68),CARD(40), 03151000
    * PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99)           03152000
    INTEGER*2 PR1(100), PR2(100), PR3(100), PR4(100), PR5(17) 03153000
    EQUIVALENCE (PREVN1(101),PR1(1)), (PREVN1(201),PR2(1)), 03154000
    * (PREVN1(301),PR3(1)), (PREVN1(401),PR4(1)),          03155000
    * (PREVN1(501),PR5(1))                                  03156000
    DATA FLTR/1/,INC/4/,R/'R '/,II/'I '/,LIMIT/517/      03157000
C.....LIMIT = SIZE OF PREVN1, PREVN2 = SIZE - 1, PREVN3 = SIZE + 1 03158000
C.....PREVN1 WORDS MUST BE IN DESCENDING ALPHABETICAL ORDER 03159000

```

```

DATA PREVN1/'WOULD WITHOUT WITHIN WITH W03160000
* I L L   W H I C H   W H E T H E R   W H E R E   W H E N E V E R   03161000
* W H E N   W H A T   W E R E   W A S   V E R Y   T W O   T O O   T03162000
* H U S   T H R   '/                                03163000
DATA PR1                                           03164000
*           /'O U G H   T H O U G H   T H O S E   T H I S   T H   E03165000
* Y   T H E S E   T H E R E F O R E   T H E R E   T H E N   T H E M03166000
* T H E I R   T H E   T H A T   T H A N   S U C H   S O M E   S I03167000
* N C E   S H O U   '/                                03168000
DATA PR2                                           03169000
*           /   ' L D   S H A L L   S A Y S   S A Y   S A M E   S A03170000
* I D   P E R   O T H E R   O N L Y   O N E   N O W   N O T   M U S03171000
* T   M U C H   M O R E   M A Y   M A N Y   I T S   I N T O   H O W03172000
* E V E R   H O W   '/                                03173000
DATA PR3                                           03174000
*           /   ' H E R E   H A V E   H A S   H A D   F R O M   F03175000
* O R   F I R S T   E V E R Y   E T C   E I T H E R   E A C H   D U03176000
* R I N G   D O N E   D O E S   D I D   C O U L D   C A N N O T   C03177000
* A N   B U T   B O T   '/                                03178000
DATA PR4                                           03179000
*           /   ' H   B E T W E E N   B E I N G   B E F O R E   03180000
* B E E N   B E C A U S E   A R E   A N Y   A N O T H E R   A N D   03181000
* A M O N G   A L W A Y S   A L T H O U G H   A L S O   A L M O S T03182000
*   A L L   A G A I N   '/                                03183000
DATA PR5                                           03184000
*           /   ' A P T E R   A B O U T   A B L E   '/       03185000
EQUIVALENCE (PREVN2(1),PREVN1(2)),                03186000
* (PREVN3(2),PREVN1(1)), (R,PREVN3(1)), (WL(1),ALT), (WLX(1),WL(1)), 03187000
* (WLL(1),WL(2)), (WLC(1),WL(3))                    03188000
DO 100 XI = 1, 338                                  03189000
100  WLX(XI) = 0                                       03190000
101  LTR = PREVN1(INC)                                03191000
      INC = INC + 1                                    03192000
      IF (LTR .NE. BLANK) GO TO 101                    03193000
      LTR1 = PREVN1(FLTR)                              03194000
      LTR2 = PREVN2(FLTR)                              03195000
C.....FROM HERE THRU AND INCLUDING LABEL 107 THIS ROUTINE IS CODE-DEPENDT03196000
      IF (LTR2 .LE. R) GO TO 104                       03197000
      LTR2 = LTR2 - 3840                                03198000
      GO TO 105                                         03199000
104  IF (LTR2 .GT. II) LTR2 = LTR2 - 1792              03200000
105  IF (LTR1 .LE. R) GO TO 106                       03201000
      LTR1 = LTR1 - 3840                                03202000
      GO TO 107                                         03203000
106  IF (LTR1 .GT. II) LTR1 = LTR1 - 1792              03204000
107  OVADDR = ((LTR1 * 26) + LTR2 + 433984) / 256      03205000
      VADDR = WL(OVADDR)                               03206000
      WL(OVADDR) = NEXT                                03207000
      WL(NEXT) = VADDR                                  03208000
      LENGTH = INC - FLTR - 1                          03209000
      WLL(NEXT) = LENGTH                                03210000
      WLC(NEXT) = -32767                                03211000
      DO 111 XI = 3,LENGTH                             03212000
      WL(NEXT+XI) = PREVN3(FLTR+XI)                   03213000
111  CONTINUE                                           03214000
      NEXT = NEXT + LENGTH + 1                         03215000
      FLTR = INC                                         03216000
      INC = INC + 2                                     03217000
      IF (INC .LT. LIMIT) GO TO 101                   03218000
      RETURN                                           03219000

```

```

END 03220000
C 03221000
C 03222000
C 03223000
SUBROUTINE DICTNY 03224000
IMPLICIT INTEGER*4 (A - Z) 03225000
DIMENSION WLX(338) 03226000
INTEGER*2 SECTN,RDBUF(80),Z,AA,CSIGN,WORD2(41),WORD(40),INDARH(14) 03227000
*,R,II,WL(8430),WLL(8429),WLC(8428),UWORD,TITXT(26),WORD1,LEN(2) 03228000
C.....WL CAN BE DIMENSIONED FOR THE FULL /C/, LIMIT = DIMENSION OF WL 03229000
INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS, 03230000
* PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR 03231000
INTEGER*2 MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK 03232000
INTEGER*2 PAGNUM,SRT,COLREG,CHRFIN 03233000
COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE, 03234000
* REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT, 03235000
* CIINC,DICT,NEXT,HIT, 03236000
* ARRAY1(3), 03237000
* LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC, 03238000
* HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR 03239000
COMMON /B/ FIELD1,FIELD2,FIELD3,SPOP,CP,LB,PER,KEEPSV(2), 03240000
* UPPER,UP1,CAP,FIRST,MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK 03241000
COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,HYPAGE,RIVER,HYPTRX,WPT,SUND,TWO 03242000
*,TWOUP,I,CPSW,ISPT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTY,LWI,N, 03243000
* LINSIZ,NSYM,SPACNG,TKTLNE,K,LSTBL,AUTO,FPLN,NOGO,NAME,SWK028,II0, 03244000
* CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUFOTB,SWI, 03245000
* JUNK,ID,PDUM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN, 03246000
* LN2,REPTTL,IC,CWIDTH,USEWS1,LTITLE,CU(8),TAB2(7),INDP(4), 03247000
* INDARR(8),FLN,ICINC,PARA,TPLN,USTART,FCM,SWWPT,ENDL,EWX,PIVOT, 03248000
* INDEX,TABSEQ,CHAR,ENDF,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX, 03249000
* WANT,GAPS,WANTIN,WSEPD,LSIDE,RSIDE,SEW(68),CARD(40), 03250000
* PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99) 03251000
DATA SECTN/1/,WCNT/0/,R/'R'/',II/'I'/',LIMIT/8430/,UWORD/0/, 03252000
* WORD1/' '/,WORD/40*' '/, 03253000
DATA TITXT/'W O R D O C C U R R E N C E D I C T I O N A R Y ' 03254000
*/ 03255000
EQUIVALENCE (Z,NUM(6)),(AA,NUM(1)),(CSIGN,SPCHAR(34)), 03256000
* (WORD2(2),WORD(1)),(WL(1),ALT),(WLX(1),WL(1)) 03257000
*,(WLL(1),WL(2)),(WLC(1),WL(3)),(WORD1,WORD(41)),(WORD2(1),LEN(2)), 03258000
* (LEN(1),LENGTH),(INDARH(1),INDARR(2)) 03259000
C.....WORD ISOLATING ROUTINE 03260000
C.....CODE-DEPENDENT 03261000
ASSIGN 502 TO BR4 03262000
529 REWIND 1 03263000
C.....ABOVE MUST JUST PRECEDE 520 03264000
520 CIINC = 0 03265000
CICNT = 0 03266000
500 CCOL = 1 03267000
IF (SECTN .GT. 1) GO TO 525 03268000
READ (IREAD,1001,END=509) RDBUF 03269000
1001 FORMAT (80A1) 03270000
525 DO 501 CCOL = CCOL, 80 03271000
CARDIC = RDBUF(CCOL) 03272000
GO TO BR4, (503,502,516,526) 03273000
C.....502 FINDS THE FIRST LETTER OF EACH WORD 03274000
502 IF (CARDIC .LE. Z .AND. CARDIC .GE. AA .AND. CARDIC .NE. KEY028) 03275000
* GO TO 504 03276000
IF (CARDIC .EQ. CPAREN) ASSIGN 516 TO BR4 03277000
GO TO 501 03278000
C.....516 SKIPS THE REST OF EACH COMMAND WORD (CPAREN ALREADY SKIPPED) 03279000

```

516	IF (CARDIC .EQ. BLANK .OR. CARDIC .EQ. KEY028) ASSIGN 502 TO BR4	03280000
	GO TO 501	03281000
C.....	503 PLACES EACH VALID WORD INTO 'WORD'	03282000
503	IF (CARDIC .LE. Z .AND. CARDIC .GE. AA .AND. CARDIC .NE. KEY028)	03283000
	* GO TO 506	03284000
	IF (CARDIC - CSIGN) 507,501,507	03285000
C.....	INTERNAL CENTS SIGN ALLOWED, AS IN MC%COY, BUT NOT LISTED	03286000
504	ASSIGN 503 TO BR4	03287000
	LENGTH = 1	03288000
	GO TO 505	03289000
526	READ (1,1000,END=527) WORD2	03290000
1000	FORMAT (55X,A2,40A1)	03291000
	CICNT = CICNT + 1	03292000
	GO TO 521	03293000
506	LENGTH = LENGTH + 1	03294000
505	WORD(LENGTH) = CARDIC	03295000
	IF (LENGTH .LT. 40) GO TO 501	03296000
507	ASSIGN 502 TO BR4	03297000
	WCNT = WCNT + 1	03298000
	IF (LENGTH .LE. 2) GO TO 501	03299000
C.....	WORDS LISTED ARE 3 TO 40 CHARACTERS LONG AND CONTAIN LETTERS ONLY	03300000
C		03301000
C.....	WORD MATCHING AND ALPHABETIZING ROUTINE	03302000
C.....	CODE-DEPENDENT	03303000
	IF (WORD(2) .LE. R) GO TO 518	03304000
	WORD(2) = WORD(2) - 3840	03305000
	GO TO 519	03306000
518	IF (WORD(2) .GT. II) WORD(2) = WORD(2) - 1792	03307000
519	IF (WORD(1) .LE. R) GO TO 522	03308000
	WORD(1) = WORD(1) - 3840	03309000
	GO TO 521	03310000
522	IF (WORD(1) .GT. II) WORD(1) = WORD(1) - 1792	03311000
521	OVADDR = ((WORD(1) * 26) + WORD(2) + 433984) / 256	03312000
	VADDR = WL(OVADDR)	03313000
	GO TO 515	03314000
512	OVADDR = VADDR	03315000
	VADDR = NVADDR	03316000
515	IF (VADDR .EQ. 0) GO TO 510	03317000
	NVADDR = WL(VADDR)	03318000
	COMPAR = WLL(VADDR)	03319000
	IF (COMPAR .GT. LENGTH) COMPAR = LENGTH	03320000
	DO 511 XI = 3, COMPAR	03321000
	IF (WL(VADDR+XI) - WORD(XI)) 512,511,510	03322000
511	CONTINUE	03323000
	IF (WLL(VADDR) - LENGTH) 512,514,510	03324000
514	WLC(VADDR) = WLC(VADDR) + 1	03325000
C.....	THE WORD MATCHES THE COMPARAND	03326000
	GO TO 501	03327000
C.....	WL TABLE CANNOT CONTAIN PRESENT WORD	03328000
508	IF (CIINC .EQ. 0) CIINC = CICNT + 5	03329000
	IF (SECTN .GT. 1) GO TO 501	03330000
	WRITE (1,1000) WORD2	03331000
	GO TO 501	03332000
510	IF (NEXT + LENGTH .GE. LIMIT) GO TO 508	03333000
	WL(OVADDR) = NEXT	03334000
	WL(NEXT) = VADDR	03335000
	WLL(NEXT) = LENGTH	03336000
	WLC(NEXT) = 1	03337000
	DO 513 XI = 3, LENGTH	03338000
	WL(NEXT+XI) = WORD(XI)	03339000

138

513	CONTINUE	03340000
	NEXT = NEXT + LENGTH + 1	03341000
501	CONTINUE	03342000
	GO TO 500	03343000
C		03344000
C....	ROUTINE FOR WRITING RESULTS	03345000
C....	CODE-DEPENDENT	03346000
509	REWIND 3	03347000
	REWIND IREAD	03348000
	ENDFILE 1	03349000
527	WORD(1) = 0	03350000
	DO 551 PTR = 1, 676	03351000
	VADDR = WL(PTR)	03352000
	IF (VADDR .EQ. 0) GO TO 551	03353000
	XI = WORD(1)	03354000
	WORD(1) = AA + ((PTR - 1) / 26) * 256	03355000
	IF (WORD(1) .GT. II) WORD(1) = WORD(1) + 1792	03356000
	WORD(2) = AA + (MOD(PTR-1,26) * 256)	03357000
	IF (WORD(2) .GT. IT) WORD(2) = WORD(2) + 1792	03358000
	IF (WORD(2) .GT. R) WORD(2) = WORD(2) + 2048	03359000
	IF (WORD(1) .GT. R) WORD(1) = WORD(1) + 2048	03360000
	IF (XI .EQ. WORD(1)) GO TO 554	03361000
	IF (COUNT .LE. 0) GO TO 554	03362000
	WRITE (3,1004) WORD(1)	03363000
1004	FORMAT (')LW3H ',A1,')MLL',67X)	03364000
554	IF (WLC(VADDR) .LE. 0) GO TO 528	03365000
	LENGTH = WLL(VADDR)	03366000
	UWORD = UWORD + 1	03367000
	DO 550 XJ = 3, LENGTH	03368000
	WORD(XJ) = WL(VADDR+XJ)	03369000
550	CONTINUE	03370000
552	LENGTH = LENGTH + 1	03371000
C....	WORD(41) IS INITIALIZED IN DATA STATEMENT TO BLANK VIA WORD1	03372000
	IF (WORD(LENGTH) .EQ. BLANK) GO TO 553	03373000
	WORD(LENGTH) = BLANK	03374000
	GO TO 552	03375000
553	WRITE (3,1003) WORD,WLC(VADDR)	03376000
1003	FORMAT (')H ',40A1,I4,')HL',28X)	03377000
528	VADDR = WL(VADDR)	03378000
	IF (VADDR .NE. 0) GO TO 554	03379000
551	CONTINUE	03380000
	IF (CIINC .EQ. 0) GO TO 524	03381000
	ASSIGN 526 TO BR4	03382000
	SECTN = SECTN + 1	03383000
	WRITE (3,1005) SECTN	03384000
1005	FORMAT (')SU SECTION',I4,' :)UC',59X)	03385000
	DO 517 XI = 1, 338	03386000
517	WLX(XI) = 0	03387000
	NEXT = 677	03388000
	IF (SECTN .EQ. 2) GO TO 529	03389000
	CIINC = CIINC - 5	03390000
	DO 523 XI = CIINC, CICNT	03391000
	BACKSPACE 1	03392000
523	CONTINUE	03393000
	GO TO 520	03394000
524	WRITE (3,1002) WCNT,UWORD	03395000
1002	FORMAT (')CH',I7,' WORDS/STRINGS)HLH',I6,' WORDS LISTED)E',28X)	03396000
C		03397000
C....	ROUTINE FOR RE-INITIALIZING PARAMETERS	03398000
	ENDFILE 3	03399000

```

REWIND 3
REWIND 1
DO 559 XI = 3, 184
559 WLX(XI) = 0
C.....SETS VARIABLES IN /C/ FROM 'ITEXT' THRU 'RSIDE' TO ZERO
CICNT = 0
CIINC = 0
DICT = -10
FIELD2 = 52
C.....'CARD FIELD THRU 52' IS IN EFFECT
IREAD = 3
IWRITE = 6
NODOC = 0
ALT = 1
BLNKLN = 1
CCGCNT = 0
CHARCO = 20
COLPAG = 6
CSEP = 1
FLN = 5
ICINC = 1
INDARH(1) = 2
C.....I.E., FIRST INDENT IS 2,0
LINPAG = 59
TLLN = LINPAG
LINSIZ = 132
LN2 = 2
LSIDE = 132
LTITLE = 1
MASK1 = 0
MYPAGE = 1
NSYM = 3
PAGENO = 1
REPTTL = 10
SPACNG = 1
TEXEND = 7788
TFLN = 1
TWOUP = 1
TXTLNE = 3
DO 555 I = 1, TEXEND
555 PAGDUM(I) = BLANK
DO 556 I = 1, 26
556 PAGDUM(I+53) = TITTXT(I)
RETURN
END

C
C
C
SUBROUTINE LOC1
IMPLICIT INTEGER*4 (A - Z)
DIMENSION WLX(352)
INTEGER*2 LL,BUFFER(80),RECORD,LCHAR,PLUS,LTR1,SLASH,ASTRSK,LCZ,
* LTR2,AA,ZZ,R,WL(8430),WLL(8429),WLR(8428),WLS(8426),$,XKY(2),II
C.....IN LOC1, LOC2, & LOC3 WL CAN BE DIMENSIONED FOR ALL /C/. LIMIT = DM
INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS,
* PERIOD,KEY028,LOCATE,CARDIC,ONLIST,DOLLAR
INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN
COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE,
* REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALD,BFOUND,CICNT1,CICNT,
* CIINC,DICT,NEXT,HIT,

```

```

03400000
03401000
03402000
03403000
03404000
03405000
03406000
03407000
03408000
03409000
03410000
03411000
03412000
03413000
03414000
03415000
03416000
03417000
03418000
03419000
03420000
03421000
03422000
03423000
03424000
03425000
03426000
03427000
03428000
03429000
03430000
03431000
03432000
03433000
03434000
03435000
03436000
03437000
03438000
03439000
03440000
03441000
03442000
03443000
03444000
03445000
03446000
03447000
03448000
03449000
03450000
03451000
03452000
03453000
03454000
03455000
03456000
03457000
03458000
03459000

```

```

* ARRAY1(3), 03460000
* LIST, COPIES, SPCHAR(42), BLANK, NUM(10), SCWORD, CARDIC, 03461000
* HYPHEN, LOWCAS, PERIOD, KEY028, LOCATE, OHLIST, DOLLAR 03462000
COMMON /C/ ALT, BLNKLN, ITEXT, LINPAG, NYPAGE, RIVER, HYPTRX, WPT, SUND, TW03463000
*, TWOUP, I, CPSW, ISPT, ASIS, CCHAR, CHARCO, NEWH, SAVCCC, KEEP, WPTX, LWI, N, 03464000
* LINSIZ, NSYM, SPACNG, TXTLNE, K, LSTBL, AUTO, PFLN, NOGO, NAME, SWK028, IIO, 03465000
* CCCNT, COLPAG, IVALUE, LINSZ, PAGENO, START, UNDERL, J, CONST, AUTOTB, SWX, 03466000
* JUNK, ID, PDUM, REM, IER700, END, CENTER, CSEP, INDENT, TEXEND, TLLN, CLEAN, 03467000
* LN2, REPTTL, IC, CWIDTH, USEWS1, LTITLE, CU(8), TAB2(7), INDP(4), 03468000
* INDARK(8), FLN, ICINC, PARA, TFLN, USTART, FCH, SWWPT, ENDL, EMX, PIVOT, 03469000
* INDEX, TABSEQ, CHAR, ENDF, LINEX(67), WORDS, LNTW, CS, ENDSAV, ID1, LINEW, 03470000
* WANT, GAPS, WANTIN, WSEPD, LSIDE, RSIDE, SWEW(68), CARD(40), 03471000
* PAGDUM(7788), SRT(99), COLBEG(8), CHRFIN(99) 03472000
DATA PLUS/'+' , R/'R' , II/'I' , LIMIT/8430/, NXTONE/704/, $/'$' , 03473000
* SLASH/'/' , ASTRSK/'*' , LCZ/ZA940/ 03474000
EQUIVALENCE (LL, NUM(4)), (AA, NUM(1)), (ZZ, NUM(6)), (WL(1), ALT), 03475000
* (LTR1, BUFFER(1)), (LTR2, BUFFER(2)), (WL(1), WLX(1)), (WLL(1), WL(2)), 03476000
* (WLR(1), WL(3)), (WLS(1), WL(5)), (KKX(1), XKIABS) 03477000
C.....THIS SUBROUTINE IS VERY CODE-SENSITIVE. 03478000
REWIND 1 03479000
REWIND 3 03480000
DO 102 XI = 1, 352 03481000
102 WLX(XI) = 0 03482000
C 03483000
C.....READ EACH $LOCATE STRING, REDUCE IT, AND PLACE IT IN 'WL': 03484000
C 03485000
DO 100 XI = 1, LOCATE 03486000
101 READ (1,1000) RECORD, BUFFER 03487000
1000 FORMAT (A1,16X,80A1) 03488000
IF (RECORD.NE. LL) GO TO 101 03489000
C.....BUFFER IS NOW LOADED WITH A RECORD TO BE $LOCATED. NOW REDUCE IT. 03490000
LASTCB = 1 03491000
XK = 0 03492000
C.....ELIMINATE EXCESS BLANKS FROM PHRASE TO BE $LOCATED. 03493000
DO 103 XI = 1, 80 03494000
LCHAR = BUFFER(XI) 03495000
IF (LCHAR.NE. BLANK) GO TO 104 03496000
LASTCB = LASTCB + 1 03497000
IF (LASTCB = 1) 103, 105, 103 03498000
104 IF (LCHAR.GT. 0 .AND. LCHAR.NE. HYPHEN .AND. LCHAR.NE. SLASH 03499000
* .AND. LCHAR.NE. ASTRSK .AND. LCHAR.NE. PLUS .AND. LCHAR.NE. $) 03500000
* GO TO 103 03501000
C.....NON-ALPHAMERIC NOT B-/*+$ IS ELIMINATED FROM PHRASE TO BE $LOCATED. 03502000
LASTCB = 0 03503000
IF (LCHAR.LE. LCZ) LCHAR = LCHAR + LOWCAS 03504000
105 XK = XK + 1 03505000
BUFFER(XK) = LCHAR 03506000
103 CONTINUE 03507000
IF (BUFFER(XK).EQ. BLANK) XK = XK - 1 03508000
IF (BUFFER(XK).EQ. PLUS) XK = 1 - XK 03509000
IF (XK.EQ. 0) GO TO 100 03510000
C.....ASSUMES BUFFER(0).NE. BLANK, ELIMINATES LONE '+' AND 80 BLANKS. 03511000
C...../XK/ = LENGTH OF PHRASE. XK.LT. 0 IF PHRASE IS A PREFIX. 03512000
XKIABS = IABS(XK) 03513000
IF (LTR1.LT. AA .OR. LTR1.GT. ZZ .OR. LTR2.LT. AA .OR. LTR2 03514000
* .GT. ZZ) GO TO 107 03515000
CHAR1 = 3 03516000
IF (LTR2.LE. R) GO TO 108 03517000
LTR2 = LTR2 - 3840 03518000
GO TO 109 03519000

```



```

108 IF (LTR2 .GT. II) LTR2 = LTR2 - 1792      03520000
109 IF (LTR1 .LE. B) GO TO 110                03521000
      LTR1 = LTR1 - 3840                      03522000
      GO TO 111                              03523000
110 IF (LTR1 .GT. II) LTR1 = LTR1 - 1792      03524000
111 OVADDR = ((LTR1 * 27) + LTR2 + 450304) / 256 03525000
117 VADDR = WL(OVADDR)                       03526000
      GO TO 113                              03527000
C.....IF 1ST OR 2ND CHARACTER IS A LETTER, USE IT FOR PLACEMENT POS'N. 03528000
107 CHAR1 = 1                               03529000
      IF (LTR1 .LT. AA .OR. LTR1 .GT. ZZ) GO TO 120 03530000
      JUNKX = LTR1                           03531000
      GO TO 121                              03532000
120 IF (LTR2 .LT. AA .OR. LTR2 .GT. ZZ) GO TO 124 03533000
      JUNKX = LTR2                           03534000
121 IF (JUNKX .LE. B) GO TO 122              03535000
      JUNKX = JUNKX - 3840                   03536000
      GO TO 123                              03537000
122 IF (JUNKX .GT. II) JUNKX = JUNKX - 1792   03538000
123 OVADDR = (JUNKX * 27 + 433984) / 256      03539000
      IF (XK .EQ. (-1)) XK = 1              03540000
C.....A SINGLE LETTER MAY NOT BE FOLLOWED BY A TERMINAL '+'. 03541000
      GO TO 117                              03542000
C.....NEITHER 1ST NOR 2ND CHARACTER IS A LETTER. 03543000
124 OVADDR = 703                            03544000
      GO TO 117                              03545000
112 OVADDR = VADDR                           03546000
      VADDR = NVADDR                         03547000
113 IF (VADDR .EQ. 0) GO TO 114              03548000
      NVADDR = WL(VADDR)                    03549000
      JUNKX = WLL(VADDR)                    03550000
      COMPAR = IABS(JUNKX)                  03551000
      IF (COMPAR .GT. XKIABS) COMPAR = XKIABS 03552000
      IF (COMPAR .LT. CHAR1) GO TO 118       03553000
      DO 115 XJ = CHAR1, COMPAR             03554000
      IF (WLS(VADDR-CHAR1+XJ) - BUFFER(XJ)) 112,115,114 03555000
115 CONTINUE                                03556000
118 IF (IABS(JUNKX) - XKIABS) 112,125,114    03557000
125 IF (XK .LT. 0) WLL(VADDR) = XK          03558000
      GO TO 100                              03559000
114 IF (NXTONE + XKIABS + 5 - CHAR1 .LE. LIMIT) GO TO 106 03560000
      WRITE (6,1003) XI                     03561000
1003 FORMAT ('1'///// ' TABLE O/P AT $LOCATE STRING NO.',15) 03562000
      GO TO 303                              03563000
106 WL(OVADDR) = NXTONE                      03564000
      WL(NXTONE) = VADDR                    03565000
      WLL(NXTONE) = XK                      03566000
      WLR(NXTONE) = 0                       03567000
      IF (XKIABS .LT. CHAR1) GO TO 119       03568000
      DO 116 XJ = CHAR1, XKIABS             03569000
      WLS(NXTONE-CHAR1+XJ) = BUFFER(XJ)     03570000
116 CONTINUE                                03571000
119 NXTONE = NXTONE + XKIABS + 5 - CHAR1     03572000
100 CONTINUE                                03573000
C                                              03574000
C.....WRITE REDUCED, FILTERED, AND SORTED $LOCATE STRINGS ONTO DRN 1. 03575000
C                                              03576000
303 CHAR1 = 3                               03577000
      REWIND 1                              03578000
      DO 300 XI = 1, 703                    03579000

```

```

VADDR = WL(XI)
IF (VADDR .EQ. 0) GO TO 300
LTR1 = AA + (((XI - 1) / 27) * 256)
IF (LTR1 .GT. II) LTR1 = LTR1 + 1792
IF (LTR1 .GT. R) LTR1 = LTR1 + 2048
JUNKX = MOD(XI+25,27)
IF (JUNKX .NE. 26) GO TO 305
CHAR1 = 1
GO TO 301
305 CHAR1 = 3
LTR2 = AA + JUNKX * 256
IF (LTR2 .GT. II) LTR2 = LTR2 + 1792
IF (LTR2 .GT. R) LTR2 = LTR2 + 2048
301 XK = WLL(VADDR)
XKIABS = IABS(XK)
IF (XKIABS .LT. CHAR1) GO TO 306
DO 302 XJ = CHAR1, XKIABS
BUFFER(XJ) = WLS(VADDR-CHAR1+XJ)
302 CONTINUE
306 IF (XK .GT. 0) GO TO 304
XKIABS = XKIABS + 1
BUFFER(XKIABS) = PLUS
304 WRITE (1,1006) XK(2),BUFFER
1006 FORMAT (15X,A2,80A1)
VADDR = WL(VADDR)
IF (VADDR .NE. 0) GO TO 301
300 CONTINUE
ENDFILE 1
REWIND 1
RETURN
END

C
C
C
SHROUTINE LOC2
IMPLICIT INTEGER*4 (A - Z)
DIMENSION COLLE3(20)
INTEGER*2 BUFFER(80),PLUS,AA,ZZ,R,WL(8430),WLL(8429),WLR(8428),
* WLP(8427),WLS(8426),$,II,CR(81),FLIST(41),CRW1,CRW2,WORDCI(41),
* COLLE0(39),COLLE1(38),COLLE2(37),COLLE4(40),SLASH,ASTRSK,30(3),E,
* V,LCZ
INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS,
* PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR
INTEGER*2 MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK
INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN
COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE,
* REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT,
* CIINC,DICT,NEXT,HIT,
* ARRAY1(3),
* LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC,
* HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR
COMMON /B/ FIELD1,FIELD2,FIELD3,SPOP,CP,LB,PER,KEEPSV(2),
* UPPER,UP1,CAP,FIRST,MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK
COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,HYPAGE,RIVER,HYPTRX,WPT,SUND,TWO
* TWOUP,I,CPSW,ISPT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTK,LWI,N,
* LINSIZ,NSYM,SPACNG,TXTLNE,K,LSTBL,AUTO,FPLN,NOGO,NAME,SWK028,IU,
* CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTJTB,SWX,
* JUNK,ID,PDU,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN,
* LN2,REPTTL,IC,CWIDTH,USEWS1,LTITLE,CU(8),TAB2(7),INDP(4),
* INDARR(8),FLN,ICINC,PARA,TPLN,USTART,FCM,SWWPT,ENDL,EWX,PIVOT,

```

```

03580000
03581000
03582000
03583000
03584000
03585000
03586000
03587000
03588000
03589000
03590000
03591000
03592000
03593000
03594000
03595000
03596000
03597000
03598000
03599000
03600000
03601000
03602000
03603000
03604000
03605000
03606000
03607000
03608000
03609000
03610000
03611000
03612000
03613000
03614000
03615000
03616000
03617000
03618000
03619000
03620000
03621000
03622000
03623000
03624000
03625000
03626000
03627000
03628000
03629000
03630000
03631000
03632000
03633000
03634000
03635000
03636000
03637000
03638000
03639000

```

```

* INDEX,TABSEQ,CHAR,ENDF,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX, 03640000
* WANT,GAPS,WANTIN,WSEPD,LSIDE,RSIDE,SWEW(68),CARD(40), 03641000
* PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99) 03642000
C....HLIMIT = (WL DIMENSION) / 3 IN LOC2 & LOC3 03643000
DATA PLUS/'+' ,R/'R' ,II/'I' ,$/ '$' ,HLIMIT/2810/,CI/0/, 03644000
* HITPT/1/,HITREC/1/,SCOUNT/0/,CRPT/1/,ASISX/0/,GO/'G' , 'O' ,0/, 03645000
* SLASH/'/' ,ASTRSK/'*' ,LCZ/ZA940/,NXTWRD/1/,WORDNO/0/ 03646000
EQUIVALENCE (WL(1),ALT), (WLL(1),WL(2)), (WLR(1),WL(3)), (WLP(1), 03647000
* WL(4)), (WLS(1),WL(5)), (COLLE0(1),COLLE4(2)), (COLLE1(1),COLLE0(2)) 03648000
*, (COLLE2(1),COLLE0(3)), (COLLE4(1),COLLE3(1)), (AA,NUM(1)), 03649000
* (ZZ,NUM(6)), (E,NUM(2)), (V,NUM(5)) 03650000
C....THIS SUBROUTINE IS VERY CODE-SENSITIVE. 03651000
C 03652000
C....PASS INPUT ACROSS STRINGS TO BE $LOCATED. 03653000
C 03654000
C....LOC1 REWINDS DRNS 1 AND 3. ENDJOB REWINDS IREAD. 03655000
C....205 READS CONTROL CARDS, ET AL, UNTIL 'GO' IS POUND. 03656000
205 ASSIGN 201 TO BR5 03657000
CCOL = 80 03658000
FLIST(1) = 1 03659000
EOF = 0 03660000
211 READ (IREAD,1004,END=207) BUFFER 03661000
CI = CI + 1 03662000
C....IS THIS RECORD THE 'GO' CARD (OR AN EQUIVALENT), QUESTION MARK. 03663000
XJ = 0 03664000
DO 215 XI = 1, 80 03665000
CARDIC = BUFFER(XI) 03666000
IF (CARDIC .EQ. BLANK) GO TO 215 03667000
XJ = XJ + 1 03668000
IF (CARDIC .NE. GO(XJ)) GO TO 211 03669000
215 CONTINUE 03670000
C....YES: THIS CONTROL CARD CONTAINS EITHER 'GO', 'G', OR ALL BLANKS. 03671000
NXTWRD = 0 03672000
WORDNO = 0 03673000
C....209 IS THE DYNAMIC ENTRY POINT. 03674000
209 NXTWRD = NXTWRD + 1 03675000
IF (NXTWRD .GT. 41) NXTWRD = 1 03676000
CR2 = FLIST(NXTWRD) 03677000
IF (EOF .NE. 0) GO TO 214 03678000
200 CCOL = CCOL + 1 03679000
IF (CCOL .LE. 80) GO TO 206 03680000
210 CCOL = 1 03681000
READ (IREAD,1004) BUFFER 03682000
1004 FORMAT (80A1) 03683000
CI = CI + 1 03684000
IF (ASISX .EQ. 0) GO TO 206 03685000
ASSIGN 201 TO BR5 03686000
IF (BUFFER(1) .NE. CPAREN) GO TO 206 03687000
CCOL = 2 03688000
ASISX = 0 03689000
206 CARDIC = BUFFER(CCOL) 03690000
IF (CARDIC .EQ. KEY028) CARDIC = BLANK 03691000
GO TO BR5, (201,203,204) 03692000
C....203 SCANS COMMAND WORDS FOR V, E, AND A - OTHERWISE IGNORES THEM. 03693000
203 IF (ASISX .NE. 0) GO TO 213 03694000
IF (CARDIC .EQ. V .OR. CARDIC .EQ. E) GO TO 212 03695000
IF (CARDIC .NE. AA) GO TO 213 03696000
ASISX = 10 03697000
ASSIGN 201 TO BR5 03698000
GO TO 210 03699000

```

```

213 IF (CARDIC .EQ. BLANK) ASSIGN 201 TO BR5 03700000
GO TO 200 03701000
C.....201 HANDLES WORD BEGINNINGS. 03702000
201 IF (CARDIC .NE. CPAREN) GO TO 202 03703000
ASSIGN 203 TO BR5 03704000
GO TO 200 03705000
202 IF (CARDIC .EQ. BLANK) GO TO 200 03706000
C.....ELIMINATE EXCESS BLANKS. 03707000
WORDNO = WORDNO + 1 03708000
IF (WORDNO .GT. 41) WORDNO = 1 03709000
FLIST(WORDNO) = CRPT 03710000
WORDCI(WORDNO) = CI 03711000
ASSIGN 204 TO BR5 03712000
C.....204 INSERTS WORDS INTO THE COMPARE REGISTER (CR). 03713000
204 IF (CARDIC .GT. 0 .AND. CARDIC .NE. BLANK .AND. CARDIC .NE. HYPHEN 03714000
* .AND. CARDIC .NE. SLASH .AND. CARDIC .NE. ASTRSK .AND. CARDIC
* .NE. PLUS .AND. CARDIC .NE. $) GO TO 200 03715000
C.....DELETE NON-ALPHAMERICS NOT B-/*+$ (TO AVOID PUNCTUATION MARKS). 03716000
IF (CARDIC .LE. LCZ) CARDIC = CARDIC + LOWCAS 03717000
CR(CRPT) = CARDIC 03718000
CRPT = CRPT + 1 03719000
IF (CRPT .GT. 81) CRPT = 1 03720000
IF (CARDIC .EQ. BLANK) ASSIGN 201 TO BR5 03721000
IF (CRPT - CR2) 200,600,200 03722000
C.....) V OR ) E ENCOUNTERED IN COMMAND WORD. 03723000
212 EOF = -10 03724000
GO TO 214 03725000
C.....END-OF-FILE ENCOUNTERED ON TEXT TAPE. 03726000
207 EOF = 10 03727000
REWIND IREAD 03728000
214 IF (NXTWRD .NE. MOD(WORDNO,41) + 1) GO TO 208 03729000
IF (EOF) 205,612,612 03730000
208 CR(CRPT) = BLANK 03731000
CRPT = CRPT + 1 03732000
IF (CRPT .GT. 81) CRPT = 1 03733000
IF (CRPT .NE. CR2) GO TO 208 03734000
C 03735000
C.....COMPARE WORD STARTING AT CR(FLIST(NXTWRD)) AGAINST $LOCATE STRINGS. 03736000
C 03737000
600 W1 = FLIST(NXTWRD) 03738000
W2 = W1 + 1 03739000
IF (W2 .GT. 81) W2 = 1 03740000
CRW1 = CR(W1) 03741000
CRW2 = CR(W2) 03742000
IF (CRW1 .GE. AA .AND. CRW1 .LE. ZZ .AND. CRW2 .GE. AA .AND. CRW2 03743000
* .LE. ZZ) GO TO 601 03744000
CHAR1 = 1 03745000
IF (CRW1 .GE. AA .AND. CRW1 .LE. ZZ) GO TO 613 03746000
IF (CRW2 .LT. AA .OR. CRW2 .GT. ZZ) GO TO 614 03747000
CRW1 = CRW2 03748000
613 CRW2 = -16320 03749000
C.....'A' - 256 = -16320. 03750000
C.....WL SUBSCRIPT = (27*CRW1-27*'A'+256)/256 = (27*CRW1+433984)/256. 03751000
GO TO 603 03752000
614 VADDR = WL(703) 03753000
GO TO 607 03754000
601 CHAR1 = 3 03755000
IF (CRW2 .LE. B) GO TO 602 03756000
CRW2 = CRW2 - 3840 03757000
GO TO 603 03758000
03759000

```

```

602 IF (CRW2 .GT. II) CRW2 = CRW2 - 1792 03760000
603 IF (CRW1 .LE. R) GO TO 604 03761000
    CRW1 = CRW1 - 3840 03762000
    GO TO 606 03763000
604 IF (CRW1 .GT. II) CRW1 = CRW1 - 1792 03764000
606 VADDR = WL(((CRW1 * 27) + CRW2 + 450304) / 256) 03765000
C.....WL SUBSCRIPT = (CRW1 * 27 + CRW2 - 28 * 'A' + 512) / 256. 03766000
    GO TO 607 03767000
608 VADDR = NVADDR 03768000
607 IF (VADDR .EQ. 0) GO TO 209 03769000
    NVADDR = WL(VADDR) 03770000
    XK = WLL(VADDR) 03771000
    W3 = W1 + CHAR1 - 1 03772000
    IF (W3 .GT. 81) W3 = W3 - 81 03773000
    XKIABS = IABS(XK) 03774000
    IF (XKIABS .LT. CHAR1) GO TO 605 03775000
    DO 609 XJ = CHAR1, XKIABS 03776000
    IF (WLS(VADDR-CHAR1+XJ) - CR(W3)) 608,610,209 03777000
610 W3 = W3 + 1 03778000
    IF (W3 .GT. 81) W3 = 1 03779000
609 CONTINUE 03780000
605 IF (CR(W3) .NE. BLANK .AND. XK .GT. 0) GO TO 608 03781000
C.....A HIT HAS BEEN MADE. 03782000
    HIT = HIT + 1 03783000
    COLLE0(HITPT) = WORDCI(NXTWRD) 03784000
    COLLE1(HITPT) = WLR(VADDR) 03785000
    COLLE2(HITPT) = WLP(VADDR) 03786000
    WLR(VADDR) = HITREC 03787000
    WLP(VADDR) = HITPT 03788000
    HITPT = HITPT + 3 03789000
    IF (HITPT .LT. 40) GO TO 611 03790000
    HITPT = 1 03791000
    WRITE (3,1002) COLLE3 03792000
1002 FORMAT (20A4) 03793000
    HITREC = HITREC + 1 03794000
611 IF (HIT .LT. HLIMIT) GO TO 608 03795000
    HIT = 0 03796000
    HITREC = 1 03797000
612 IF (HITPT .EQ. 1) GO TO 500 03798000
    WRITE (3,1002) COLLE3 03799000
    HITPT = 1 03800000
C 03801000
C.....WRITE THE INDEX RECORDS ONTO DRN 3. ALSO ZERO OUT WLR(OVADDR). 03802000
C 03803000
500 DO 502 XJ = 1, 703 03804000
    OVADDR = WL(XJ) 03805000
C.....DO NOT USE VADDR OR NVADDR WHICH HAVE VALUES NEEDED IF SECTN .GT. 103806000
501 IF (OVADDR .EQ. 0) GO TO 502 03807000
    COLLE4(HITPT) = WLR(OVADDR) 03808000
    COLLE0(HITPT) = WLP(OVADDR) 03809000
    WLR(OVADDR) = 0 03810000
    OVADDR = WL(OVADDR) 03811000
    HITPT = HITPT + 2 03812000
    IF (HITPT .LT. 41) GO TO 501 03813000
    HITPT = 1 03814000
    WRITE (3,1002) COLLE3 03815000
    GO TO 501 03816000
502 CONTINUE 03817000
    IF (HITPT .EQ. 1) GO TO 503 03818000
    WRITE (3,1002) COLLE3 03819000

```

146

```

HITPT = 1                                03820000
503 SCOUNT = SCOUNT + 1                    03821000
    IF (EOF .NE. 10 .OR. NXTWRD .NE. MOD(WORDNO,41) + 1) GO TO 608 03822000
    ENDFILE 3                              03823000
    REWIND 3                               03824000
    HIT = (SCOUNT - 1) * HLIMIT + HIT      03825000
    RETURN                                 03826000
    END                                    03827000
C                                          03828000
C                                          03829000
C                                          03830000
SUBROUTINE LOC3                           03831000
IMPLICIT INTEGER*4 (A - Z)                03832000
DIMENSION COLLE3(20)                      03833000
INTEGER*2 BUFFER(80),WL(8430),WLL(8429),WLR(8428),XKX(2), 03834000
* COLLE0(39),COLLE4(40),LHREC(3),LHPOS(3),CIHIT(21)      03835000
INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS, 03836000
* PERIOD,KEY020,LOCATE,CARDIC,OMLIST,DOLLAR 03837000
INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN 03838000
COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE, 03839000
* REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT, 03840000
* CLINC,DICT,NEXT,HIT, 03841000
* ARRAY1(3), 03842000
* LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC, 03843000
* HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR 03844000
COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,MYPAGE,RIVER,HYPTRX,WPT,SUND,TWO 03845000
* TWOUP,1,CPSW,ISPOT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTX,LWI,N, 03846000
* LINSIZ,NSYM,SPACNG,TXTLNE,K,LSTBL,AUTO,FPLN,NOGO,NAME,SWK028,IU, 03847000
* CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTJTB,SWX, 03848000
* JUNK,ID,PDUM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN, 03849000
* LN2,REPTTL,IC,CWIDTH,USEWS1,LTITLE,CU(8),TAB2(7),INDP(4), 03850000
* INDARR(8),FLN,ICINC,PARA,TFLN,USTART,FCM,SWWPT,ENDL,EWX,PIVOT, 03851000
* INDEX,TABSEQ,CHAR,ENDF,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX, 03852000
* WANT,GAPS,WANTIN,WSEPD,LSIDE,RSIDE,SWEW(68),CARD(40), 03853000
* PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99) 03854000
DATA HLIMIT/2810/,SECTN/0/,LNZ/57/,PGE/0/ 03855000
EQUIVALENCE (WL(1),ALT),(WLL(1),WL(2)),(WLR(1),WL(3)),(XKX(1), 03856000
* XKIABS),(COLLE0(1),COLLE4(2)),(COLLE4(1),COLLEJ(1)) 03857000
C                                          03858000
C.....READ HIT COLLECTION RECORDS FROM DRN 3 FOR THE (SECTN)TH SECTION. 03859000
C                                          03860000
504 SECTN = SECTN + 1                      03861000
    JUNKX = HIT                            03862000
    IF (HIT .GT. HLIMIT) JUNKX = HLIMIT    03863000
    HIT = HIT - JUNKX                      03864000
    JUNKX = JUNKX + 3                      03865000
    IF (JUNKX .NE. 0) READ (3,1005) (WL(X1), X1 = 1, JUNKX) 03866000
1005 FORMAT (2X,39A2)                     03867000
C                                          03868000
C.....WRITE THE FORMATED FINAL RESULTS ONTO DRN 6. 03869000
C                                          03870000
    XKX(1) = 0                             03871000
C.....READ AN INDEX RECORD (20 ENTRIES PER RECORD). 03872000
505 READ (3,1002,END=517) COLLE3          03873000
    DO 515 HITPT = 1, 39, 2                03874000
    READ (1,1006,END=516) XKX(2),BUFFER 03875000
    ASSIGN 507 TO BR5                       03876000
    GO TO 700                               03877000
1002 FORMAT (20A4)                         03878000
1006 FORMAT (15X,A2,80A1)                  03879000

```

```

507  WRITE (6,1008) (BUFFER(XJ), XJ = 1, XKIABS)
1008  FORMAT (' ',80A1)
      LHREC(1) = COLLE4(HITPT)
      IF (LHREC(1) .GT. 0) GO TO 511
      WRITE (6,1009)
1009  FORMAT ('+',83X,'NOT FOUND')
      GO TO 515
511   LHPOS(1) = COLLE0(HITPT)
      LHREC(3) = 0
      I1 = 1
      I2 = 2
      I3 = 3
509   PRCHIT = 39 * (LHREC(I1) - 1) + LHPOS(I1)
      LHREC(I1) = PRCHIT
      LHREC(I2) = WLL(PRCHIT)
      LHPOS(I2) = WLR(PRCHIT)
      WLL(PRCHIT) = LHREC(I3)
      IF (LHREC(I2) .EQ. 0) GO TO 510
      JUNKX = I1
      I1 = I2
      I2 = I3
      I3 = JUNKX
      GO TO 509
C.....HIT CHAIN FOR THIS $LOCATE STRING NOW POINTS FORWARD.
C.....PRINT ITS CARD IMAGE NUMBERS.
513   WRITE (6,1010) CIHIT
1010  FORMAT (7X,21I6)
510   DO 512 XI = 1, 21
      CIHIT(XI) = WL(PRCHIT)
      PRCHIT = WLL(PRCHIT)
      IF (PRCHIT .EQ. 0) GO TO 514
512   CONTINUE
      ASSIGN 513 TO BR5
      GO TO 700
514   ASSIGN 506 TO BR5
C.....PAGINATION ROUTINE.
700   IF (LNZ .LT. 57) GO TO 701
      PGE = PGE + 1
      WRITE (6,1007) SECTN,PGE
1007  FORMAT ('1',29X,71HCARD IMAGE NUMBERS WHERE INDICATED WORDS/PHRASE
      *S ARE LOCATED -- SECTION,I3,16X,'PAGE',I3//)
      LNZ = 0
701   LNZ = LNZ + 1
      GO TO BR5, (507,513,506)
C.....END OF PAGINATION ROUTINE.
506   WRITE (6,1010) (CIHIT(XJ), XJ = 1, XI)
515   CONTINUE
      GO TO 505
516   IF (HIT .EQ. 0) GO TO 517
      REWIND 1
      IF (HITPT .EQ. 1) BACKSPACE 3
      LNZ = 57
      GO TO 504
517   LOCATE = 0
      RETURN
      END
C
C
C
      SUBROUTINE EDITOR

```

```

03880000
03881000
03882000
03883000
03884000
03885000
03886000
03887000
03888000
03889000
03890000
03891000
03892000
03893000
03894000
03895000
03896000
03897000
03898000
03899000
03900000
03901000
03902000
03903000
03904000
03905000
03906000
03907000
03908000
03909000
03910000
03911000
03912000
03913000
03914000
03915000
03916000
03917000
03918000
03919000
03920000
03921000
03922000
03923000
03924000
03925000
03926000
03927000
03928000
03929000
03930000
03931000
03932000
03933000
03934000
03935000
03936000
03937000
03938000
03939000

```

```

IMPLICIT INTEGER*4 (A - Z)                                03940000
DIMENSION BUFDM(40),SAVE1(6),CARRAY(3)                    03941000
INTEGER*2 VV(13),BUFFER(80),VV$(12),VAR(8),VAR1,VAR2,VAR3,VAR4, 03942000
* VARX(7),OVRDE,F1,F2,CWORD,TITLEX,BUFPT,PREN29,OVRCC(7787) 03943000
INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS, 03944000
* PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR                03945000
INTEGER*2 MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK                 03946000
INTEGER*2 PAGDUM,SRT,COLBEG,CHRFIN                         03947000
COMMON IOUTPG,COL,LN,ERRCNT                                03948000
COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE,     03949000
* REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CIENT1,CICNT, 03950000
* CIINC,DICT,NEXT,HIT,                                     03951000
* OVRDE,F1,F2,CWORD,TITLEX,BUFPT,                          03952000
* LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC,      03953000
* HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR         03954000
COMMON /B/ FIELD1,FIELD2,FIELD3,SPOP,CP,LB,PER,KEEPSV(2);   03955000
* UPPER,UP1,CAP,FIRST,MASK1,EXCNT,CPAREN,EOSCHR,SAVMSK     03956000
COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,MYPAGE,RIVER,HYPTRX,WPT,SUND,TW03957000
*,TWOUP,I,CPSW,ISPOT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTX,LWI,W,03958000
* LINSIZ,NSYM,SPACNG,TXTLNE,K,LSTBL,AUTO,PFLN,NOGO,NAME,SWK028,IIO,03959000
* CCCNT,COLFAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTOTB,SWX,03960000
* JUNK,ID,PDUM,REM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN,03961000
* LN2,REPTTL,IC,CWIDTH,USEWS1,LTITLE,CU(8),TAB2(7),INDP(4), 03962000
* INDARR(8),FLN,ICINC,PARA,TFLN,USTART,FCM,SWWPT,ENDL,EWX,PIVOT, 03963000
* INDEX,TABSEQ,CHAR,ENDP,LINEX(67),WORDS,LNTW,CS,ENDSAV,ID1,LINEX, 03964000
* WANT,GAPS,WANTIN,WSEPD,L,LSIDE,RSIDE,SWEW(68),CARD(40);   03965000
* PAGDUM(7788),SRT(99),COLBEG(8),CHRFIN(99)                03966000
DATA VV,EOP,OLDSER,FWRITE,XWORD,LIST$,XVAR2,CLOSEE        03967000
*/-1192,18455,18551,18487,18486,18790,18744,18699,18711,18680, 03968000
* 18775,18712,18758,                                       03969000
* 0,0,0,10000,0,0,') E '/                                03970000
EQUIVALENCE (VV$(1),VV(2)),(VAR1,VAR(1)),(VAR2,VAR(2)),(BUFFER(1), 03971000
*BUFDM(1)),(BUFFER(1),LINEX(1)),(VAR3,VAR(3)),(VAR4,VAR(4)) 03972000
*,(VAR(1),CU(5)),(VARX(1),VAR(2)),(SAVE1(1),LINEX(41)),(PREN29, 03973000
* SPCHAR(31)),(OVRCC(1),PAGDUM(2)),(CARRAY(1),IOUTPG)      03974000
COMMON /EHRMAN/ DARKER,DROPCH,BACKCH,BACKFL,BACKCT,BAKPTF, 03975000
* BACKWD,BAKPOS,BACHAR,BACKST,BACKND,NULLSW,                03976000
* CCWIDT,NOTRIV,MASK2,EDCCWI,UNDRSW,EDCOL1                  03977000
INTEGER*2 DARKER,DROPCH,BACKCH,BACKFL,BACKCT,BAKPTF,NULLSW, 03978000
* BACKWD(68),BAKPOS(100),BACKST(8),BACKND(8),CCWIDT,NOTRIV, 03979000
* MASK2,EDCCWI,UNDRSW,EDCOL1                                03980000
LOGICAL*1 BACHAR(100)                                       03981000
C                                                                03982000
C                                                                03983000
C                                                                03984000
IF (K .LT. 0) GO TO 100                                     03985000
C*****A*****03986000
802 READ (5,1000,END=803) BUFFER                           03987000
WRITE (1,1001) CCGCNT,CARRAY,BUFFER                       03988000
1001 FORMAT ('A',4A4,80A1)                                  03989000
CCCNT = CCCNT + 1                                           03990000
1000 FORMAT (80A1)                                           03991000
CALL CCRDR (VAR)                                             03992000
819 SAVCCC = CCCNT                                           03993000
IF (NAME .NE. VV(1) .AND. EDCOL1 .NE. DOLLAR) GO TO 8041 03994000
C.....DO NOT TREAT HOMONYMS AS CONTROL CARDS              03995000
DO 804 I = 1, 13                                             03996000
IF (NAME .EQ. VV(I)) GO TO (805,806,845,807,808,809,810,843,847, 03997000
GO $DE $IN $EN $DU $PU $NO $LI $ME 03998000
C.....GO $DE $IN $EN $DU $PU $NO $LI $ME 03999000
* 848,844,870,875), I

```


C..... \$JO \$OV \$LO \$OM	04000000
804 CONTINUE	04001000
8041 CALL MSG(218)	04002000
GO TO 802	04003000
C***** C *****	04004000
803 CALL MSG (800)	04005000
C.....GO	04006000
C.....THE PART OF 'CU' ARRAY USED BY 'VAR' IS ZEROED WHEN 'GO' IS READ	04007000
805 IF (INVALID .NE. 0) GO TO 824	04008000
IF (MERGE .NE. 0) REWIND IREAD	04009000
IF (FWRITE .EQ. EOF) GO TO 801	04010000
824 IF (IREAD .EQ. 4) LIST = LIST\$	04011000
PUNCH = 0	04012000
NODOC = 10	04013000
LOCATE = 0	04014000
MASK1 = 0	04015000
CALL MSG (804)	04016000
GO TO 867	04017000
801 LIST = LIST + LIST\$	04018000
IF (NODOC * OVRDE .EQ. 0) GO TO 867	04019000
C.....NEXT 2 LINES ENABLE \$OVERRIDE CARDS TO WORK WITH \$NO DOCUMENT	04020000
OVRDE = -OVRDE	04021000
NODOC = -10	04022000
867 RETURN	04023000
C.....\$NO DOCUMENT	04024000
810 NODOC = 10	04025000
GO TO 802	04026000
C.....\$PUNCH THE INPUT TAPE	04027000
809 PUNCH = 10	04028000
LIST = 10	04029000
GO TO 802	04030000
C.....\$LIST THE INPUT TAPE	04031000
843 LIST\$ = 10	04032000
GO TO 802	04033000
C.....\$OMIT LISTING OF EDITED TAPE	04034000
875 OMLIST = -10	04035000
GO TO 802	04036000
C.....\$DUPLICATE THE OLD MASTER FROM ITS PRESENT POSITION	04037000
808 IF (EOF .EQ. 0) GO TO 813	04038000
CALL MSG (802)	04039000
GO TO 802	04040000
813 IF (MERGE .EQ. 0) GO TO 835	04041000
CALL MSG (847)	04042000
GO TO 802	04043000
835 IF (INVALID) 800,812,800	04044000
C.....\$END CHANGES	04045000
807 IF (FWRITE .GT. EOF + INVALID) GO TO 812	04046000
800 EOF = 5	04047000
FWRITE = 5	04048000
GO TO 802	04049000
812 READTO = 4000000	04050000
VAR1 = -10	04051000
GO TO 826	04052000
C.....\$INSERT BEFORE XX1,YY1	04053000
845 IF (-VAR3 .EQ. VAR4) GO TO 856	04054000
CALL MSG (212)	04055000
GO TO 846	04056000
C.....\$DELETE XX1,YY1 (THROUGH XX2,YY2)	04057000
806 IF (-VAR3 .NE. VAR4) GO TO 829	04058000
VAR3 = VAR1	04059000

150	VAR4 = VAR2	04060000
829	IF (VAR4 .EQ. 0) VAR4 = 10000	04061000
856	NOGO = 0	04062000
846	IF (EOF .NE. 0) CALL MSG (802)	04063000
	IF (MERGE .NE. 0) CALL MSG (847)	04064000
	IF ((VAR1 .GT. OLDSER .OR. (VAR1 .EQ. OLDSER .AND. VAR2 .GT. XWORD	04065000
	* .AND. REMNNT .NE. 0)) .AND. (VAR3 .GT. VAR1 .OR. (VAR3 .EQ. VAR1	04066000
	* .AND. VAR4 .GE. VAR2) .OR. (-VAR3) .EQ. VAR4)) GO TO 815	04067000
	CALL MSG (212)	04068000
815	IF (NOGO .NE. 0) GO TO 861	04069000
	READTO = VAR1 - OLDSER - 1	04070000
826	IF (FWRITE .NE. 0) GO TO 814	04071000
	REWIND 4	04072000
	FWRITE = 5	04073000
	IREAD = 4	04074000
814	BFOUND = 10	04075000
	INSWRD = 10000	04076000
	IF (COVEA .EQ. 0) GO TO 811	04077000
	IF (OLDSER .EQ. VAR1 .AND. XWORD .EQ. VAR2 - 1) GO TO 861	04078000
	IF (REMNNT .EQ. 0) GO TO 827	04079000
	CALL MSG (814)	04080000
811	IF (OLDSER .EQ. (VAR1 - 1) .AND.	04081000
	* (XWORD + VAR2 + XVAR2) .EQ. 10000) GO TO 861	04082000
	IF (REMNNT .EQ. 0) GO TO 827	04083000
	IF (VAR1 .EQ. OLDSER) INSWRD = VAR2 - XWORD	04084000
	FINISH = 1	04085000
	CALL CONDSE (EOF)	04086000
	FINISH = 0	04087000
827	IF (READTO) 816, 838, 839	04088000
839	INSWRD = 10000	04089000
	DO 822 IN = 1, READTO	04090000
	READ (2,1000,END=818) BUFFER	04091000
	CALL CONDSE (EOF)	04092000
822	CONTINUE	04093000
	OLDSER = OLDSER + READTO	04094000
838	IF (VAR2 .EQ. 0 .OR. COVEA .NE. 0) GO TO 816	04095000
	READ (2,1000,END=818) BUFFER	04096000
	OLDSER = OLDSER + 1	04097000
	INSWRD = VAR2	04098000
	CALL CONDSE (EOF)	04099000
	C.....HAVE REACHED INSERTION POINT	04100000
816	XVAR1 = VAR1	04101000
	XVAR2 = VAR2	04102000
	XVAR3 = VAR3	04103000
	XVAR4 = VAR4	04104000
	IF (INSWRD .EQ. WORDS .OR. INSWRD .EQ. 10000) GO TO 862	04105000
	CALL MSG (805)	04106000
	GO TO 860	04107000
862	IF (VAR2 .NE. 0) GO TO 858	04108000
	IF (COVEA) 861,859,861	04109000
858	IF (COVEA .EQ. 0) GO TO 861	04110000
859	CALL MSG (806)	04111000
860	XWORD = 10000	04112000
861	INSWRD = 10000	04113000
	BFOUND = 10	04114000
825	READ (5,1000,END=803) BUFFER	04115000
	C.....READ AN UPDATE CARD	04116000
	WRITE (1,1001) CCGCNT,CARRAY,BUFFER	04117000
	CCCNT = CCCNT + 1	04118000
	CALL CCRDR (VAR)	04119000

IF (EDCOL1 .NE. DOLLAR) GO TO 8171	04120000
C.....DO NOT TREAT HOMONYMS AS CONTROL CARDS	04121000
DO 817 I = 1, 12	04122000
C.....ABOVE STATEMENT REFLECTS SIZE OF VV\$	04123000
IF (NAME .EQ. VV\$(I)) GO TO 823	04124000
817 CONTINUE	04125000
8171 CALL CONDSE (EOF)	04126000
C.....WRITE REPLACEMENT TEXT	04127000
GO TO 825	04128000
823 IF (NOGO .NE. 0) GO TO 819	04129000
IF (BFOUND .GT. 0 .OR. COVEA .NE. 0) GO TO 821	04130000
DO 837 I = 1, 80	04131000
BUFFER(I) = BLANK	04132000
837 CONTINUE	04133000
CALL CONDSE (EOF)	04134000
C.....SQUEEZE OUT A BLANK	04135000
821 BFOUND = 10	04136000
C.....NOW GO ELSEWHERE IF WE'RE DELETING	04137000
IF (XVAR3 .NE. 0) GO TO 820	04138000
C.....NOTE THAT WORD X1,Y1 (OR CARD IMAGE X1) IS NOT WRITTEN HERE ONTO	04139000
C.....NEW MASTER. THUS, MULTIPLE IDENTICAL \$INSERT'S ARE ALLOWED.	04140000
XWORD = XVAR2 - 1	04141000
IF (XVAR2 .EQ. 0) XWORD = 10000	04142000
GO TO 819	04143000
820 INSWRD = 10000	04144000
IF (XVAR3 .LT. 32767) GO TO 869	04145000
IF (COVEA .NE. 0) GO TO 818	04146000
BUFDUM(1) = CLOSEE	04147000
CALL CONDSE (0)	04148000
818 EOF = 5	04149000
CALL CONDSE (EOF)	04150000
IF (CCCNT - SAVCCC) 802,802,819	04151000
869 DELETE = 10	04152000
XWORD = XVAR4	04153000
READTO = XVAR3 - XVAR1	04154000
IF (REMNT .EQ. 0) GO TO 831	04155000
READTO = READTO - 1	04156000
FINISH = 1	04157000
IF (XVAR3 .EQ. XVAR1) INSWRD = XVAR4 - XVAR2 + 2	04158000
CALL CONDSE (EOF)	04159000
FINISH = 0	04160000
IF (XVAR3 .EQ. XVAR1) GO TO 841	04161000
831 OLDSE = OLDSE + READTO	04162000
INSWRD = 10000	04163000
832 IF (READTO .EQ. 0) GO TO 833	04164000
READTO = READTO - 1	04165000
READ (2,1000,END=871) BUFFER	04166000
CALL CONDSE (EOF)	04167000
GO TO 832	04168000
833 INSWRD = XVAR4 + 1	04169000
842 READ (2,1000,END=871) BUFFER	04170000
OLDSE = OLDSE + 1	04171000
CALL CONDSE (EOF)	04172000
841 IF (INSWRD .GE. 10000) GO TO 836	04173000
IF (INSWRD - WORDS - 1) 836,828,840	04174000
828 IF (BFOUND .NE. 0 .AND. COVEA .NE. 0) GO TO 830	04175000
C.....MUST FLUSH OUT THE CARD WITH A PARTIAL WORD ON IT IF THE COMMAND	04176000
C.....WORD ENDING IN V/E/A WAS SPLIT OVER THE END OF THE CARD	04177000
XWORD = 0	04178000
INSWRD = 1	04179000

GO TO 842	04180000
871 EOF = 5	04181000
840 CALL MSG (807)	04182000
830 XWORD = 10000	04183000
836 DELETE = 0	04184000
GO TO 819	04185000
C.....\$JOIN TAPES XX1 ... XX8	04186000
848 MERGE = -20	04187000
GO TO 854	04188000
C.....\$MERGE TAPES XX1 ... XX8	04189000
847 MERGE = 20	04190000
854 IF (PWRITE .NE. 0) CALL MSG (847)	04191000
DO 851 IN = 1, 8	04192000
TAPE = VAR(IN)	04193000
IF (TAPE .EQ. 0) GO TO 849	04194000
IF (TAPE .LT. 9 .AND. TAPE .NE. 4) CALL MSG (212)	04195000
IF (INVALID .NE. 0) GO TO 851	04196000
REWIND TAPE	04197000
IF (VARX(IN) * (IN - 8) .EQ. 0) MERGE = 20	04198000
852 READ (TAPE, 1000, END=853) BUFFER	04199000
CALL CONDSE (0)	04200000
GO TO 852	04201000
853 CALL CONDSE (10)	04202000
851 CONTINUE	04203000
849 IF (VAR(1) .EQ. 0) CALL MSG (212)	04204000
MERGE = 20	04205000
GO TO 802	04206000
C.....\$OVERRIDE FIRST CONTROL CARD GROUP	04207000
844 SAVE1(1) = NODOC	04208000
SAVE1(2) = COVEA	04209000
SAVE1(3) = F1	04210000
SAVE1(4) = F2	04211000
SAVE1(5) = TITLEX	04212000
SAVE1(6) = CPAREN	04213000
NODOC = -20	04214000
COVEA = 10	04215000
F1 = 1	04216000
F2 = 80	04217000
CPAREN = PREN29	04218000
866 READ (5, 1000, END=803) BUFFER	04219000
WRITE (1, 1001) CCCNT, CARRAY, BUFFER	04220000
CCCNT = CCCNT + 1	04221000
SAVCCC = CCCNT	04222000
CALL CONDSE (0)	04223000
IF (NAME .NE. VV(1) .AND. EDCOL1 .NE. DOLLAR) GO TO 8631	04224000
C.....\$AVOID HOMONYMS FOR CONTROL CARDS	04225000
DO 863 I = 1, 13	04226000
C.....\$ABOVE STATEMENT REFLECTS SIZE OF VV ARRAY	04227000
IF (NAME .EQ. VV(I)) GO TO 864	04228000
863 CONTINUE	04229000
8631 IF (NAME .NE. (-12155) .AND. NAME .NE. (-19384)) GO TO 857	04230000
C.....\$TITLE	04231000
JUNK = 857	04232000
GO TO 868	04233000
857 IF (NAME .LT. 0) GO TO 855	04234000
JUNK = 218	04235000
868 CALL MSG (JUNK)	04236000
COVEA = 10	04237000
GO TO 866	04238000
855 DO 865 I = 1, 80	04239000

```

OVRDE = OVRDE - 1                                04240000
JUNK = -OVRDE                                     04241000
OVRCC(JUNK) = BUFFER(I)                           04242000
C.....REPLACE ABOVE 2 WITH, OVRCC(-OVRDE) = BUFFER(I) WHEN H BUG FIXED. 04243000
865  CONTINUE                                     04244000
      GO TO 866                                   04245000
864  CALL CCRDR (VAR)                             04246000
      NODOC = SAVE1(1)                            04247000
      COVEA = SAVE1(2)                            04248000
      F1 = SAVE1(3)                               04249000
      F2 = SAVE1(4)                               04250000
      TITLEX = SAVE1(5)                           04251000
      CPAREN = SAVE1(6)                           04252000
      GO TO 819                                    04253000
C.....$LOCATE THE FOLLOWING WORDS/PHRASES          04254000
870  READ (5,1000,END=803) BUFFER \               04255000
      WRITE (1,1001) CCGCNT,CARRAY,BUFFER         04256000
      CCCNT = CCCNT + 1                           04257000
      SAVCCC = CCCNT                              04258000
      CALL CCRDR (VAR)                             04259000
      IF (NAME .NE. VV(1) .AND. EDCOL1 .NE. DOLLAR) GO TO 8721 04260000
C.....AVOID HOMONYMS FOR CONTROL CARDS            04261000
      DO 872 I = 1, 13                             04262000
      IF (NAME .EQ. VV(I)) GO TO 819               04263000
872  CONTINUE                                     04264000
8721 WRITE (1,1002) BUFFER                         04265000
1002 FORMAT ('L',16X,80A1)                         04266000
      LOCATE = LOCATE + 1                          04267000
C.....LOCATE = NUMBER OF ARGUMENTS TO BE LOCATED 04268000
      GO TO 870                                    04269000
C*****04270000
C.....THIS IS THE "CREATE" ROUTINE                04271000
C*****04272000
C.....RESET K EACH TIME TO AVOID ERROR FLAGS FOR BAD COMMAND WORDS 04273000
100  K = -10                                       04274000
      READ (5,1000,END=105) BUFFER                04275000
      CALL CONDSE (0)                             04276000
      GO TO 100                                    04277000
105  CALL CONDSE (10)                             04278000
      GO TO 867                                    04279000
      END                                          04280000
C                                                  04281000
C                                                  04282000
C                                                  04283000
SUBROUTINE MSG (/MSGNO/)                          04284000
  IMPLICIT INTEGER*4 (A - Z)                      04285000
  INTEGER*2 LIST,COPIES,SPCHAR,BLANK,NUM,SCWORD,HYPHEN,LOWCAS, 04286000
  * PERIOD,KEY028,LOCATE,CARDIC,OMLIST,DOLLAR     04287000
  INTEGER*2 PAGDUM,SET,COLBEG,CHRFIN              04288000
  COMMON /A/ POSN,IREAD,IWRITE,CCGCNT,PUNCH,NODOC,MERGE, 04289000
  * REMNNT,INSWRD,FINISH,DELETE,COVEA,INVALID,BFOUND,CICNT1,CICNT, 04290000
  * CIINC,DICT,NEXT,HIT,                          04291000
  * ARRAY1(3),                                     04292000
  * LIST,COPIES,SPCHAR(42),BLANK,NUM(10),SCWORD,CARDIC, 04293000
  * HYPHEN,LOWCAS,PERIOD,KEY028,LOCATE,OMLIST,DOLLAR 04294000
  COMMON /C/ ALT,BLNKLN,ITEXT,LINPAG,MYPAGE,RIVER,HYPTRX,WPT,SUND,TWO 04295000
  *,TWOUP,I,CPS*,ISPOT,ASIS,CCHAR,CHARCO,NEWH,SAVCCC,KEEP,WPTX,LWI,N, 04296000
  * LINSIZ,NSYN,SPACNG,TXTLNE,K,LSTBL,AUTO,FPLN,NOGO,NAME,SWK028,IIU, 04297000
  * CCCNT,COLPAG,IVALUE,LINSZ,PAGENO,START,UNDERL,J,CONST,AUTOTB,SWX, 04298000
  * JUNK,ID,PDIM,RFM,IER700,END,CENTER,CSEP,INDENT,TEXEND,TLLN,CLEAN, 04299000

```

)

۴

1.