

```
*****
*           Examples of Print Commands for           *
*           Boilerplating                           *
*****
```

The following series of commands are offered for your study without comment. Feel free to use them, change them or add your own.

F1--Form letter using variables from keyboard

F2--Form letter using Text data file

F3--Form letter using fixed length data file

H1--Headings using company name and page number

H2--Headings using company name, date and page number

F1--Form letter using variables from keyboard

```
\
\SETUP
\
\COPY0,FORMS
\GET DATE="Today's Date?"
\TEXT
\
\GET NAME="Full name? (First name first)"
\SET COMPANY="",GET COMPANY="Company, if any?"
\GET ADDR1="First line of address?"
\GET ADDR2="Second line of address, if any?"
\GET SALUT="Salutation?"
\
\RIGHT,:DATE\
\LEFT,LINE4
\ :NAME\
\IF COMPANY<>"",:COMPANY,NL
\ :ADDR1\
\ :ADDR2\
```

Dear \:SALUT\,

F2--Form letter using Text Data File

\

\SETUP

\

\GET DATE="Today's Date?"

\GET FNAME="Name of date file?"

\GET START="Starting record number?"

\GET STOP="Ending record number (if other than end of file)?"

\IF &START<>0,START#START

\IF &STOP<>0,STOP#STOP

\

\FILE T12, :FNAME

\DATA ,NAME,COMPANY,ADDR1,ADDR2,COUNTRY,SALUT,CODE,PS

\

\TEXT

\

\RIGHT, :DATE\

\LEFT,LINE4

\:NAME\

\IF COMPANY<>"",:COMPANY,NL

\:ADDR1\

\:ADDR2\

\IF COUNTRY<>"",:COUNTRY,NL

Dear \:SALUT\,

F3--Form letter using fixed length data file

\
\SETUP

\
\GET DATE="Today's Date?"

\GET FNAME="Name of data file?"

\GET START="Starting record number?"

\GET STOP="Ending record number (if other than end of file)?"

\IF &START<>0,START#START

\IF &STOP<>0,STOP#STOP

\
\FILE F128,:FNAME

\DATA NAME(25),COMPANY(25),ADDRESS(20),CITY(15),STATE(2),ZIP(5),SALUT(15),10

\
\TEXT

\
\RIGHT,:DATE\

\LEFT,LINE4

\:NAME\

\IF COMPANY<>"",:COMPANY,NL

\:ADDRESS\

\:CITY\, \:STATE\ \:ZIP\

De \:SALUT\,

A FOREWORD TO THE INSTRUCTIONS

The step-by-step instructional program for the **MAGIC WAND** Word Processing System is for the person who has not previously worked on a computer or word processor. We have made only a few assumptions about what you already know, i.e., that you know how to turn on your computer, mount a disk on your disk drive, load CP/M (an operating language for your machine), and prepare the printer for printing. Indeed, we must assume these things since we do not know what kind of equipment you are using.

The program is self-paced. You can take as much, or as little, time as you need. You may repeat or skip sections if you wish, but please take the lessons in order. Each lesson assumes that you have learned the material in the preceding lessons.

We have tried to avoid using technical jargon, but sometimes it was not possible to do so. Consequently, we have included a glossary to define some of the expressions we could not avoid. The glossary is not intended to be comprehensive, but hopefully any questions you have will be answered there or elsewhere.

One final piece of advice. If you have not had much experience with computers, you may be apprehensive about using the **MAGIC WAND**. Don't be. There is no mistake that cannot be corrected if you take the time to think it through.

TERMINAL

The terminal keyboard acts like a typewriter with the following exceptions:

1. You can lock the alphabet into uppercase without shifting the non-letter keys.
2. Most terminals come with a repeat function, either as a separate REPEAT key or as an automatic repeat on any key that is depressed longer than usual.

CURSOR

The cursor is a bar of light that lets you know your location on the screen. The movement of the cursor is the equivalent of the movement of the carriage on a typewriter.

RETURN KEY

When you print text on the Text Screen, the machine automatically ends a line with a complete word (determined by the presence of a following blank). Use the RETURN key *only* when you need to force a new line, e.g., at the end of paragraphs. You will need to break a few habits in order to adjust to this, but once you do, your typing speed will have greatly increased. If you press RETURN accidentally, you may use the LINE DELETE Control Key to correct your mistake. (See Lesson Two) **Note:** The location of a

carriage return in the text is represented on the screen by a tilde (~).

USING CONTROL KEYS

Pressing CONTROL while pressing another key is a way of *supershifting* the keyboard. It tells the computer that you are giving it a command. Pressing CONTROL without pressing another key at the same time has no effect. Example: to execute a CONTROL C, you would:

- 1) Press the CONTROL key and hold it.
- 2) Press C as if you were typing it.
- 3) Release C.
- 4) Release CONTROL.

In the lessons we refer to Control keys by their function, rather than by the actual key. This way we have been able to take advantage of any special keys on your terminal. Use the reference card that came with your package to find out which keys to use on your particular terminal.

FORMAT OF INSTRUCTIONS

Each lesson is divided into sections. The sections are composed of numbered lines and lettered lines. The lettered lines describe the general operation of a function. The numbered lines give you specific ways to use the function in the lessons. Next to each section is a simulation of what your work should look like.

COMMANDS

Some commands are activated by pressing RETURN. You can always recognize these because we tell you to **type** them. We do not always tell you to press RETURN to activate a command, because we assume that you will do this automatically. On the Text Screen most commands are given with Control Keys which are *not* followed by RETURN. You can always recognize a command that does not require RETURN because we tell you to **press** a key. Example: You are in CP/M preparing to load the first sample file. The instructions tell you to type **EDIT SAMPLE1 LESSON1**. They do not tell you to press RETURN afterwards, but if you

do not press RETURN, nothing happens.

THE COMMAND AND TEXT SCREENS

EDIT consists of two different screens, the Command Screen and the Text Screen. As the name implies, you give commands on the Command Screen which may or may not affect the file you are editing. On the Text Screen practically everything you do affects the text file. You can get from the Command Screen to the Text Screen any time you press RETURN without typing in a command. You can get from the Text Screen back to the Command Screen any time you press ESCAPE.

TABLE OF CONTENTS

Preface	iii
Table of contents	v
Foreword to Instructions	1
Lesson One, Basic EDIT features, part one	3
Lesson Two, Basic EDIT features, part two	7
Lesson Three, Templates, Boilerplates, etc.	16
Lesson Four, Introduction to PRINT features	23
Lesson Five, Embedded PRINT commands	31
Lesson Six, Variables	42
Lesson Seven, External Data Files, Form Letters	51
Lesson Eight, Conditional Commands	58
Lesson Nine, Multi-page documents	67
CP/M® for MAGIC WAND Users	79
<small>CP/M is a registered trademark of Digital Research Corp.</small>	
Notes on EDIT	83
Notes on PRINT	96
Some Notes for Programmers	123
Glossary	125
Index	127

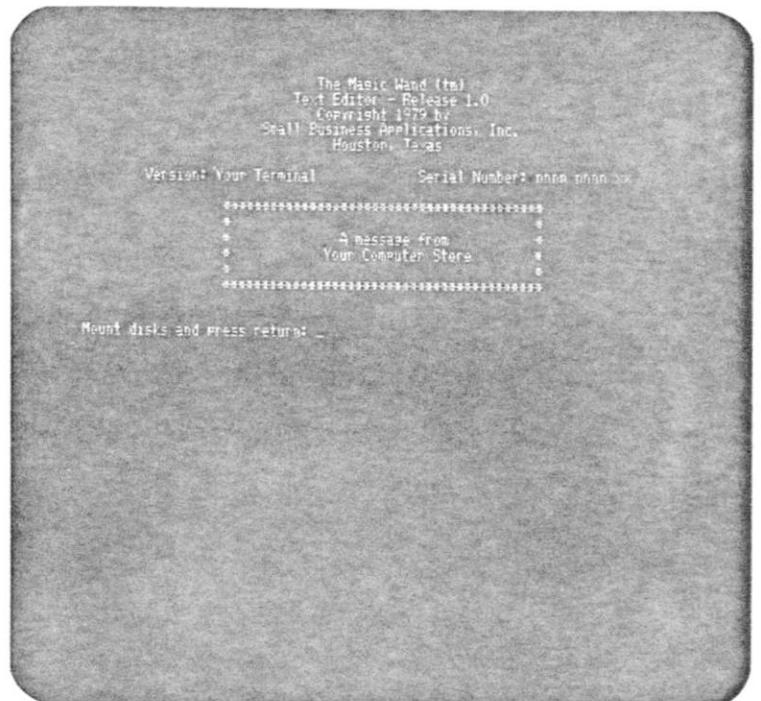
LESSON ONE

The first two lessons deal with basic editing features: how to load and end EDIT, how to make changes to the text and print it with the printer.

This is the situation: You are a secretary working for Abraham Lincoln. You have a rough draft for an address he plans to give at Gettysburg. Somehow it does not seem to be quite right. As the lessons progress, we give you revisions to make to it so that, slowly, it becomes the speech that was eventually given.

LOADING EDIT PROGRAM

- A. Turn on the computer and load CP/M.
- B. The machine prints **A>** when it is ready for a command.
- C. Load the diskette containing the **MAGIC WAND** and accompanying sample files on **DRIVE A**.
- D. Type **EDIT SAMPLE1 LESSON1**.
(Remember to press RETURN)
- E. The machine prints the screen to the right.



INTRODUCTION TO THE COMMAND SCREEN

- A. The machine gives you the opportunity to change disks. (Since we don't need to change disks, press RETURN.)
- B. The machine notifies you that it is inputting the SAMPLE1 text file from the disk.
- C. The machine prints the Command Screen (at right) followed by an asterisk indicating that it is waiting for a command.
- D. Type **L65**. (This sets the line length at 65 characters.) (Did you remember to press RETURN?)
- E. Make sure your printer is ready to print.
- F. Type **P**. (So you can have a copy of the text before you edit it.)
- G. The printer prints the file with a line length of 65 characters.
- H. When the printing is complete, press RETURN without typing anything else to enter the Text Screen.

```

                                Edit System Status
Reading   Finished SAMPLE1
Writing  Active   LESSON1
Workspace contains 8 Lines/Paragraphs
Character: 34052-Total  1797-In Use  32255-Remaining
Mode: Text
Line Width: 80
Tab Columns: 1 9 17 25 33 41 49 57 65 73
*

```

CURSOR MOVEMENT CHARACTER REPLACEMENT

- A. All additions and changes to the text take place at the cursor, so you need to be able to move the cursor to any location on the screen.
- B. To move the cursor without changing the text, use the **UP**, **DOWN**, **LEFT** and **RIGHT** Control Keys.
- C. To replace one character with another, move the cursor to the character to be changed and type the correction.
- D. Any character including a blank may be substituted for any other character.
- E. **Note:** The Space Bar is like any other key on the keyboard. Instead of printing characters, it prints blanks. *Do not use the space bar to move the cursor unless you want to blank out the current cursor position.*
 1. Practice moving the cursor.
 2. Capitalize **gettysburg** in the first paragraph.
 3. Change **notion** in the second paragraph to **nation**.
 4. Change the comma following **Now** in the third paragraph to a blank (by hitting the space bar).

```

                                LINCOLN'S ADDRESS AT GETTYSBURG, 1863
...
It's great to be here in Pennsylvania. Mrs. Lincoln and I
appreciate the hospitality you have shown us during our stay.
I wish that I could say that everyone loves it here. I met a
soldier who told me that he would rather have died at Gettysburg
than have to live in Philadelphia.
I could go on like this all day, but let us be serious for a
moment. Eightyseven years ago our ancestors brought forth on
this continent a new nation, conceived and dedicated to the
PROPOSITION that all men are made pretty much the same.
Now we are engaged in a real big civil war, testing whether
that nation conceived and dedicated like that can endure for
very long. We are now on a real big battle field of that war.
We have come to dedicate some of that field, as a memento for
those who here have their lives that that nation might live.
It is altogether fitting and proper that we should do this, and
I am not at all sorry that I can.
But, let's face it, we can not dedicate—we can not
consecrate—we can not even hallow out—this ground. The brave
men, living or dead, who struggled here, have consecrated it,
far above our poor power to add or subtract, multiply or divide.

```

CHARACTER DELETE

- A. Move the cursor to the character you wish to delete.
- B. Press the CHARACTER DELETE Control Key.
- C. The machine deletes the character and moves all the characters to the right of the cursor one space to the left. The cursor does not move.
- D. If this makes room for the first word of the next line, the machine moves it up and rewrites the line.
- E. To delete more than one character, press the CHARACTER DELETE Control Key once for each character you wish to delete.
 1. Delete the extra **l** in **hospitallity** in the first paragraph.
 2. Delete **r** in **livers** in the third paragraph.
 3. Delete **a** in **meat** in the third paragraph.
 4. Delete **here** in the first line. (Remember to delete one of the spaces as well.)

CHARACTER INSERT

- A. Move the cursor to the character (or space) *in front of which* you wish to insert a character.
- B. Press the CHARACTER INSERT Control Key.
- C. Type the character to be inserted.
- D. The machine inserts the new character and moves the cursor and all characters to the right of the cursor one space to the right.
- E. If this pushes a word off the line, the machine moves the word to the next line and rewrites the line.
- F. The machine continues to insert each character that you type until you use any other control key (backspace, tab, delete, cursor movement, etc.).
 1. Insert another **D** in **ADRESS** on the first line. (Move the cursor to either the **D** or the **R**.)
 2. Insert a hyphen between **eighty** and **seven** in the second paragraph. (Move the cursor to the **s**.)
 3. Insert **in liberty** between **conceived** and **and** in the second paragraph. (Move the cursor to either the blank following **conceived** or to the **a** in **and**. Remember to add necessary spaces.)
 4. Insert **or any nation** between **nation** and **conceived** in the third paragraph. (Move either the blank following **nation** or the **c** in **conceived**.)

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be in Pennsylvania, Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia.

I could go on like this all day, but let us be serious for a moment. Eighty-seven years ago our ancestors brought forth on this continent a new nation, conceived and dedicated to the proposition that all men are made pretty much the same.

Now we are engaged in a real big civil war, testing whether that nation conceived and dedicated like that can endure for very long. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came.

But, let's face it, we can not dedicate—we can not consecrate—we can not even hollow out—this ground. The brave men, living or dead, who straggled here, have consecrated it, far above our poor power to add or subtract, multiply or divide.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be in Pennsylvania, Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia.

I could go on like this all day, but let us be serious for a moment. Eighty-seven years ago our ancestors brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are made pretty much the same.

Now we are engaged in a real big civil war, testing whether that nation or any nation conceived and dedicated like that can endure for very long. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came.

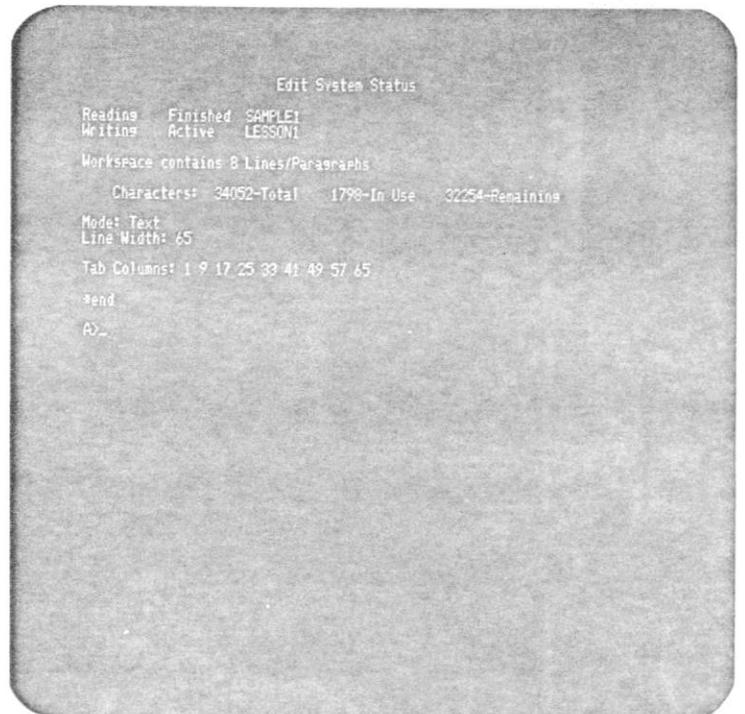
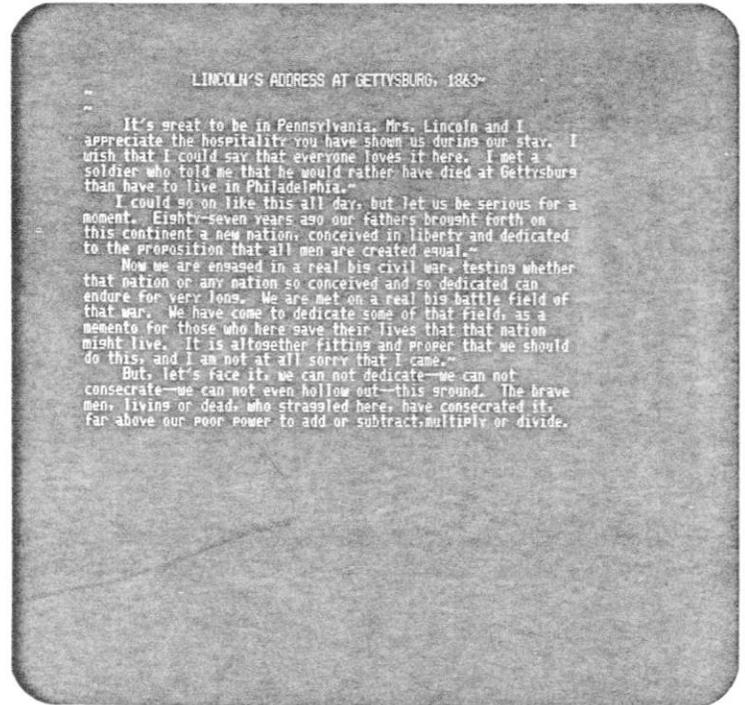
But, let's face it, we can not dedicate—we can not consecrate—we can not even hollow out—this ground. The brave men, living or dead, who straggled here, have consecrated it, far above our poor power to add or subtract, multiply or divide.

USING WHAT YOU HAVE LEARNED

- A. Usually there is more than one way to make changes on the Text Screen.
- B. When changing a word or expression, you might insert the new expression first and then delete the old expression.
- C. You might choose instead to overwrite the old expression as much as possible, using either the CHARACTER INSERT or CHARACTER DELETE Control Key as needed.
- D. What works best for you is the right way for you.
 1. Change made pretty much the same to **created equal** in the second paragraph.
 2. Change **ancestors** to **fathers** in the second paragraph.
 3. Change **conceived and dedicated like that** to **so conceived and so dedicated** in the third paragraph.

EXITING FROM EDIT

- A. Press the ESCAPE key.
- B. The machine returns to the Command Screen.
- C. Type END. (Remember to press RETURN.)
- D. The machine stores the newly edited text under the name LESSON1.
- E. The machine reloads CP/M from DRIVE A and prints A> on the screen.
- F. **Note:** If the disk on DRIVE A does not contain a copy of CP/M, the machine hangs up until you load a disk containing CP/M on DRIVE A.



LESSON TWO

RELOADING EDIT AND LESSON1

1. Load EDIT and text file LESSON1. (Type **EDIT LESSON1**)
2. Set the line length at 65 characters. (Type **L65**)
3. Press RETURN to enter the Text Screen.

```

Edit System Status
Updating LESSON1 (Input Finished)
Workspace contains 8 Lines/Paragraphs
  Characters: 34052-Total  1798-In Use  32254-Remaining
Model: Text
Line Width: 80
Tab Columns: 1 9 17 25 33 41 49 57 65 73

```

POSITIONING TEXT ON SCREEN
(SCROLLING)

- A. Press the FORWARD LINE SCROLL Control Key.
 - B. The machine moves the text up one line on the screen and adds the next line from the file at the bottom of the screen.
 - C. Press the BACKWARD LINE SCROLL Control Key.
 - D. The machine moves the text down one line on the screen and adds the preceding line at the top of the screen. (On some terminals the machine must reprint the entire screen.)
 - E. Press the FORWARD PAGE SCROLL Control Key.
 - F. The machine prints the next screen of text. The last line of the previous screen is the first line of the new screen.
 - G. Press the BACKWARD PAGE SCROLL Control Key.
 - H. The machine prints the preceding screen of text. The first line of the previous screen is the last line of the new screen.
- Note:** Scrolling the text does not change the position of the cursor on the screen.
1. Experiment with the scrolling Control Keys to get a feeling for what they can do.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

```

It's great to be in Pennsylvania. Mrs. Lincoln and I
appreciate the hospitality you have shown us during our stay. I
wish that I could say that everyone loves it here. I met a
soldier who told me that he would rather have died at Gettysburg
than have to live in Philadelphia.

I could go on like this all day, but let us be serious for a
moment. Eighty-seven years ago our fathers brought forth on
this continent a new nation, conceived in liberty and dedicated
to the proposition that all men are created equal.

Now we are engaged in a real big civil war, testing whether
that nation or any nation so conceived and so dedicated can
endure for very long. We are met on a real big battle field of
that war. We have come to dedicate some of that field, as a
memento for those who here gave their lives that that nation
might live. It is altogether fitting and proper that we should
do this, and I am not at all sorry that I came.

But, let's face it, we can not dedicate—we can not
consecrate—we can not even hallow out—this ground. The brave
men, living or dead, who struggled here, have consecrated it,
far above our poor power to add or subtract, multiply or divide.

```

OTHER CURSOR MOVEMENTS

- A. Press the HOME Control Key.
- B. The cursor moves to the beginning of the line it is on.
- C. If the cursor is already at the beginning of a line, it moves to the beginning of the first line on the screen.
- D. Press the TAB Control Key.
- E. The cursor moves to the next tab marker to the right on the line it is on.
- F. If the cursor is beyond the last tab marker on the line, it moves to the first position of the next line.
- G. Press the TOP Control Key.
- H. The machine prints the first screen of text and positions the cursor on the first character of the first line of text.
- I. Press the BOTTOM Control Key.
- J. The machine writes the last screen of text and positions the cursor at the beginning of the first line following the end of the text.

Using the SCROLL, HOME, TAB, TOP and BOTTOM Control Keys, in addition to regular cursor movements, make the following changes:

1. Change endure for very long in the third paragraph to long endure.
2. Change even hollow out in the fourth paragraph to hallow.
3. Change let's face it in the fourth paragraph to in a larger sense.

this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal."

Now we are engaged in a real big civil war, testing whether that nation or any nation so conceived and so dedicated can endure for very long. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came."

But, let's face it, we can not dedicate—we can not consecrate—we can not even hollow out—this ground. The brave men, living or dead, who straggled here, have consecrated it, far above our poor power to add or subtract, multiply or divide. The world can never forget what they did here."

It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth."

than have to live in Philadelphia."

I could go on like this all day, but let us be serious for a moment. Eighty-seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal."

Now we are engaged in a real big civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came."

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who straggled here, have consecrated it, far above our poor power to add or subtract, multiply or divide. The world can never forget what they did here."

It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion that this nation, under God, shall

LINE DELETE

- A. Move the cursor to the beginning of the line.
- B. Press the **LINE DELETE** Control Key twice. (This helps to prevent an accidental deletion of the entire line.)
- C. The machine deletes the line and moves the remaining text up one line.
- D. To delete only the last part of a line, move the cursor to the first character to be deleted.
- E. Press the **LINE DELETE** Control Key twice.
- F. The machine deletes from the cursor to the end of the line and moves the remaining text to the cursor.
- G. On subsequent, consecutive line deletions, you need to press the **LINE DELETE** Control Key only once for each line.
- H. The **LINE DELETE** Control Key is the only way to delete a carriage return.
- I. To delete a carriage return, move the cursor to the position of the carriage return [represented by a tilde (~)] and press the **LINE DELETE** Control Key twice.
 1. Delete from **I could go on to for a moment.** in the second paragraph. (You need to use a combination of **LINE DELETE** and **CHARACTER DELETE**. Make sure that your paragraph is indented properly.)
 2. Delete **and I am not at all sorry I came** in the third paragraph. Replace the final comma with a period. (Press **RETURN** to reestablish the end of the paragraph)
 3. Delete **multiply and divide.** in the fourth paragraph. Change **subtract**, to **detract**. Correct the punctuation and spacing.
 4. Merge the fourth and fifth paragraphs by deleting the carriage return at the end of the fourth paragraph.

than have to live in Philadelphia."

I could go on like this all day, but let us be serious for a moment. Eighty-seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal."

Now we are engaged in a real big civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came."

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or subtract. Multiply or divide. The world can never forget what they did here."

It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion that this nation, under God, shall

than have to live in Philadelphia."

Eighty-seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal."

Now we are engaged in a real big civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this."

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world can never forget what they did here. It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth."

FULL INSERT

- A. Position the cursor on the character in front of which you wish to make an insertion. (Just like a CHARACTER INSERT.)
- B. Press the FULL INSERT Control Key.
- C. The machine blanks the screen from the cursor to the bottom of the screen.
- D. For reference, the machine prints the next line of text on the message line at the bottom of the screen.
- E. Type in the text to be inserted making any necessary correction as you go along.
- F. The machine remains in Full Insert mode until you press the END INSERT Control Key, or the ESCAPE, TOP, BOTTOM or SEARCH Control Keys.
 1. Insert between **world** and **can** at the beginning of the last paragraph: **will little note, nor long remember, what we say here, but it** (Remember to press the END INSERT Control Key when you are finished. Make sure that the spacing is correct.)
 2. Insert between **devotion** and **that** in the last paragraph: **— that we here highly resolve that these dead shall not have died in vain—**.

SETTING BLOCK MARKERS

- A. Move the cursor to the first character in a block of text.
- B. Press the BLOCK MARKER Control Key.
- C. The machine inserts a marker (_) and moves the remainder of the line one space to the right.
- D. Move the cursor to the first character following the block.
- E. If the block ends with a carriage return, move the cursor to the first position on the next line.
- F. Press the BLOCK MARKER Control Key again.
- G. The machine inserts a second marker and moves the remainder of the line one space to the right.
 1. Set a BLOCK MARKER in front of **It is rather for us** in the last paragraph. (Move the cursor to the **I**.)
 2. Set a second BLOCK MARKER following **task before us**. (Move the cursor to the blank following **us**.)

continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.—

Now we are engaged in a real big civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.—

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here; but it can never forget what they did here.— It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth.—

continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.—

Now we are engaged in a real big civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.—

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here; but it can never forget what they did here. _ It is rather for us to be here dedicated to the real big task before us. _ It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth.—

BLOCK MOVEMENT

- A. After setting the block markers, move the cursor to the character in front of which you wish to move the block.
- B. Press **ESCAPE** to exit the Text Screen.
- C. The machine prints the Command Screen.
- D. Type **BM (Block Move)**.
- E. The machine reprints the Command Screen.
- F. Press **RETURN**.
- G. The machine reprints the current screen of text with the movement completed and the cursor on the first character of the moved block.
 1. Move the marked block of text between so nobly advanced and —that from those in last paragraph. (Move the cursor to the first dash.)
 2. Correct the punctuation and spacing.

altogether fitting and proper that we should do this."
 But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the real big task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth."

BLOCK DELETE

- A. Using the **BLOCK MARKER** Control Key, set markers around the text to be deleted.
- B. Make sure that the cursor is not left within the block.
- C. Press **ESCAPE** to exit the Text Screen.
- D. The machine prints the Command Screen.
- E. Type **BD (Block Delete)**.
- F. The machine prints the number of characters in the block and requests verification that you wish to delete the block.
- G. Type **Y**.
- H. The machine reprints the Command Screen.
- I. Press **RETURN** to re-enter the Text Screen.
- J. The machine reprints current the screen of text with the block deleted.
 1. Delete Paragraph One. (Remember to insert the second marker at the first position of the second paragraph.)

LINCOLN'S ADDRESS AT GETTYSBURG, 1863-

"It's great to be in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia."
 Eighty-seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal."
 Now we are engaged in a real big civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here save their lives that that nation might live. It is altogether fitting and proper that we should do this."
 But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to

SEARCH

- A. Press the SEARCH Control Key.
- B. The cursor drops to the message line at the bottom of the screen.
- C. The machine prints a colon (:) and waits for you to type the characters or words you are searching (search string). (See glossary for string)
- D. Type the search string exactly as it appears, including upper and lower case.
- E. Press RETURN.
- F. The machine reprints the screen with the cursor on the first character of the first occurrence of the search string after the original position of the cursor.
- G. If the search string only occurs before the cursor, the machine does not find it.
- H. If the search string is not found, the machine prints a message at the bottom of the screen.
- I. To search for the next occurrence of the same string, press the REPEAT SEARCH Control Key.
 1. Search for incomplete and change to unfinished.
 2. Search for straggled and change to struggled. (Remember to change the position of the cursor before the search.)

SEARCH AND REPLACE

- A. Follow steps A through D in the SEARCH section.
- B. Press the SEARCH Control Key again.
- C. The machine prints a colon on the message line following the search string and waits for you to type the correction (replacement string).
- D. Type the replacement string and press RETURN.
- E. The message line should look like this —
:Search String:Replacement String
- F. The machine reprints the screen with the cursor positioned immediately following the correction.
- G. Steps G, H and I of the SEARCH section still apply.
- H. Note: SEARCH AND REPLACE is especially useful when you need to change a short expression into a longer one.
 1. Change Eighty- to Four score and . (:Eighty-:Four score and)
 2. Change some to a portion. (:some:a portion)
 3. Change memento to final resting place. (:memento:final resting place)

Eighty-seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal."

Now we are engaged in a real big civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this."

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the real big task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall straggled

have come to dedicate a portion of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this."

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the real big task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth."

:memento:final resting place

MULTIPLE SEARCH AND REPLACE

- A. Follow steps A through D above.
- B. Press the SEARCH Control Key for a third time.
- C. The machine prints a third colon behind the replacement string on the message line and waits for you to type a number.
- D. Type the number of times you wish to change the search string and press RETURN, e.g., if you wish to change the first four occurrences following the cursor, type 4.
- E. The message line should look like this:
:Search String:Replacement String:Times
- F. To change all occurrences of the search string following the cursor, press RETURN without entering a number.
- G. The machine makes the changes and rewrites the screen with the cursor following the last change.
- H. The machine prints the number of times the string was found on the message line at the bottom of the screen if it was fewer than specified (or if no number was specified).
 1. Change real **big** to **great** every time it occurs. (:real big:great:)
 2. Change **americans** to **the people** twice. (:americans:the people:2)
 3. Use the REPEAT SEARCH to change the final occurrence of **americans** to **the people**.

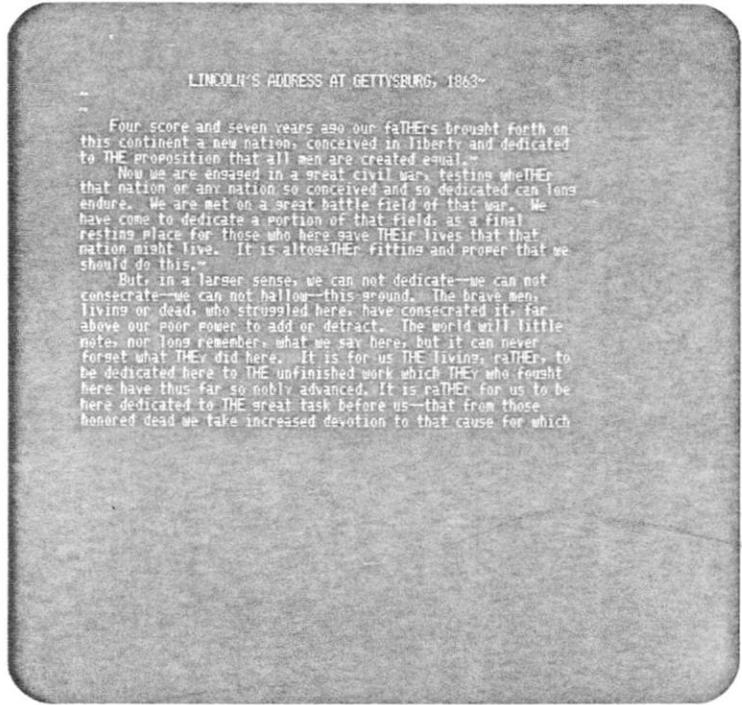
here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of americans, by americans, for americans, shall not perish from the earth.”

Found 3 occurrences.

that government of the people, by the people, for the people, shall not perish from the earth.”

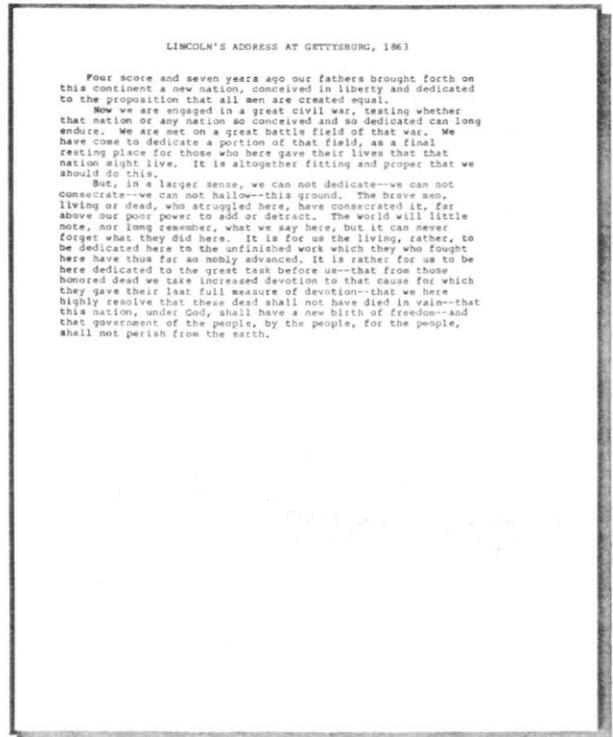
MORE SEARCH AND REPLACE

1. Move the cursor to the top of the text.
(Press the TOP control Key.)
2. Change all occurrences of **the** to **THE**.
(:the:THE:)
3. Notice how the machine has changed **the** in **faTHERs** and **whEThER** but has left unchanged **The**.
4. This is because the machine looks for the search string exactly as you type it.
5. Change **THE** back to **the** (:THE:the:)
6. Now change **the** to **THE**, but this time use leading and trailing blanks.
(: the : THE :).
7. Notice how only the word **the** has been changed.
8. Change **THE** back to **the**.
9. If you had wanted to change **The** as well, you would have needed to search for **The** in a different search statement.



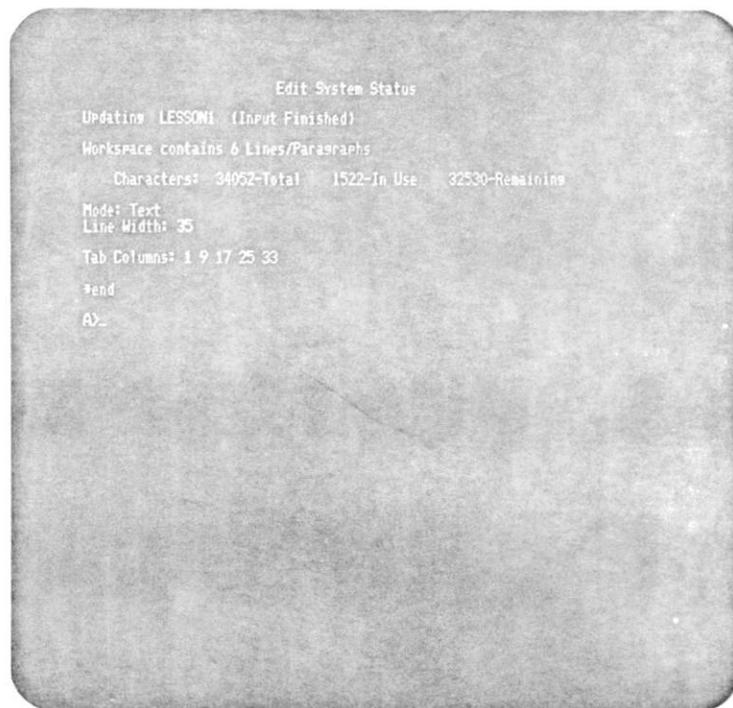
MORE PRINTING

- A. Press ESCAPE to exit the Text Screen.
- B. The machine prints the Command Screen.
- C. Make sure the printer is ready.
- D. Type **P**.
- E. The printer prints the edited file.
- F. When the printing is complete, the machine reprints the Command Screen.
 1. Proof-read the text to make sure there are no mistakes.
 2. Print the text with the current line length (65 characters).
 3. Print the text with a line length of 50 characters. (Type **L50** before printing)
 4. Print the text with a line length of 35 characters. (Type **L35**)



ENDING EDIT

- A. Type **END**.
- B. The machine stores the newly edited text as **LESSON1** and renames the original **LESSON1** as **LESSON1.BAK**.
- C. The machine reloads CP/M from DRIVE A and prints **A>**.



REVIEW

1. Reload SAMPLE1 (Type **EDIT SAMPLE1 LESSON1A.**)
2. Use the copy of the Gettysburg Address that you just printed.
3. Turn SAMPLE1 into the Gettysburg Address.
4. Make the corrections, additions, deletions in any order that you choose, using the methods you prefer.
5. You may refer back to the lessons if you need to review, but try to do it using only the reference card as a guide.
6. Print one more copy of the edited file.
7. Type **END** to exit EDIT.

LESSON THREE

TEMPLATES, BOILERPLATES, DISPLAY, INSERT, HELP FILES

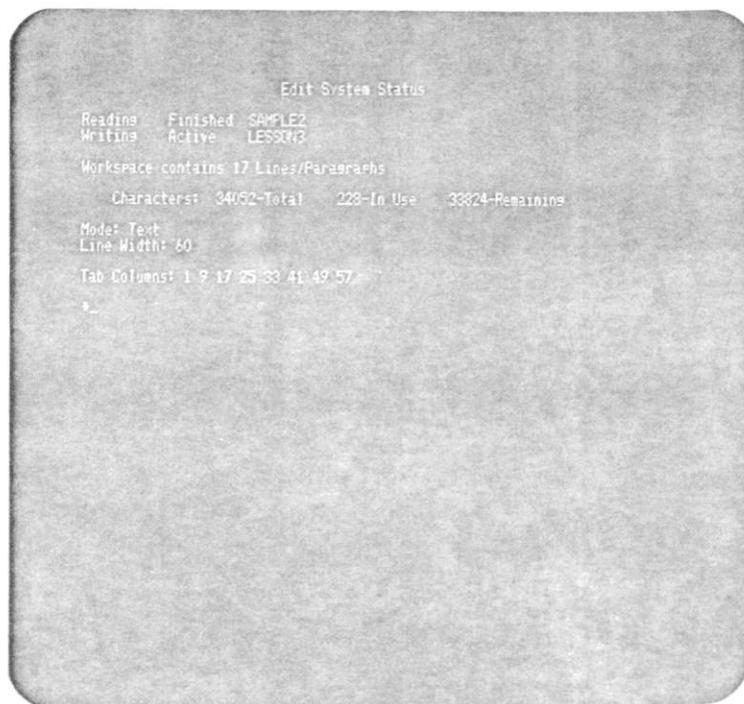
You are a secretary working in the White House for President Lincoln. The fame of his Address at Gettysburg continues to spread, and hardly a day goes by that he does not receive letters about it.

President Lincoln wants to answer all of these letters personally, but does not have time to dictate. To save him time, you have established a file of stock responses which you can use when preparing the answers.

In this lesson you will learn how to use a template for a letter, a file of paragraphs for boilerplating, and a HELP file for easy reference. It starts to get a little tricky, so take it slowly.

LOADING EDIT PROGRAM
SAMPLE2 FILE

1. Type **EDIT SAMPLE2 LESSON3**.
2. The machine loads **EDIT** and gives you an opportunity to change disks.
3. Press **RETURN**.
4. The machine loads **SAMPLE2** and prints the Command Screen.
5. Type **L60** to set the line length at 60 characters.
6. Press **RETURN** to get to the Text Screen.



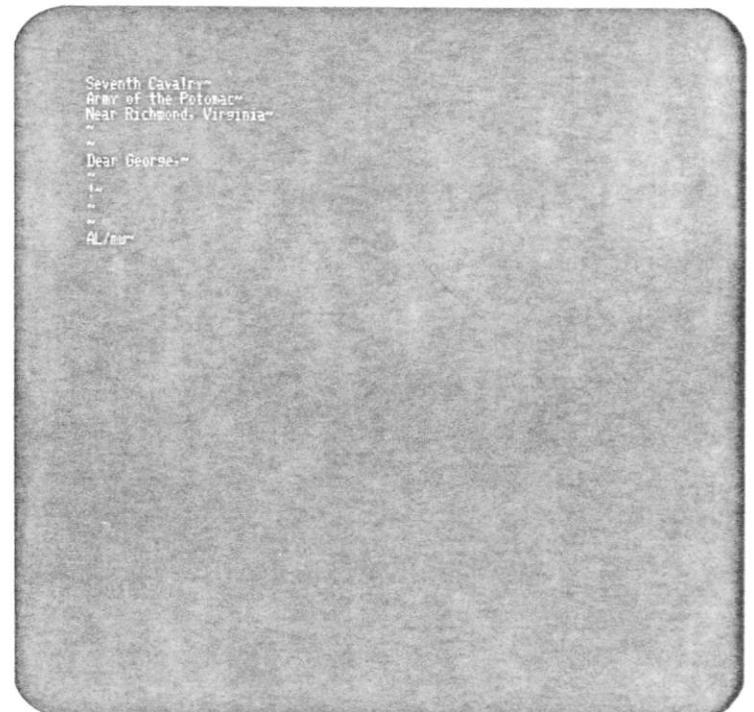
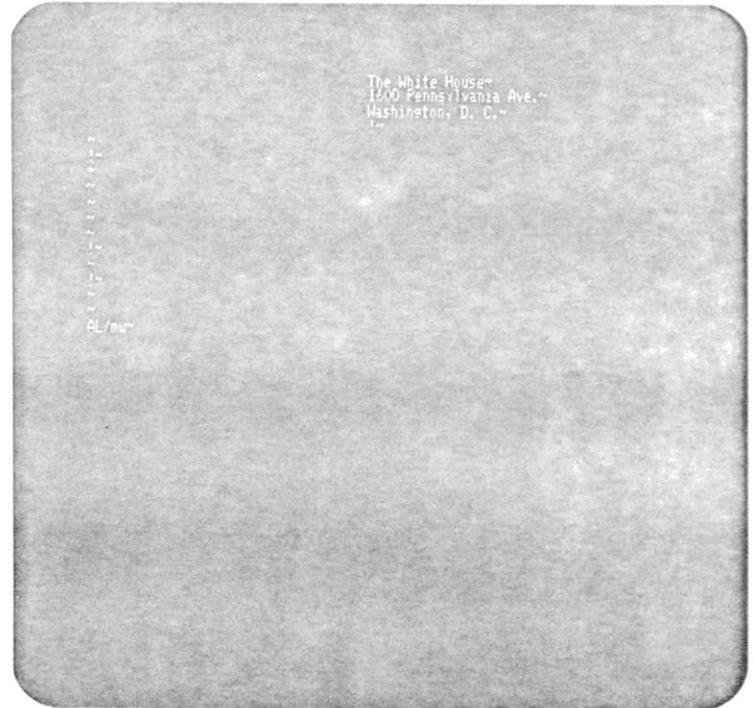
USING A TEMPLATE

Definition: A template is the skeleton of a document with markers indicating the places that need to be filled in.

- A. The machine prints on the Text Screen a sample of a template for a letter.
- B. Press the REPEAT SEARCH Control Key.
- C. The machine searches for the first occurrence of an exclamation point. (**Note:** It always looks for an exclamation point if you have not previously designated a search string.)
- D. The machine reprints the screen with the cursor located at the first exclamation point.
- E. Each successive REPEAT SEARCH command moves the cursor to the next exclamation point.

Using the REPEAT SEARCH Control Key to move the cursor, type the following information in the template:

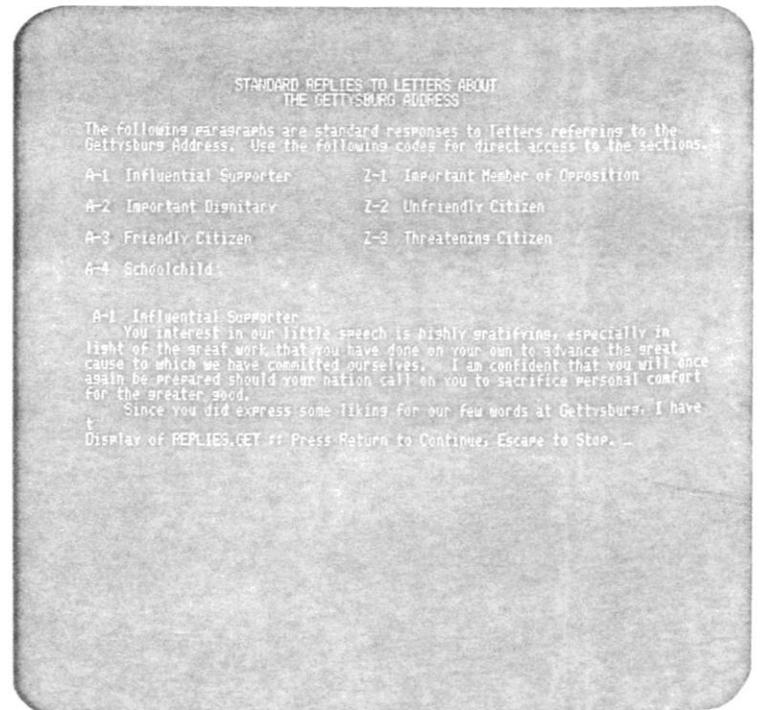
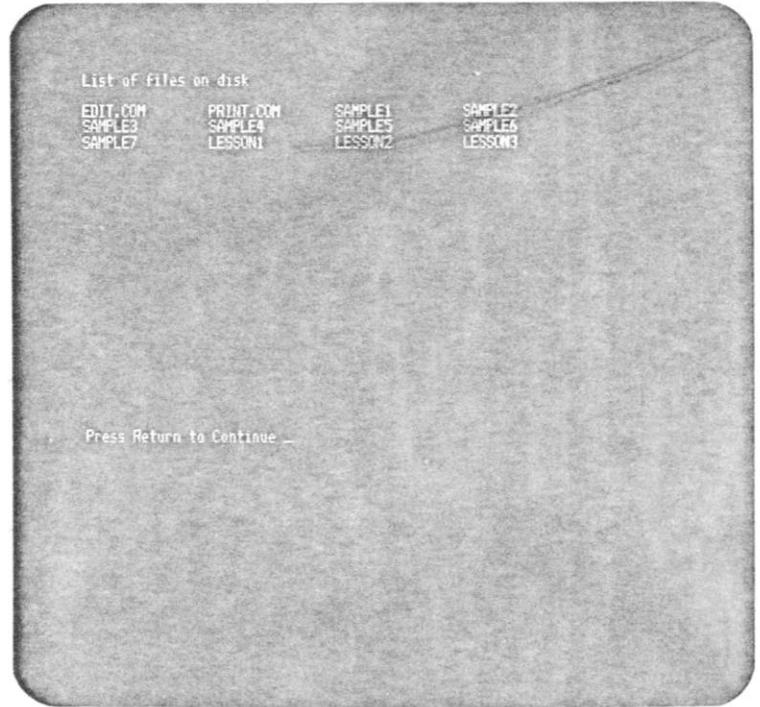
1. Date (**Jan. 14, 1865**) at the first exclamation point.
2. Name and address at the next exclamation point:
Gen. Geo. A. Custer
Seventh Cavalry
Army of the Potomac
Near Richmond, Virginia.
3. Salutation (**Dear George,**) at the next.



FILES/DISPLAY COMMANDS

The FILES command allows you to determine the contents of a disk while using EDIT. The DISPLAY command allows you to display the contents of a text file on the screen without exiting EDIT.

- A. Press ESCAPE to get to the Command Screen.
- B. Type F (Files).
- C. The machine clears the screen and prints a list of the files on the disk on DRIVE A.
- D. Press RETURN to return to the Command Screen.
- E. Type DREPLIES.GET (Display file REPLIES.GET).
- F. The machine prints the first screen of file REPLIES.GET.
- G. On the message line at the bottom of the screen the machine prompts you to press RETURN to display the next screen, or ESCAPE to return to the Command Screen.
- H. The machine automatically returns to the Command Screen when you have displayed the entire file.
 - 1. Using the FILES command, make sure that the file REPLIES.GET (a series of standard responses to inquiries about the Gettysburg Address) is on DRIVE A. (Type F)
 - 2. Use the DISPLAY Command so you can see what is in this file. (Type DREPLIES.GET)
 - 3. Note: Automatic wordsplitting does not function during DISPLAY or INSERT, so the machine may break lines in the middle of words. This does not change how they will look when printed.
 - 4. When you have completed the file, return to the Command Screen.



INCLUDE COMMAND

The INCLUDE command allows you to merge part or all of one file with the file you are editing. It is used most frequently to "boiler-plate" a letter, i.e., to include stock paragraphs from a formatted file in a personalized letter, as in the current example.

- A. While on the Command Screen, type **IREPLIES.GET** to establish REPLIES.GET as an INCLUDE file.
- B. The machine reprints the Command Screen indicating that Including is now active for REPLIES.GET.
- C. Type **I** (to begin the INCLUDE sequence).
- D. The machine displays the first screen of the Include file.
- E. **Note:** You may format an INCLUDE file by sections to which you can refer to by code or section name. (See below)
- F. The machine gives you an option (on the message line) to include this screen in the text you are editing (by pressing **Y**) or to skip it (by pressing RETURN).
- G. The machine then gives you an option to continue with the next screen (by pressing RETURN), to jump to a particular section (by typing the section code followed by RETURN), or to return to the Command Screen (by pressing ESCAPE).
- H. The text is included at the current cursor location in the file you are editing. (Similar to block manipulation).
 1. On the Text Screen move the cursor to the next exclamation point.
 2. Delete the exclamation point so that it will not appear in the final letter. (Use the CHARACTER DELETE Control Key)
 3. Establish REPLIES.GET as an INCLUDE file. (ESCAPE to the Command Screen and type IREPLIES.GET)
 4. Begin the INCLUDE Sequence (Type **I**) and include the paragraph entitled **A-1 — Influential Supporter** in the current letter. (Type **A-1** when the machine requests a section name. When the machine prints that section on the screen type **Y**.)
 5. Return to the Command Screen (press ESCAPE) and print the letter (Type **P**). **Note:** Make sure your printer is ready.
 6. Type **QUIT** to exit EDIT and return to CP/M. (**Note:** By typing QUIT instead of END, the LESSON3 file is not saved, and SAMPLE2 remains intact. See **NOTES** for complete discussion.)

STANDARD REPLIES TO LETTERS ABOUT THE GETTYSBURG ADDRESS

The following paragraphs are standard responses to letters referring to the Gettysburg Address. Use the following codes for direct access to the sections.

A-1 Influential Supporter	Z-1 Important Member of Opposition
A-2 Important Dignitary	Z-2 Unfriendly Citizen
A-3 Friendly Citizen	Z-3 Threatening Citizen
A-4 Schoolchild	

Press Return to Continue, Y to Accept.

```

Seventh Cavalry-
Army of the Potomac-
Near Richmond, Virginia-
~
Dear George-
~
Your interest in our little speech is highly
gratifying, especially in light of the great work that you
have done on your own to advance the great cause to which
we have committed ourselves. I am confident that you will
once again be prepared should your nation call on you to
sacrifice personal comfort for the greater good."
Since you did express some liking for our few words at
Gettysburg, I have taken this opportunity to send you a
copy of it in our own hand."
Until the opportunity of seeing you once again, I
remain,
~
Your servant,
~
A. Lincoln-

```

HELP FILES

- A. A HELP file is any file that you design to be used for reference instead of text generation.
- B. You may use either the DISPLAY or INCLUDE command to access a HELP file. (DFILENAME or IFILENAME).
- C. If you have arranged your HELP file in sections, use the INCLUDE Sequence.
- D. If the file is not arranged in sections, it is faster and easier to use the DISPLAY command.
- E. **Note:** Since automatic wordsplitting does not function in INCLUDE or DISPLAY, you should format HELP files with carriage returns to break the lines.
 1. Edit SAMPLE2 and save as LESSON3B (Type EDIT SAMPLE2 LESSON3B).
 2. Create a letter with the following information:
 November 16, 1864
 Monsignor Patrick Sheehan of
 Trinity Church in Boston, Mass.
 Use A-2 — Important Dignitary paragraph. (Remember to use LESSON3A, the file you just updated, instead of REPLIES.GET.)
 3. Using the INCLUDE sequence, access SALUTES (a file of proper forms of address and salutation for dignitaries) to determine how to address a Monsignor. (Remember: press ESCAPE to enter the Command Screen; type ISALUTES to designate the INCLUDE file; type I to begin the sequence, etc.) Do not insert any of the file.
 4. **Note:** As soon as you activate LESSON3A as an INCLUDE file, SALUTES is inactivated.
 5. Print the letter with a line length of 60 characters.
 6. Exit EDIT by typing END.

Table of Addresses and Salutations for Government Church and School Dignitaries

This is a file of frequently encountered dignitaries and how properly to address them. The left hand column contains the person's title; the center column has the correct way(s) to address correspondence to the person; the right hand column has the acceptable salutations, ranging from the most formal to most casual.

If this is an INSERT file, access by section using the following codes:

A—Ambassador thru Archbishop	H—Mayor thru Monsignor
B—Bishop thru Brother	P—The Pope thru Priest
Ca—Cabinet thru City Council	R—Rabbi thru Rector
Co—Congressman thru Consul	S—Senator thru Vice-President
D—Dean thru Judge	

Press Return to Continue, Y to Accept

The White House
1600 Pennsylvania Ave.
Washington, D. C.
November 16, 1864

The Right Reverend
Monsignor Patrick Sheehan
Trinity Church
Boston, Mass.

Dear Monsignor:

We are pleased that you have taken time out of your busy day to write us concerning our speech at Gettysburg. Although your interest is gratifying, we are much more gratified by the leadership you give to our people. We could not hope for victory on the battlefield did we not have individuals like you who keep the home fires burning.

We hope you will like the facsimile of our speech which we enclose. Should you ever be in Washington, we hope you will call at the White House so that a member of our staff can personally guide you through our humble abode.

Very truly yours,

Abraham Lincoln

AL/mw
encl.

REVIEW

Using everything you have learned in this lesson, create letters with the following information. You may skip this section if you wish, but since you will probably be creating many boilerplate letters, you need to understand the process completely before you go on.

1. **FILES**
Text: **SAMPLE2**
Include **LESSON3A**
HELP: **SALUTES**
2. **LETTER 1**
June 14, 1865
Patrick MacFarland (The mayor of
Worcester, Mass)
J-1
3. **LETTER 2**
Oct 19, 1864
Lisa Jeffreys, age 9
Wilson School
Fort Bend, Indiana
A-4 — Schoolchild
4. **LETTER 3**
Robert Johnson
123 Annapolis Street
Baltimore, MD
Z-2 — Unfriendly Citizen
5. After you create each letter, print a copy of it and type **QUIT** to exit the Editor.

INTRODUCTION TO PRINT FEATURES

The PRINT program takes a file that you created with EDIT and processes it for printing by using special commands that you give it. You may embed these commands directly in the text file or give them through the keyboard as you print the file.

Some of the commands control the *shape* of the text, how it is placed on the page, the margins, spacing, etc. These commands give you complete control of your printer, so that you can make it do things you never thought possible.

Other commands determine *what* is printed. You can access external data files, e.g., name and address lists. You can create fill-in-the-blank files that you fill in automatically as you print. You can create a file with commands that are executed only if certain conditions are met, so that one file can print several different letters.

All of these features can save you a lot of time, but to use them, you need to understand how the commands work. Some of the commands are easy and can be mastered in a matter of minutes. Others are more complicated, but if you take your time and go through the lessons slowly, you should be able to master them as well.

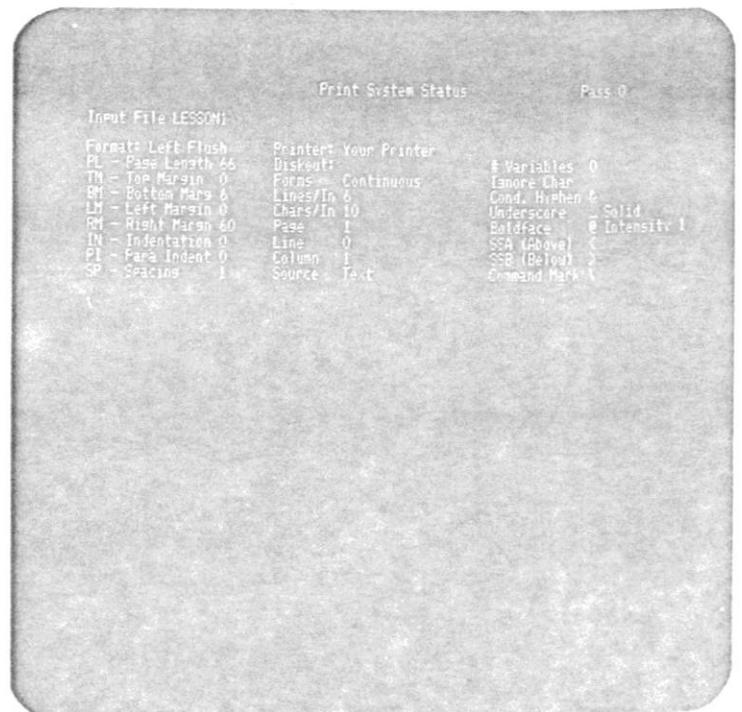
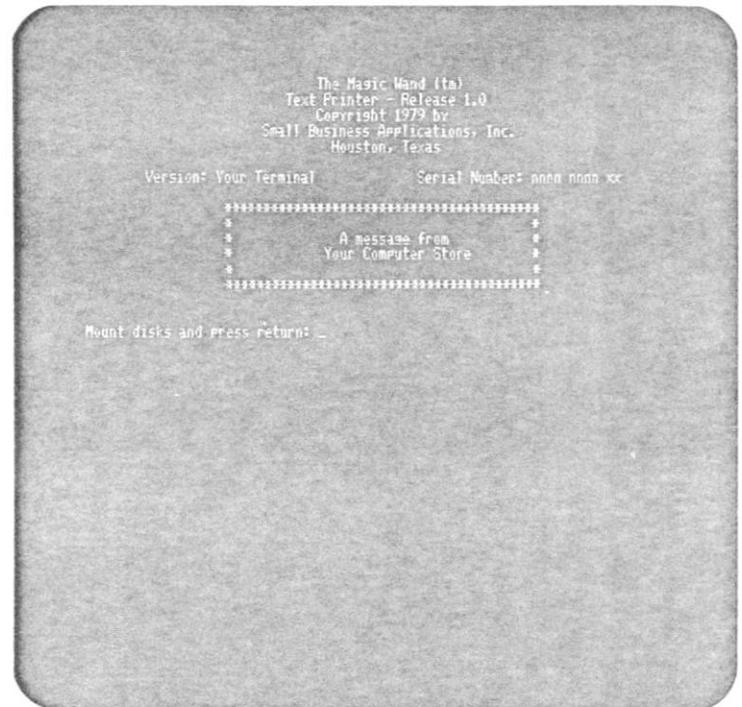
Lesson 4 deals with the formatting commands, what they do and how to give them from the keyboard. Lesson 5 shows you how to embed these commands in your text file and how to use special print features. Lesson 6 covers variables and fill-in-the-blank documents. Lesson 7 teaches you how to access external data files. Lesson 8 discusses conditional commands. Lesson 9 covers multi-page documents.

All of these lessons use up a lot of paper, so load up your printer and let's go.

LESSON FOUR

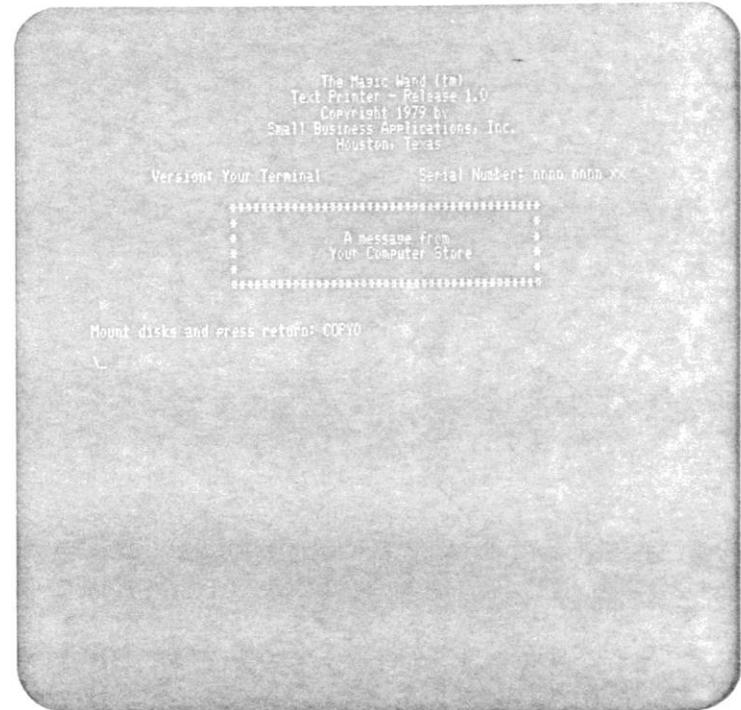
LOADING THE PRINT PROGRAM

- A. Turn on your computer, load CP/M and make sure that your printer is ready.
 - B. Place the **MAGIC WAND** disk in the current drive. (DRIVE A)
 - C. Type **PRINT LESSON1**. (Remember to press **RETURN**.)
 - D. The machine prints the sign-on screen then tells you to mount disks and press **RETURN**.
 - E. Before pressing **RETURN**, type **DS** (**Display Status**).
 - F. The machine prints the Status Screen (at right).
- Note:** You will know what all of this screen means by the time you complete the lessons.
- G. Then the machine prints a back slash (\) indicating it is waiting to receive a command.
 - H. Press **RETURN** without entering a command to start printing.
 - I. When the printing is complete, the machine writes **end of job** on the screen and returns to CP/M.
 1. Prepare to print **LESSON1**. (Type **PRINT LESSON1**.)
 2. Type **DS** when told to mount disks.
 3. Press **RETURN** (without entering a command) to see what the file looks like using the default values for the print commands.



GIVING COMMANDS FROM THE KEYBOARD

- A. When the machine prints a back slash on the screen, type a command and press RETURN.
- B. The machine prints another back slash indicating it has accepted your command and is ready for another one.
- C. If you type an invalid command, the machine refuses it and prints an error message on the screen.
- D. When you have given all of the commands you want to give, press RETURN without giving a command.
- E. You may give more than one command on the same line if you separate them with commas.
- F. Pressing any key on the keyboard while printing is in progress interrupts the printer.
- G. When this happens, the machine prints a back slash and awaits your command as it did previously.
- H. Press RETURN without entering a command to resume printing.
- I. **Note:** Many typewriters print from a small internal memory, so they may not stop immediately when you interrupt the printer.
 1. Prepare to print LESSON1 again. (Type **PRINT LESSON1**.)
 2. When told to mount disks, type **COPY0** and press RETURN. (Now when you finish printing the file, instead of returning to CP/M, the machine will ask you whether you wish to print another copy.)



LEFT MARGIN

- A. When the back slash appears, type **LM10** (Left Margin) (Remember to press RETURN).
- B. The machine sets the left margin to begin ten spaces from the extreme left of the printer.
- C. Type **DS**.
- D. The machine reprints the Status Screen indicating that the left margin (LM) is now 10.
 1. Set the left margin at 10 characters and begin to print the file (Type **LM10** and press RETURN, then press RETURN again).
 2. Interrupt the printer, type **LM15**, and resume printing. (Press any key; and when a back slash appears, type the command followed by *two* carriage returns — one to enter the command and one to resume printing.)
 3. Interrupt the printer again, type **LM5**, and finish printing.
 4. Notice how the LM command positions the text on the page without affecting the width of the line.
 5. Use the **DS** command frequently in this and subsequent sections to see how the machine records your commands.

```

Print System Status          Pass 1

Input File LESSON1
Format: Left Flush
PL - Page Length 66
TM - Top Margin 0
BM - Bottom Marg 6
LM - Left Margin 10
RM - Right Margin 60
IN - Indentation 0
PI - Para Indent 0
SP - Spacing 1

Printer: Your Printer
Diskout:
Format: Continuous
Lines/In 6
Chars/In 10
Page 1
Line 0
Column 1
Source Text

# Variables 0
Ignore Char
Cond. When 6
Underscore Solid
BoldFace Intensity 1
SSA (Above)
SSB (Below)
Command Mark \

\
\LM15.

```

RIGHT MARGIN

- A. When the machine asks you if you want to print another copy, press **Y**.
- B. Immediately, before the printing starts, press any key on the keyboard to interrupt.
- C. When the machine prints a back slash, type **RM60**. (Right Margin)
- D. The machine sets the right margin sixty characters from the left margin.
 1. Begin to print the file with a right margin of 60 characters. (Type **RM60**)
 2. Interrupt the printer, change the right margin to 50 characters and resume printing. (Type **RM50**)
 3. Interrupt the printer again, change the right margin to 40 characters and finish printing. (Type **RM40**)
 4. **Note:** You do not need to wait for the printer to stop printing before you type in your commands. If the printer has a full memory when you interrupt, you may be able to give a command and resume printing without stopping the printer.

```

Print System Status          Pass 1

Input File LESSON1
Format: Left Flush
PL - Page Length 66
TM - Top Margin 0
BM - Bottom Marg 6
LM - Left Margin 5
RM - Right Margin 60
IN - Indentation 0
PI - Para Indent 0
SP - Spacing 1

Printer: Your Printer
Diskout:
Format: Continuous
Lines/In 6
Chars/In 10
Page 1
Line 0
Column 1
Source Text

# Variables 0
Ignore Char
Cond. When 6
Underscore Solid
BoldFace Intensity 1
SSA (Above)
SSB (Below)
Command Mark \

\
Ready to Print (Y/N)? y
\RM50
\

```

LEFT AND RIGHT FLUSH

- A. You have been printing using a *flush left* format, i.e., the left margin is even (flush) but the right margin is uneven. (This is also called *ragged right*.)
- B. Type **RIGHT** to change it to a flush right format, i.e., the right margin is even while the left is uneven. (Also called *ragged left*.)
- C. To change back to flush left, type **LEFT**.
1. Begin to print LESSON1 with a left margin of ten spaces, a right margin 55 spaces from the left margin with a flush right format. (After beginning the next copy of the file, interrupt the printer and type **LM10, RM55, RIGHT**.)
 2. Interrupt the printer again, change the format to flush left, and finish printing. (Type **LEFT**)

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

INDENTATION

- A. Type **IN10**. (INdent)
- B. The machine establishes a temporary left margin 10 spaces from the permanent left margin established with the **LM** command.
- C. Note: The **IN** command does not affect the right margin, i.e., if you have a right margin of 60 and an indentation of 10, your actual line length is 50 characters (sixty minus ten).
1. Begin to print this copy with a left margin of ten spaces and a right margin of 60 spaces. (Type **LM10, RM60**)
 2. Interrupt printing, set an indentation of ten characters, and resume printing. (Type **IN10**)
 3. Interrupt printing again, change the indentation to 15 characters and the right margin to 55 character and resume printing. (Type **IN15, RM55**.)
 4. Interrupt printing again, reset the original format, and finish printing. (Type **IN0, RM60**)

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

PARAGRAPH INDENTATION

- A. Type **PI5**. (Paragraph Indentation)
- B. Machine sets automatic paragraph indentation to five spaces from the current left margin, i.e., the left margin plus any indentation.
- C. Since the paragraphs in the **LESSON1** file are already indented five spaces, they will now be indented ten spaces.
- D. The paragraph indentation only applies to the first line immediately following a carriage return.
- E. You may, if you choose, have negative indentation, in which case the first line following a carriage return is longer than the following lines.
- F. The machine does not allow you to print to the left of the margin established by the **LM** command. If you want negative paragraph indentation, you must first give an **IN** command greater than or equal to the number of extra spaces in the paragraph indentation.
 1. Print **LESSON1** with a left margin of 10, a right margin of 55 and a paragraph indentation of 5. (Type **LM10, RM55, PI5**).
 2. Print **LESSON1** again with a left margin of 5, a right margin of 60, an indentation of 5, and a paragraph indentation of -5. (Type **LM5, RM60, IN5, PI-5**). This compensates for the indentation in the file.
 3. Note: If you type **PI-5** without first typing **IN5**, the machine gives you an error message and refuses to carry out the command.
 4. Print **LESSON1** again with a left margin of zero, a right margin of 65, an indentation of 10, and a paragraph indentation of -10. (Type **LM0, RM65, IN10, PI-10**). This is called a "hanging" paragraph.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

JUSTIFIED TEXT

- A. A text is in *justified* format when both its left and right margins are even.
- B. The machine does this by adding extra spaces to the line.
- C. The machine may add the extra space only to the spaces between the words. This is called *blank insertion*.
- D. On a standard printer, the extra space is inserted as character-sized blanks. On specialty printers, the extra space is divided evenly among the spaces between all of the words.
- E. To change to a justified format using blank insertion, type **JUST**. (**JUST**ified)
- F. If you have a specialty printer, you may choose to justify by inserting very small spaces between the characters themselves. This is called *character spreading*.
- G. To change to a justified format using character spreading, type **JUSTC**. (**JUST**ified **Character** spreading)
- H. **Note:** Tastes vary. If you have a specialty printer, you will need to decide for yourself which justified format you prefer.
 1. Print LESSON1 with a left margin of 10 and a right margin of 55, justified using blank insertion. (Type **LM10**, **RM55**, **JUST**.) [Remember to change **PI** and **IN** to zero.]
 2. If you have a specialty printer, repeat using character spreading. (Type **LM10**, **RM55**, **JUSTC**.)
 3. **Note:** You can make justified text look more even by using hyphenation, which is covered in Lesson 5.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate--we can not consecrate--we can not hallow--this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us--that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion--that we here highly resolve that these dead shall not have died in vain--that this nation, under God, shall have a new birth of freedom--and that government of the people, by the people, for the people, shall not perish from the earth.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate--we can not consecrate--we can not hallow--this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us--that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion--that we here highly resolve that these dead shall not have died in vain--that this nation, under God, shall have a new birth of freedom--and that government of the people, by the people, for the people, shall not perish from the earth.

CENTERED AND LITERAL TEXT

- A. To center a line, type **CTR.** (CenTeR)
- B. The machine prints the next line in the center of the line with an equal number of spaces to the left and right.
- C. The machine positions the line based on the current line length, i.e., the right margin minus the indentation.
- D. The CTR command applies only to the next line following the command, after which the machine returns to the previous format (left, justified, etc.).
- E. To inactivate automatic wordsplitting, type **LIT.** (LITeral) (You will probably not need to use this command unless you do some unusual printing.)
 1. Print LESSON1 with a left margin of 10, a right margin of 55, with the top line centered and the rest of the text in a literal format. (Type LM10, RM55, LIT, CTR.)
 2. **Note:** You may need to start the PRINT program from scratch in order to center the first line. To do so, press N when the machine asks you if you are ready to print. Then type **PRINT LESSON1.** When the machine pauses, type in the commands and press RETURN.

SPACING

- A. Type **SP2.** (SPace)
- B. The machine prints out the text with double-spacing.
- C. If you have a specialty printer, type **SP+1.**
- D. The machine prints out the text with one-and-a-half spacing.
 1. Begin printing LESSON1 with a left margin of 5, a right margin of 55, justified text, the first line centered and double-spacing. (Type LM5, RM55, JUST, CTR, SP2.)
 2. Interrupt the printing, change to triple-spacing and resume printing. (Type SP3.)
 3. If you have a specialty printer, interrupt the printing, change to one-and-a-half spacing and resume printing. (Type SP+1)
 4. If you have a specialty printer, interrupt the printing again, change to two-and-a-half spacing, and finish printing. (Type SP+2)

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract.

The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so

dedicated can long endure. We are met on a great

battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

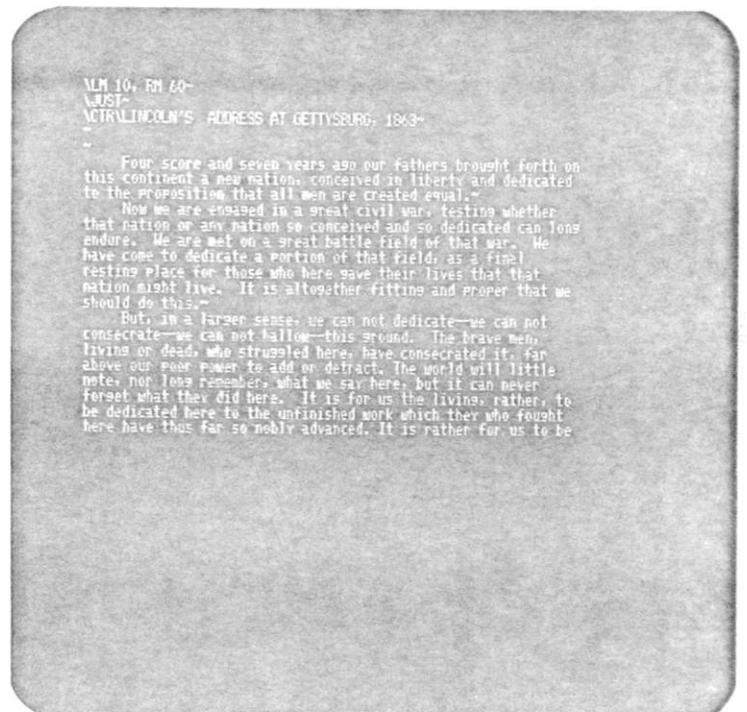
But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

LESSON FIVE

EMBEDDED PRINT COMMANDS AND
SPECIAL PRINT FEATURES

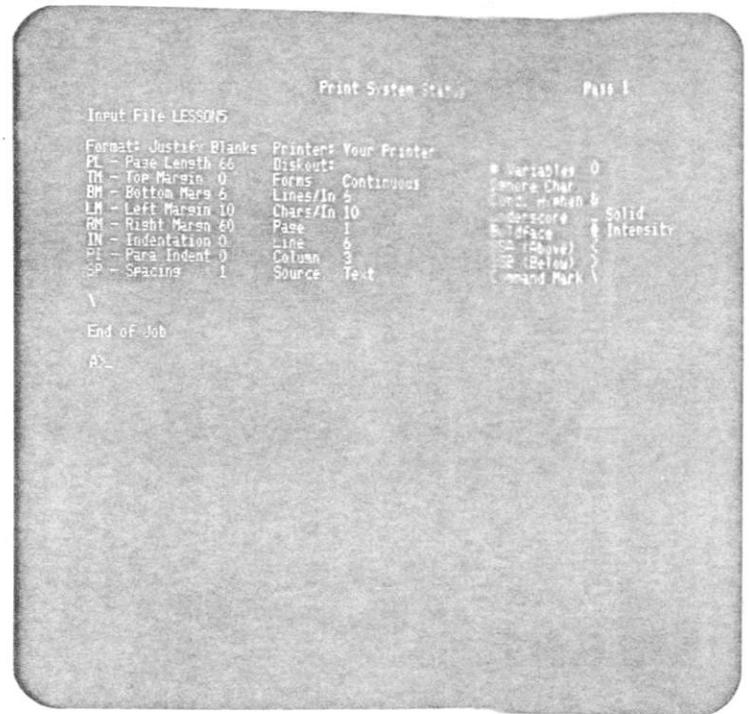
HOW TO EMBED A COMMAND

- A. Edit file LESSON1 to be saved as LESSON5. (Type **EDIT LESSON1 LESSON5**)
- B. When you first get to the Text Screen, the cursor is in the first position of the first line.
- C. Press the FULL INSERT Control Key to open the file at the beginning.
- D. Type a back slash (\) to begin the command line
- E. Type in the commands you wish to make and press RETURN.
- F. You may put all of your commands on one line separated by commas, or on different lines as long as each line begins with a back slash.
- G. Press the END INSERT Control Key.
- H. The cursor is now at the beginning of the title line.
- I. Using the CHARACTER INSERT Control Key, insert \CTR\ in front of "LINCOLN'S..."
- J. Return to the Command Screen and type **END**.
 1. Embed these commands following the above instructions:
 Left Margin — 10 characters (**LM10**)
 Right Margin — 60 characters from left margin (**RM60**)
 Justified Copy (**JUST**)
 Center the title (**CTR**)



HOW THE MACHINE INTERPRETS A COMMAND

- A. Type **PRINT LESSON5**.
- B. The machine loads **PRINT**
- C. After you mount disks and press **RETURN**, the machine begins to read **LESSON5**
- D. When the machine encounters the backslash in the first line, it interprets everything that follows on that line as a command.
- E. The machine does not print the command; as far as the printer is concerned, the command is invisible.
- F. The machine treats everything following a backslash as a command until it comes to a carriage return or another backslash, whichever comes first.
- G. At the beginning of the text, the machine processes all of the commands it finds until it reaches the first *text* character, in this case the **L** in **LINCOLN**.
- H. The machine prints the text using the current print parameters, including the commands that you embedded in the text.
 1. Follow the above procedure. (Type **PRINT LESSON5**)
 2. Press **RETURN** without entering a command.
 3. After the machine starts to print the file, interrupt the printing and display the Status Screen. (Press any key, and type **DS**.)
 4. Notice how the Status Screen reflects the commands you embedded.
 5. Print the remainder of the file. (Press **RETURN**)



EMBEDDING ADDITIONAL COMMANDS

- A. Reload LESSON5 into the Editor. (Type **EDIT LESSON5.**)
- B. Move the cursor to the beginning of the first paragraph and press RETURN.
- C. The machine inserts a line in the text and moves the remainder of the text down one line.
- D. Type a back slash followed by commands and press RETURN.
- E. Exit EDIT using **END**
 1. Before the first paragraph insert commands to set a paragraph indentation of 5 characters and spacing at 1½ spaces [two for standard printers]. (Type **\PI5,SP+1** [or **SP2**] and press RETURN.)
 2. Between paragraphs one and two, insert commands to indent 5 characters, move the right margin in 5 characters and change to single spacing. (Type **\IN5,RM-5,SP1** and press RETURN.)
 3. **Note:** In most formatting commands, if you put a plus or minus sign in front of the number, the machine adds or subtracts the number from the current value. E.g., if you give the command **RM-5** when the right margin is 60 characters, the machine sets the right margin as 55 characters (60 minus 5).
 4. Between paragraphs two and three insert commands to stop indentation, return the right margin to its original position and reset the spacing to 1½. (Type **IN0,RM+5,SP+1** [or **SP2**] and press RETURN.)
 5. End EDIT.
 6. Print LESSON5.

```

\LM 10, RM 60-
\JUST-
\CTRLINCOLN'S ADDRESS AT GETTYSBURG, 1863-

```

```

\PI5,SP2-
Four score and seven years ago our fathers brought forth on
this continent a new nation, conceived in liberty and dedicated
to the proposition that all men are created equal.
\INS, RM-5, SP1-
Now we are engaged in a great civil war, testing whether
that nation or any nation so conceived and so dedicated can long
endure. We are met on a great battle field of that war. We
have come to dedicate a portion of that field, as a final
resting place for those who here gave their lives that that
nation might live. It is altogether fitting and proper that we
should do this.
\IN0, RM+5, SP1-
But, in a larger sense, we can not dedicate—we can not
consecrate—we can not hallow—this ground. The brave men,
living or dead, who struggled here, have consecrated it, far
above our poor power to add or detract. The world will little
note, nor long remember, what we say here, but it can never

```

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

NOTE COMMANDS

- A. While editing a text file, type *
- B. The asterisk line is an internal note for the person who is editing the file.
- C. When executing the PRINT program, the machine ignores everything from the asterisk until the next carriage return or back slash, whichever comes first.
- D. Note: You cannot give additional commands on the same line following an asterisk.
- E. While editing a file, type \NOTE
- F. When executing the PRINT program, the machine prints everything between NOTE and the next carriage return (or back slash) on the screen but not on the printer.

Note: As above, the NOTE command must be the last command on a command line.

1. While editing LESSON5, open the top of the text using the FULL INSERT Control Key.
2. On the first line type *Date of last revision — Nov. 18, 1863 and press RETURN.
3. On the next line type \NOTE Lincoln's address as given at Gettysburg, November 19, 1863 and press RETURN.
4. Press the END INSERT Control Key to close the text.

```

\* Date of last revision — Nov 18, 1863
\NOTE Lincoln's address as given at Gettysburg November 19, 1863
\LN 10, RM 60
\JUST
\CTRLINCOLN'S ADDRESS AT GETTYSBURG, 1863
~
\PI5,SP2
Four score and seven years ago our fathers brought forth on
this continent a new nation, conceived in liberty and dedicated
to the proposition that all men are created equal.
\LN7,RF-5,3PI
Now we are engaged in a great civil war, testing whether
that nation or any nation so conceived and so dedicated can long
endure. We are met on a great battle field of that war. We have
come to dedicate a portion of that field, as a final resting
place for those who here gave their lives that that nation might
live. It is altogether fitting and proper that we should do
this.
\LN0,RM45,SP41
But, in a larger sense, we can not dedicate—we can not
consecrate—we can not hallow—this ground. The brave men,
living or dead, who struggled here, have consecrated it, far

```

GHOST HYPHENATION

Definition: A Ghost Hyphen is a special character (called a recognition character) that indicates where a word may be hyphenated if it would not otherwise fit on a line. The machine ignores the special character except when it makes a hyphenation decision. (Also called a *discretionary* or *conditional* hyphen.)

- A. The machine recognizes an ampersand (&) as a Ghost Hyphen.
- B. You have the word **con&ceived** in your text file.
- C. If you print the file from EDIT (by typing P), the machine prints the ampersand.
- D. If you print the file using PRINT, the machine prints **conceived**, ignoring the ampersand.
- E. If the machine cannot fit **conceived** on the line while printing from PRINT, it may print instead **con-** on the first line and **ceived** on the next. (It only does this if **con-** fits on the line.)
- F. **Note:** The machine breaks a word *only* at Ghost Hyphens. If you have a hyphenated word (e.g., **mother-in-law**) that may be broken at the end of a line, you must first insert Ghost Hyphens (e.g., **mother-&in-&law**).
 1. Before you place any Ghost Hyphens in the LESSON1 file, print three copies of the file, one with a line length of 60 characters (RM60), one with a line length of 58 characters (RM58) and one with a line length of 55 characters (RM55). The format should be left flush (LEFT). (Type in the necessary commands from the keyboard.)
 2. Load LESSON1 into EDIT. (Type EDIT LESSON1) Leave the line length at the maximum for your terminal (probably 80 characters).

(Sample print at Right Margin 58)
LINCOLN'S ADDRESS AT GETTISBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

(Sample print at Right Margin 55)
LINCOLN'S ADDRESS AT GETTISBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

3. Using the CHARACTER INSERT Control Key, insert Ghost Hyphens (&) into the text as follows (refer to the screen at right):
 - a. con&ceived and lib&er&ty in the first paragraph.
 - b. wheth&er, ded&i&ca&ted, bat&tles, por&tion and al&to&geth&er in the second paragraph.
 - c. de&tract, lit&tles, re&mem&ber, ded&i&ca&ted, ad&vanced, rath&er, be&fore, us — &that, in&creased, de&vo&tions, meas&ure, high&ly, vain — &that, na&tions, free&dom — &and and gov&ern&ment.
 - d. Use MULTIPLE SEARCH AND REPLACE to change each **people** into **peo&ple**.
4. Exit the Text Screen and type **END**.
5. Using **PRINT**, print **LESSON1** using the same line lengths as before (RM60, RM58, RM55)
6. Notice how the right margins are much more even when some of the words have been hyphenated.
7. **Note:** The number of hyphens (if any) you want to appear in a text is a matter of personal taste.
8. If you expect to use Ghost Hyphens frequently, you may want to repeat this section, making your own decisions about where to place the Ghost Hyphens. If so, load **LESSON1.BAK** into the Editor and save it as **LESSON5A** (Type **EDIT LESSON1.BAK LESSON5A**).

(Sample print at Right Margin 58)
LINCOLN'S ADDRESS AT GETTYSBURG, 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

(Sample print at Right Margin 55)
LINCOLN'S ADDRESS AT GETTYSBURG, 1863

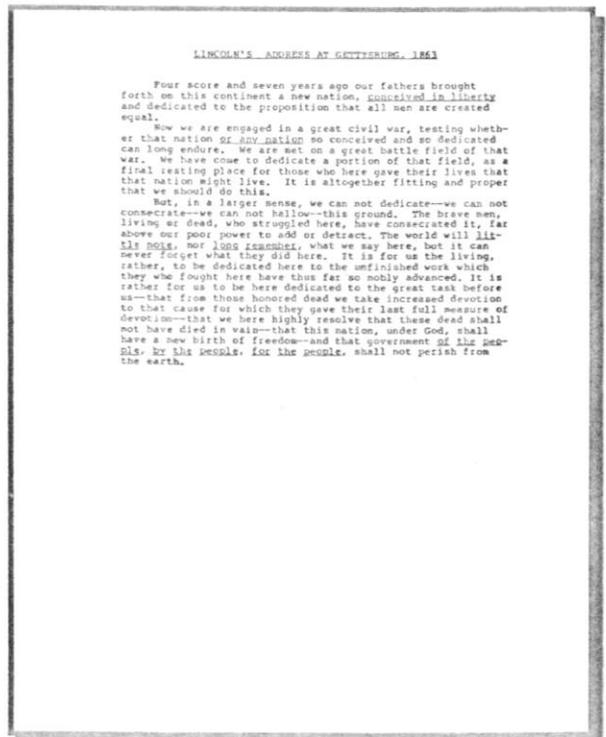
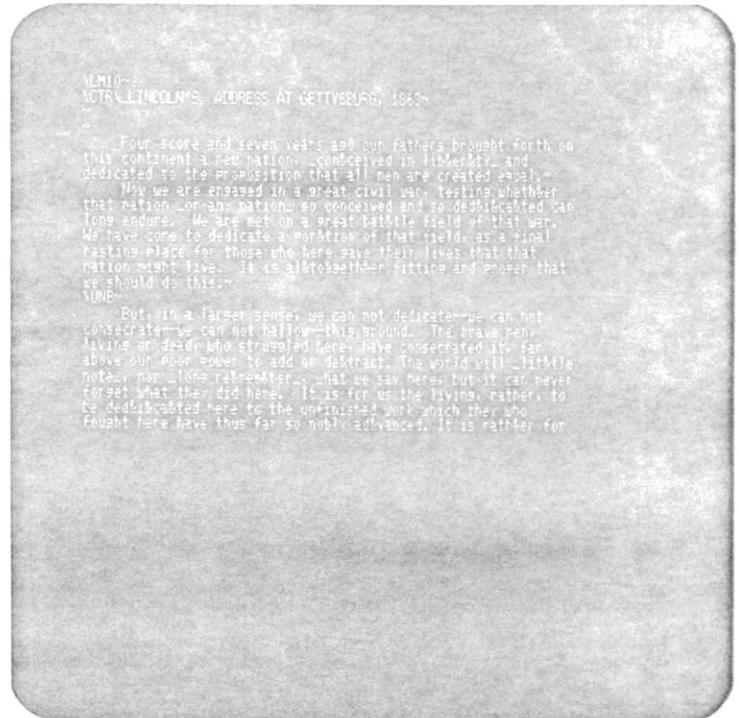
Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

UNDERSCORING

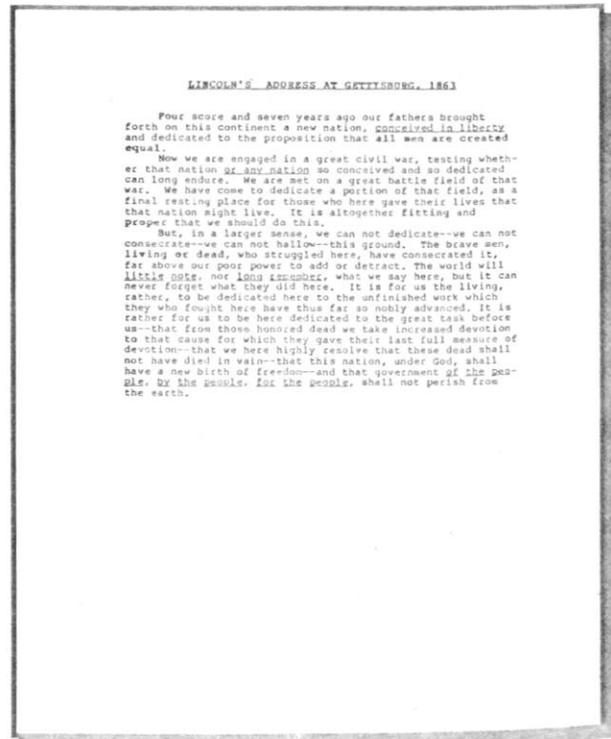
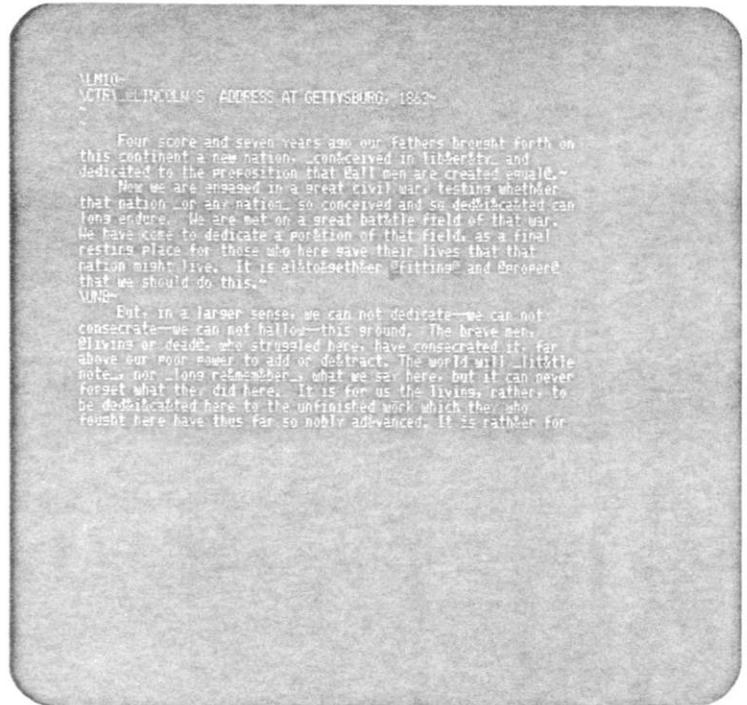
- A. As in ghost hyphenation, you use a special character (a *recognition* character) to mark off the text that you want to underscore.
- B. The special character is an **underline** character ().
- C. If you are using PRINT and the machine encounters an underline character, it begins underscoring at the next character.
- D. The machine prints the entire line and then goes back to underscore those characters that you marked for underscoring.
- E. The machine continues to underscore until it encounters another underline character or a carriage return, whichever comes first.
- F. Unless you tell it otherwise, the machine performs solid underscoring, i.e., it underscores all of the characters from the first underline character to the next, including blanks and punctuation.
- G. If you want *broken* underscoring, type **UNB** (**UN**der**score****Broken**).
- H. Now, the machine only underscores letters and numbers, no blanks or punctuation marks.
- I. To change it back to **solid** underscoring, type **UNS** (**UN**der**score** **Solid**).
 1. Edit LESSON1.
 2. Establish a left margin of 10, a right margin of 60 and a flush left format.
 3. Center and underscore the title. (Using the CHARACTER INSERT Control Key, insert `\CTR\ _` in front of LINCOLN'S. Since the line ends in a carriage return, you do not need to insert another underline character.)
 4. Set the text so that the machine underscores **conceived in liberty** in the first paragraph. (Insert an underline character in front of **con&ceived** and following **lib&er&ty**.)
 5. Underscore **or any nation** in the second paragraph.
 6. Between the second and third paragraphs, change to broken underscoring. (Type **UNB** and press RETURN.)
 7. Underscore **little note** and **long remember** in the third paragraph.
 8. Underscore **of the people, by the people, for the people**. (Insert an underline character in front of the first **of** and after the last **peo&ple**.)
 9. End EDIT and print LESSON1.



BOLD FACE

Applicable only to specialty printers

- A. You designate bold face the same way as underscoring, i.e., by marking the text with a special (recognition) character.
- B. The special character for bold face is the at-sign (@).
- C. When the machine encounters an at-sign, instead of moving a full space after striking the character, it moves only a fraction of a space and strikes the character again.
- D. Characters in bold face take up slightly more room on the page than regular characters.
- E. It may take the machine considerably longer to print characters in bold face.
- F. You may set the bold face from one additional strike of the character to as many as nine additional strikes.
- G. As you add more strikes, the character becomes thicker, takes up more room, and takes the machine longer to print.
- H. The machine is initially set to bold face using one additional strike.
- I. To change the number of strikes, type **BF n** , where n is a number from one to nine, e.g., to change the number of additional strikes to three, type **BF3**.
 1. Edit LESSON1.
 2. Bold face the title as well as underscoring it. (Using the CHARACTER INSERT Control Key, insert an at-sign behind the underline character (_@).
 3. Bold face **all men are created equal** in the first paragraph. (Insert an at-sign in front of **all** and following **equal**.)
 4. Bold face **fitting and proper** in the second paragraph.
 5. Bold face **living or dead** in the third paragraph.
 6. End EDIT and print LESSON1.
 7. Print LESSON1 again, but this time change the additional strikes to two. (Before beginning to print the file, type the command **BF2**.)
 8. Repeat using three additional strikes (**BF3**), etc., until you get a feeling for the different intensities of bold face.



SUBSCRIPTING/SUPERSCRIPTING

Applicable only to specialty printers

- A. As in underscoring and bold face, you use recognition characters to tell the machine to subscript or superscript.
- B. The recognition characters are a greater-than character (>) for subscripting and a less-than character (<) for superscripting.
- C. When the machine encounters the subscripting recognition character (>), it moves the carriage down half a line.
- D. This movement is permanent, unless you move the carriage back up with a superscripting recognition character (<).
- E. Superscripting is exactly the reverse of subscripting.
- F. Note: This is probably not something that you need every day, but when you need it you need it.
- G. Note: Superscripting is the only way you can move the carriage up the page.
 1. Edit LESSON1.
 2. Superscript ADDRESS in the title line.(<ADDRESS>)
 3. Subscript GETTYSBURG (>GETTYSBURG<)
 4. End EDIT and print LESSON1.

ALM10-
 ADDRESS IN INCLIN'S ADDRESS AT GETTYSBURG, 1863-

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal. Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for

LINCOLN'S ADDRESS AT GETTYSBURG 1863

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battle field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task before us—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.

CHANGING RECOGNITION CHARACTERS

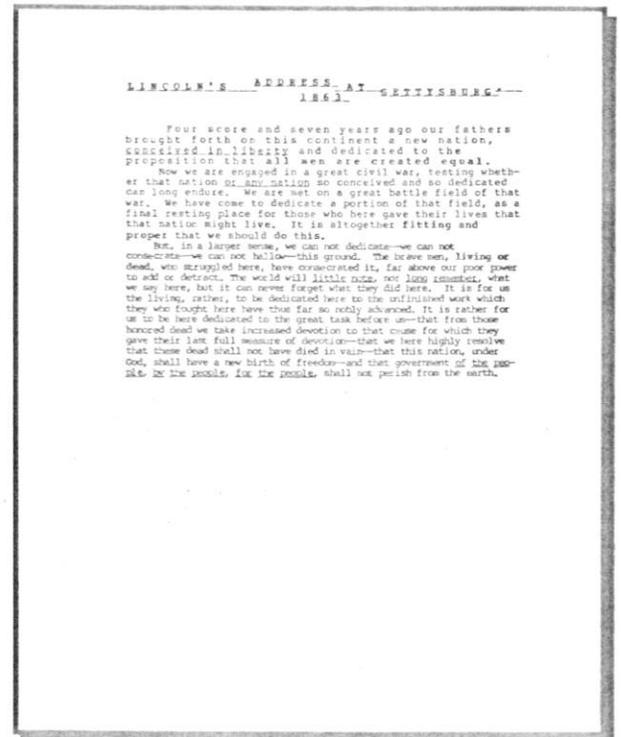
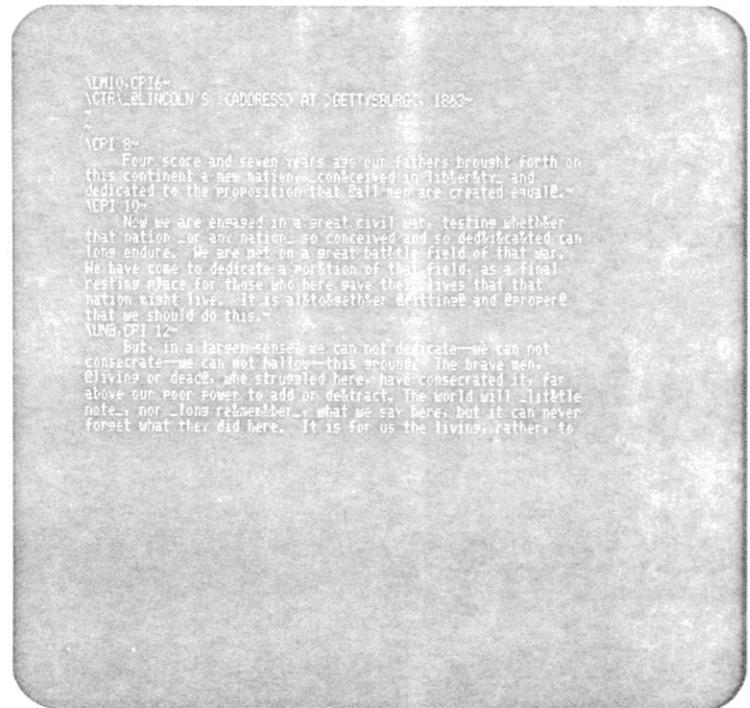
- A. You have learned six recognition characters so far: the back slash (\) (command marker), the ampersand (&) (Ghost Hyphen), the underline character () (underscoring), the at-sign (@) (bold face), the greater-than character (>) (subscripting), the less-than character (<) (superscripting).
- B. The machine does not print any character being used as a recognition character.
- C. If you want to print one of the recognition characters, you must first inactivate or change it.
- D. The commands to inactivate or change the recognition characters are as follows:
- | FUNCTION | INACTIVATE | CHANGE |
|----------------|------------|---------------------|
| Command Marker | CMD | CMD <i>c</i> |
| Ghost Hyphen | HY | HY <i>c</i> |
| Underscoring | UN | UN <i>c</i> |
| Bold face | BF | BF <i>c</i> |
| Superscripting | SSA | SSA <i>c</i> |
| Subscripting | SSB | SSB <i>c</i> |
- E. Note: *c* is any non-numeric character, e.g., a letter or, more likely, a symbol.
- F. Note: To reactivate a recognition character, follow the procedure for changing it, e.g., to reactivate the ampersand as the Ghost Hyphen, type **HY&**.
1. Print LESSON1.
 2. Experiment with inactivating and changing recognition characters from the keyboard.

VARIABLE PITCH CONTROL

Applicable for specialty printers only

Definition: Pitch is the number of characters per inch that your printer prints on the page, the standards being ten pitch and twelve pitch.

- A. If you are printing at ten characters per inch (cpi) and wish to change to twelve characters per inch, type **CPI12**.
- B. The machine now prints at 12 cpi unless you change the pitch again.
- C. The machine interprets commands based on the current pitch, e.g., **LM10** in ten pitch gives you a left margin of one inch (10 X 1/10) but in twelve pitch gives you a left margin of only 5/6 inch. (10 X 1/12)
- D. Changing the pitch after a command has been executed, does not effect the command, e.g., if you establish a left margin of one inch by typing **LM10** when the pitch is 10 cpi, the left margin remains one inch even if you change the pitch to 12 cpi.
- E. The machine recalculates parameters based on the current pitch, e.g., you have a left margin of 10 characters at 10 cpi and you change the pitch to 12 cpi. If you then type **DS** to see the current status, the Status Screen reads **LM12**.
- F. You may use the **CPI** command to set the pitch at other than ten or twelve cpi, e.g., **CPI8**, **CPI6**, etc.
- G. **Note:** See the notes for a complete discussion of the variable pitch and variable line commands.
 1. Edit **LESSON1**.
 2. On the first command line, set the pitch for the title at six characters per inch. (Type **CPI6**) (Remember to set your left and right margins before you change the pitch.)
 3. Before the first paragraph, change the pitch to eight characters per inch. (Type **CPI8**)
 4. Change the pitch for the second paragraph to ten characters per inch. (Type **CPI10**)
 5. Change the pitch for the third paragraph to twelve characters per inch. (Type **CPI12**)
 6. End **EDIT** and print **LESSON1**.
 7. Notice how the left and right margins remain the same even though the number of characters per line is different.



LESSON SIX

This lesson deals with variables and how you can use them in a text file. A Variable is something that may have more than one value. The opposite of a Variable is a Constant, which has only one value.

If you have a fill-in-the-blank form, the blanks are equivalent to variables. The *names* of the variables on the form might be Name, Address, Age, Occupation, etc., and the information that you fill in becomes the *value* of the variables.

In this lesson you revise some of the letters you created in Lesson Three. When you are through, you will have created a fill-in-the-blank file that enables you to generate new letters by typing in just a few pieces of information at run time.

Incidentally, any of the commands covered in this lesson may be given from the keyboard as well as being embedded in the text.

NAMING A VARIABLE

- A. The name of a variable may be up to seven characters long.
- B. The name may be any combination of uppercase letters and numbers as long as the first character of the name is a letter.
- C. As with other commands, the machine converts lowercase letters to uppercase before processing.
- D. Examples of legal variable names: **NAME**, **AMTDUE**, **ADDR1**
- E. Examples of illegal names: **1STNAME** (does not begin with a letter), **AMT DUE** (blank not permitted), **ADDR#1** (# not permitted).
- F. **Note:** The variables in this lesson are strictly for printing, and consequently, they are all *strings*. (See glossary for *string*)
- G. Other ways of using variables are covered in subsequent lessons.

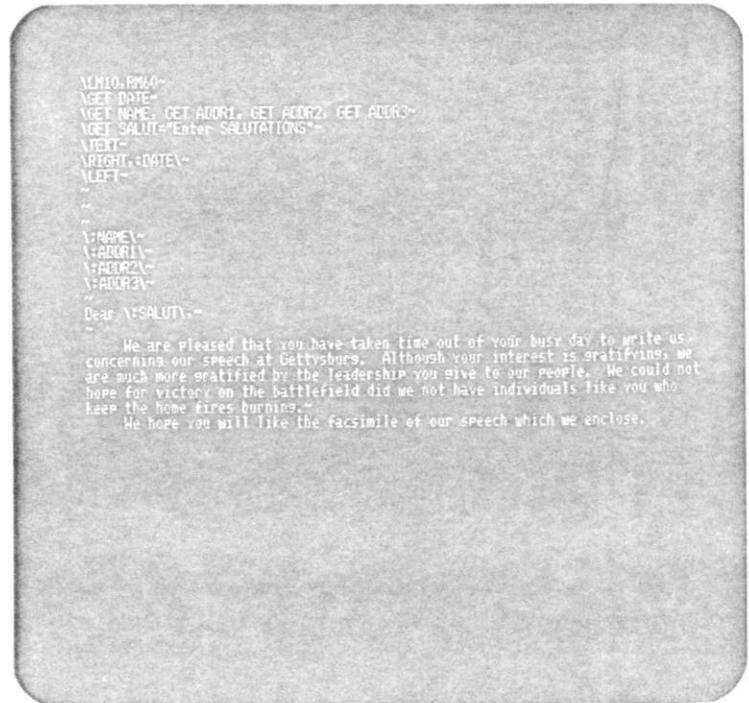
GETTING VARIABLES

- A. Embed the command, `GET VARIABLE`, in the text, e.g. `\GET NAME`.
- B. When the machine encounters this command, it prints `Enter VARIABLE` on the screen (e.g., `Enter NAME`) and waits for you to type in the value of the variable.
- C. When this happens, type up to 55 characters and press RETURN. (e.g. `Gen. George A. Custer`)
- D. **Note:** You may type any combination of characters — upper and lower case letters, numbers, punctuation marks, blanks, etc.
- E. The machine assigns the characters that you typed as the value of the variable.
- F. You may have the machine print something other than `Enter VARIABLE` on the screen.
- G. To do this, add to the GET command = "prompt", e.g., `GET NAME="Full Name (First name first)"`.
- H. The machine now prints the prompt on the screen instead of `Enter VARIABLE`, i.e., `Full Name (First name first)` instead of `Enter NAME`.
 1. Edit LESSON3B.
 2. Using the FULL INSERT Control Key, open up the text on the first line.
 3. Embed commands to set the left margin at ten characters and the right margin at 60 characters. (Type `\LM10,RM60` and press RETURN.)
 4. On the next line type `\GET DATE`.
 5. On the next line embed the commands to get a Name, Address Line one, Address Line two, and Address Line three. (Type `\GET NAME,GET ADDR1,GET ADDR2,GET ADDR3`)
 6. On the next line embed the command to get the salutation with the variable name as `SALUT` and a prompt string of `Enter SALUTATION`. (Type `\GET SALUT="Enter SALUTATION"`)



PRINTING A VARIABLE

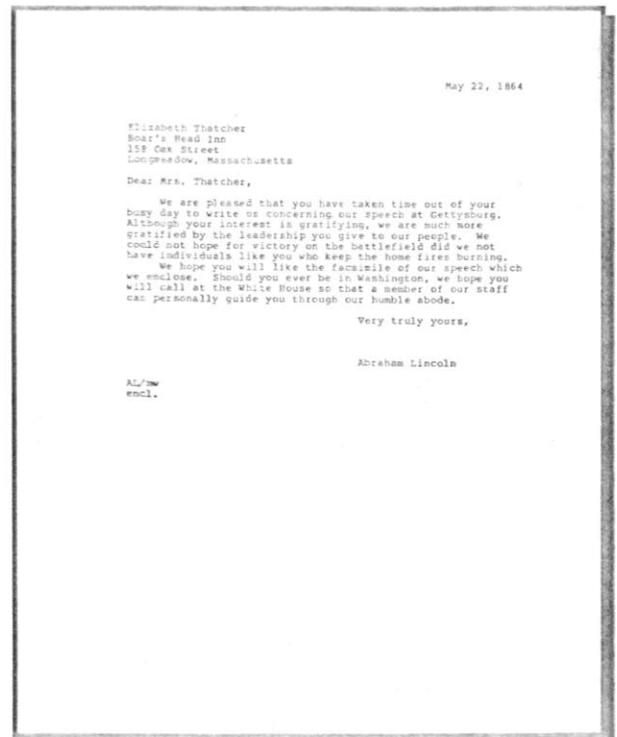
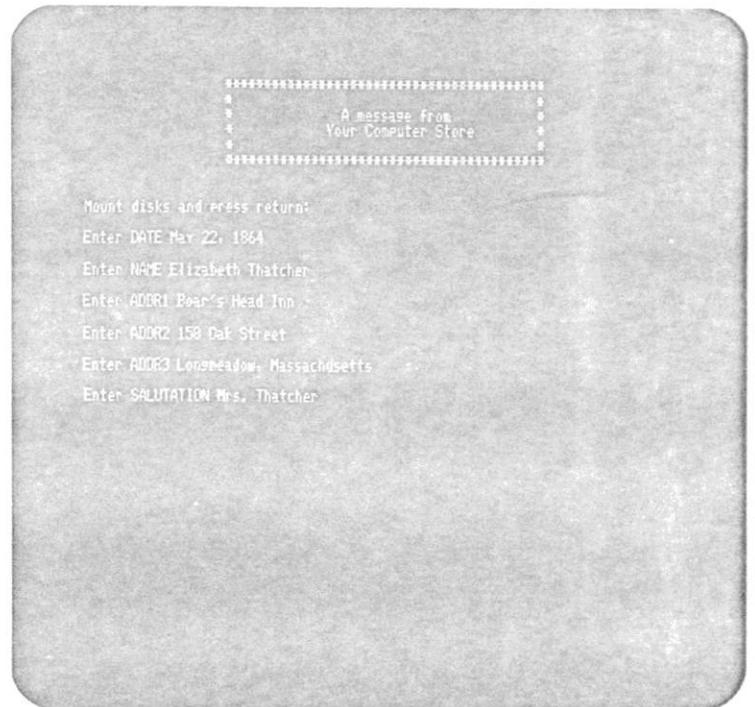
- A. To print the current value of a variable as part of the text, type `\:VARIABLE\`, e.g., `\:NAME\`.
- B. **Note:** The colon tells the machine that the expression is a variable and not a command.
- C. When the machine encounters the variable name, it includes the current value of that variable as part of the current text line.
- D. If the variable has no value at present, the machine prints nothing, not even blanks.
 1. On the next line in the text, embed the command to print the date flush right. (Type `\RIGHT\;:DATE\` followed by a carriage return.)
 2. **Note:** The second back slash is necessary if you want the date to be printed on a line by itself.
 3. Change to flush left. (Type `\LEFT`)
 4. Skip three lines (press RETURN three times) and embed the commands to print the name and address on the next four lines.
Type `\:NAME\
\:ADDR1\
\:ADDR2\
\:ADDR3\
\:`
 5. Skip a line (Press RETURN) and type the salutation. (Type `Dear \:SALUT\,`)



THE TEXT COMMAND

- A. As you learned in lesson Five, the machine processes commands until it encounters the first text character, i.e., the first character that is not part of a command.
- B. Although variables may become part of the printed text, the machine treats them like commands.
- C. Consequently, as the file now exists, the machine does not print the value of DATE when it encounters :DATE. (Note: This is because it does not come to the first text character (the carriage return following :DATE) until after it has already passed the command to print :DATE.)
- D. You could insert a carriage return prior to the command to print :DATE.
- E. Or you could type \TEXT preceding the command to print :DATE.
- F. When the machine encounters the TEXT command, it responds as if it had received a text character.
- G. Now when it encounters the command to print :DATE, it can do so.
 1. Embed the TEXT command so that the machine can print the current value of DATE at the beginning of the file. (Type \TEXT on the line preceding the :DATE command.)
 2. Exit the Text Screen and type END.
 3. Print LESSON3B, using the following information:


```
DATE    May 22, 1864
NAME    Elizabeth Thatcher
ADDR1   Boar's Head Inn
ADDR2   158 Oak Street
ADDR3   Longmeadow, Massachusetts
SALUTATION Mrs. Thatcher
(Type PRINT LESSON3B. When the machine starts requesting information, type in the above values.)
```



MULTIPLE PASSES OF SAME FILE

- A. At the beginning of the file, embed the commands to print an indefinite number of copies of the file. (Type `\COPY0`)
- B. As the machine complete each pass of the file, it now asks you if you wish to continue.
- C. Embed the command for a page feed at the end of the file. (Type `\NP [New Page]`)
- D. Each time the machine processes the file and encounters the GET commands, it asks you enter a value.
- E. When this happens, type in the new value for the variable and press RETURN.
- F. If the value of the variable has not changed from the previous pass, press RETURN without entering anything, e.g., the date would probably be the same on each pass through the file.
- G. If you need to blank out the previous value of a variable without replacing it, press the SPACE BAR followed by RETURN. (In the current file, this would most likely happen with ADDR3).
 1. Edit LESSON3B and save as LESSON6. (Type `EDIT LESSON3B LESSON6`)
 2. Embed the multiple copy and new page commands. (Type `\COPY0` at the beginning and `\NP` at the end)
 3. If you are not using continuous paper, embed the command for single form paper. (Type `\COPY0, FORM S`)
 4. Use the names and addresses from LESSON3 to create new letters (or make up your own).
 5. Continue printing new letters until you feel completely comfortable with this kind of letter.

May 23, 1864

John Doe
123 Elm Street
Anytown, USA

Dear John,

We are pleased that you have taken time out of your busy day to write us concerning our speech at Gettysburg. Although your interest is gratifying, we are much more gratified by the leadership you give to our people. We could not hope for victory on the battlefield did we not have individuals like you who keep the home fires burning. We hope you will like the facsimile of our speech which we enclose. Should you ever be in Washington, we hope you will call at the White House so that a member of our staff can personally guide you through our humble abode.

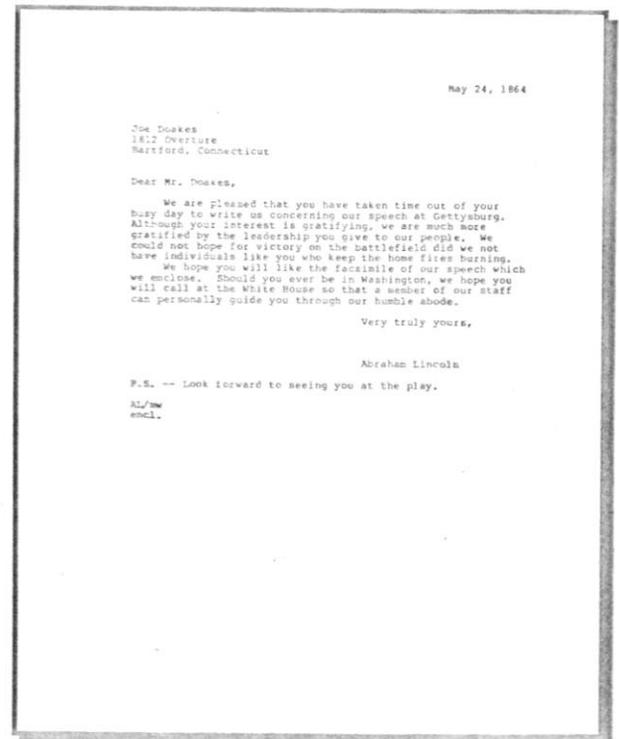
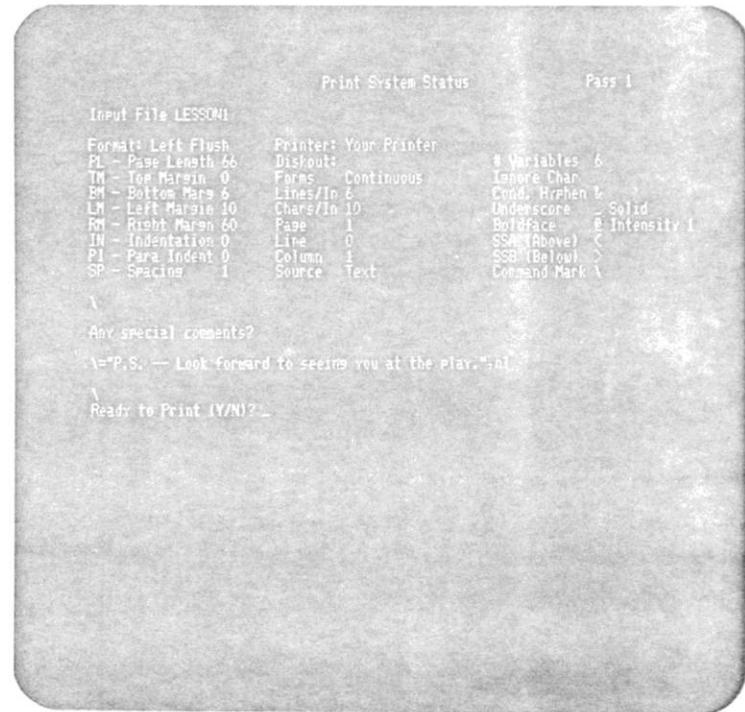
Very truly yours,

Abraham Lincoln

AL/mw
en2

THE WAIT/NEW LINE COMMANDS ENTERING TEXT FROM THE KEYBOARD

- A. Type `\WAIT`.
- B. The machine halts printing the file and goes to the screen for a command as if you had interrupted from the keyboard.
- C. You may print a message on the screen to remind yourself why you are waiting.
- D. To do so, type `\WAIT Message`, e.g., `\WAIT Any additional comments?`
- E. To enter text directly from the keyboard, type `=` "text" after the machine prints a back slash on the screen and press RETURN. (e.g. to enter "Hello", type `= "Hello` and press RETURN).
- F. The machine prints what you have just entered on the current line of the text as if it were part of the text file.
- G. With one command you may only enter the amount of text that fits on one line on the screen; however, you may repeat the `=` (equals quote) command as many times as you wish.
- H. The machine starts to print subsequent equals quote commands immediately following the previous text.
- I. If you need to force a new line from the keyboard, type `NL` (New Line).
- J. The machine responds as if it had encountered a carriage return in the text and moves the page forward one line.
- K. **Note:** You cannot enter a carriage return from the keyboard because pressing RETURN activates the command.
- L. To resume printing the file, press RETURN without entering a command.
 1. Edit LESSON6.
 2. Following the signature but before the identifiers, embed a WAIT command so that you may enter a post script directly from the keyboard. (Type `\WAIT Any special comments?`)
 3. End EDIT and print the file.
 4. When the machine asks you for any special comments, type a post script about how you are looking forward to seeing them soon, e.g., type `= "P.S. Look forward to seeing you at the play.`
 5. Force a new line so that the post script and the identifiers do not run together. (Type `NL`)
 6. Repeat the file using different postscripts until you feel comfortable entering text from the keyboard.



DOLLAR VARIABLES VARIABLES IN A TEXT LINE

- A. Put a dollar sign (\$) instead of a colon in front of a variable name.
- B. When the machine encounters this command, it formats the current value of the variable with commas and possibly a decimal point (as if it were dollars and cents).
- C. The machine includes only numbers in the dollar format.
- D. The machine ignores all other characters except the period, the minus sign and the plus sign. (**Note:** this is so that the machine can accept values that already contain commas, dollar signs, etc.)
- E. The first period in the value becomes the decimal point.
- F. **Note:** If the value does not include a period, the machine formats it as a whole number.
- G. The last sign the machine encounters, either minus or plus, determines whether the dollar format is positive or negative.
- H. **Note:** *You may put any number in dollar format; it does not have to be actually dollars.*
 1. Load LESSON6 into the Printer (Type **PRINT LESSON6**). You will not actually print the text.
 2. Clear the screen. (Type **CLS**)
 3. When the machine prints a back slash on the screen, use the SET and SHOW commands to set the value of the variable DOLLAR and display it on the screen in dollar format. (**Note:** these are only examples of how the dollar format works; you would never actually print most of these values in dollar format.)
 - a. DOLLAR as 1234 (Type **SET DOLLAR="1234",SHOW \$DOLLAR**)
[Result: 1,234]
 - b. DOLLAR as 1234.56 (SET **DOLLAR="1234.56",SHOW \$DOLLAR**)
[Result: 1,234.56]
 - c. DOLLAR as -1234. [Result:
1,234.00-]
 - d. DOLLAR as 1600 Pennsylvania Ave.
[Result: 1,600.00]
 - e. DOLLAR as REC. 1234. [Result: .12]
 - f. DOLLAR as Gen. George A. Custer.
[Result: .00]
 4. Type **QUIT** to exit from PRINT without processing the file.

```

Aset dollar="1234",show $dollar
1,234

Aset dollar="1234.56",show $dollar
1,234.56

Aset dollar="-1234.",show $dollar
1,234.00-

Aset dollar="1600 Pennsylvania Ave.",show $dollar
1,600.00

Aset dollar="REC. 1234",show $dollar
.12

Aset dollar="Gen. George A. Custer",show $dollar
.00

Aquit
AD_

```

USING THE DOLLAR FORMAT

- A. You may print the current value of a variable in dollar format.
- B. To do so, embed the variable in the text with the dollar sign taking the place of the colon, i.e., `\$VARIABLE\`.
- C. When the machine encounters this command, it converts the current value of the variable to dollar format and inserts it into the current text line.
- D. If you want a dollar sign to appear in front of a variable in dollar format, you must include the dollar sign as part of the text file.
 1. Edit SAMPLE3. (A form letter thanking people who have contributed to Lincoln's re-election campaign.)
 2. Print SAMPLE3 from EDIT using a line length of 65 characters. (Type **L65**, then type **P**)
 3. Notice how the variables are set up, with **CITY** and **STATE** as separate variables.
 4. Exit EDIT using **QUIT**.
 5. Print SAMPLE3 (**PRINT SAMPLE3**) using the following information:

Robert M. Curry
33 Beacon Street
Boston
Massachusetts
1000
 6. **Winona Bates Crandsdell**
1956 Chicago Ave.
Rock Island
Illinois
10.00

```

\LM10, RM60, COPY0
\GET DATE="Today's Date?"
\GET NAME="Full name? (first name first)"
\GET ADDR1="First line of address?"
\GET ADDR2="Second line of address, if any?"
\GET CITY="City?"
\GET STATE="State? (No Abbreviations!)"
\GET SALUT="Salutation?"
\GET CONTRIB="Amount of Contribution? ('0' if none)"
\RIGHT, :DATE\
\LEFT\

```

```

\;NAME\
\;ADDR1\
\;ADDR2\
\;CITY\, \;STATE\

```

```

\SP2
Dear \;SALUT\,

```

```

Thank you for your recent contribution of \$\CONTRIB\
to the Committee to Re-elect President Lincoln. We are pleased
that you think enough of the President to participate in his
campaign in this way.
We have always had strong supporters in \;STATE\, but with
people like you on our side, we expect to do even better this
time around.
Thank you again, \;SALUT\. If we may be of service to you,
we hope you will not hesitate to ask.
\IND\
Sincerely,

```

```

\SP1
Joseph B. Miller,
Finance Committee
\IND\
JBM/mw
\SP

```

June 1, 1964

```

Robert M. Curry
33 Beacon Street
Boston, Massachusetts

```

Dear Mr. Curry,

```

Thank you for your recent contribution of $1,000 to the
Committee to Re-elect President Lincoln. We are pleased
that you think enough of the President to participate in his
campaign in this way.

```

```

We have always had strong supporters in Massachusetts,
but with people like you on our side, we expect to do even
better this time around.

```

```

Thank you again, Mr. Curry. If we may be of service to
you, we hope you will not hesitate to ask.

```

Sincerely,

```

Joseph B. Miller,
Finance Committee

```

JBM/mw

LESSON SEVEN

Being able to fill in the value of variables from the keyboard at run time is a real time-saver compared to creating each letter individually in the Editor. Even so, you must still sit at the keyboard throughout the entire session. In this lesson you learn how to fill in the value of variables from an external data file. Of course, you must first put the information into the data file, but that does not take very long as you also learn in this lesson.

DEFINING THE DATA FILE

- A. Type `\FILE` so that the machine knows you are defining a file.
- B. On the same line type `T12`.
- C. The machine now knows that your file is a *text* file with twelve elements per record. (See the glossary for *text file*, *element* and *record*.)
- D. On the same line type `,SAMPLE4` and press RETURN.
- E. The machine now knows that the file is named `SAMPLE4`.
- F. The entire command line looks like this:
`\FILE T12,SAMPLE4`
- G. **Note:** The `FILE` command must appear on a line by itself.
 1. Edit `SAMPLE3` and save as `LESSON7`. (Type `EDIT SAMPLE3 LESSON7`)
 2. Except for `GET DATE`, delete the `GET` commands. (Use the `LINE DELETE` Control Key.)
 3. In their place embed a command defining a text data file named `SAMPLE4` with twelve elements per record. (Type `\FILE T12,SAMPLE4` and press RETURN.)

```

ALPH0-RM60-DUP10~
GET DATE="Today's Date?"
^FILE T12,SAMPLE4~
~
^RIGHT,+DATE~
^LEFT~
~
^NAME~
^ADDR1~
^ADDR2~
^CITY, ^STATE~
~
^SP2~
Dear ^SALUT~

```

```

Thank you for your recent contribution of $14000R18A to the Committee to
Re-elect President Lincoln. We are pleased that you think enough of the
President to participate in his campaign in this way.
We have always had strong supporters in ^STATE1, but with people like you
on our side, we expect to do even better this time around.

```

DEFINING THE DATA STRUCTURE

- A. Type `\DATA VARIABLE1,VARIABLE2`, etc.
- B. When the machine encounters the **T12** in the **FILE** command, it knows to read the first twelve lines of the file. (Remember, a *line* is any number of characters ending in a carriage return.)
- C. The **DATA** command tells the machine what to do with those lines once it has read them.
- D. The variables in the **DATA** command are like other variables, i.e., they may have a value of up to 55 characters in any combination.
- E. If a line in the data file is longer than 55 characters, the value of the corresponding variable is limited to the first 55 characters of the line.
- F. The data file may contain elements that you do not use in the current text file.
 1. You may skip over unused elements by typing a comma in the **DATA** statement without typing a variable name.
 2. If your **DATA** statement assigns less than the specified number of elements, the remaining elements are skipped over.
 1. Display on the screen **SAMPLE4**. (ESCAPE to the Command Screen and type **DSAMPLE4**.)
 2. Notice that the twelve lines of Record One consists of the **RECORD NUMBER**, **NAME**, **ADDR1**, **ADDR2**, **CITY**, **STATE**, **SALUT**, **CONTRIB**, **TIMES**, **TEXT CODE**, **P.S.** (Y or N), and a blank line for spacing.
 3. Write a **DATA** statement for this file, using only the variables that appear elsewhere in the text. Skip over unneeded elements. (Type `\DATA ,NAME,ADDR1,ADDR2,CITY,STATE,SALUT,CONTRIB,,,,`) (Note: The last four commas are optional.)

```

RECORD #1
Gen. George A. Custer
Seventh Cavalry
Army of the Potomac
Near Richmond
Virginia
George
100
1
#1
N

```

```

RECORD #2
J. Wilkes Booth, Esq.
Real Playhouse

```

```

Baltimore
Harrland
Dr. Booth
0

```

```

23

```

```

Display of File SAMPLE4 **: Press Return to continue, Escape to stop

```

```

ALM10-RM40.COPY0-
GET DATE="Today's Date?"
FILE T12,SAMPLE4-
\DATA ,NAME,ADDR1,ADDR2,CITY,STATE,SALUT,CONTRIB-

```

```

\RIGHT, :BATEV-
\LEFTV-

```

```

\NAMEV-
\ADDR1V-
\ADDR2V-
\CITYV, \STATEV-

```

```

\SP2-
Dear \SALUTV-

```

```

Thank you for your recent contribution of $%CONTRIB% to the Committee to
Re-elect President Lincoln. We are pleased that you think enough of the
President to participate in his campaign in this way.
We have always had strong supporters in \STATEV, but with people like you

```

PRINTING USING A DATA FILE

- A. The machine always starts with the first record of a file unless you tell it otherwise.
- B. After opening the data file, reading the first record and assigning values based on the DATA statement, the machine prints the file one time. (This is *pass one*.)
- C. After completing the first pass, the machine reads the next record from the data file and prints the file again using the new information in that record.
- D. The machine prints one pass per record until there are no more complete records to read from the data file.
- E. Each time the machine encounters the GET DATE command, it prints **Enter DATE** on the screen and waits for you to type in a value.
- F. Since the date does not change from pass to pass, simply press RETURN.
- G. **Note:** You learn later in this lesson how to eliminate this duplication of effort.
 1. Delete the COPY0 command (It is no longer necessary, since the machine cycles the file automatically.)
 2. End EDIT.
 3. Print the file you just edited. (Type **PRINT LESSON7.**)

June 4, 1864

Hon. Patrick MacFarland
Mayor, City of Worcester
City Hall
Worcester, Massachusetts

Dear Mayor MacFarland,

Thank you for your recent contribution of \$1,200 to the Committee to Re-elect President Lincoln. We are pleased that you think enough of the President to participate in his campaign in this way.

We have always had strong supporters in Massachusetts, but with people like you on our side, we expect to do even better this time around.

Thank you again, Mayor MacFarland. If we may be of service to you, we hope you will not hesitate to ask.

Sincerely,

Joseph H. Miller,
Finance Committee

JHM/mw

June 4, 1864

Gen. George A. Custer
Seventh Cavalry
Army of the Potomac
Near Richmond, Virginia

Dear George,

Thank you for your recent contribution of \$100 to the Committee to Re-elect President Lincoln. We are pleased that you think enough of the President to participate in his campaign in this way.

We have always had strong supporters in Virginia, but with people like you on our side, we expect to do even better this time around.

Thank you again, George. If we may be of service to you, we hope you will not hesitate to ask.

Sincerely,

Joseph H. Miller,
Finance Committee

JHM/mw

CREATING A DATA FILE

- A. Load your data file into the Editor.
- B. Type in the new records using exactly the same number of lines as the other data records.
- C. You may insert the new records anywhere in the file that you choose, e.g., in alphabetical order, or, as in SAMPLE4, by record number.
- D. You may also change existing records in the same way, e.g., if a person on your mailing list has a new address.
- E. When you have made all the additions and changes that need to be made, end EDIT.
 - 1. Load SAMPLE4 into the Editor and save as LESSON7D. (Type EDIT SAMPLE4 LESSON7D.)
 - 2. At the end of the file [Press the BOTTOM Control Key], add the following two records:

Record 6	Record 7
Clara Barton	Jefferson Davis
Mt. Washington	Confederate Office
Army Hospital	Building
	150 Magnolia Drive
Trenton	Mobile
New Jersey	Alabama
Nurse Barton	President Davis
12.50	150.
1	3
A2	Z1
N	Y

- 3. Remember to leave an extra line at the end of each record, including the last one.
- 4. End EDIT.
- 5. **Note:** You do not need all of this information immediately, but you will use it in Lesson Eight.



THE SETUP COMMAND

- A. On the first line, in the first position, type `\SETUP`.
- B. The machine only executes the commands between the `SETUP` and `TEXT` commands on the first pass through the file.
- C. You may only include commands in this initial section; the machine ignores any text characters that are generated.
- D. Include those commands that need to be executed only once, e.g., the `FILE` and `DATA` commands, the `GET DATE` command, etc.
- E. Include only those formatting commands that remain unchanged throughout the file.
- F. **Note:** Like the `TEXT` command, the `SETUP` command must be the only command in the command line.
- G. When there is a `SETUP` command in the file, the machine pauses when it encounters the `TEXT` command on the first pass.
- H. After it pauses, it prints the Status Screen (as if you had typed `DS`), and allows you to enter commands from the keyboard.
- I. On subsequent passes, the machine does not pause at the `TEXT` command.
- J. **Note:** You must not have a `SETUP` command without a `TEXT` command, because the machine does not print any text until it encounters a `TEXT` command.
 1. Edit `LESSON7`.
 2. Insert the `SETUP` statement on the first line. (Type `\SETUP`)
 3. Rearrange your commands so that those that are executed only once are between the `SETUP` and `TEXT` commands. (Use the `BLOCK MOVE` Command to move the formatting commands to follow the `TEXT` command)

```

\SETUP~
ALR10, RM60~
VGET DATE="Today's Date?"~
VFILE T12, SAMPLE4~
VDATA : NAME, ADDR1, ADDR2, CITY, STATE, SALUT, CONTRIB~
VTEXT~
VRIGHT, -DATE~
VLEFT~
~
~
V:NAME~
V:ADDR1~
V:ADDR2~
V:CITY, V:STATE~
~
VSP2~
Dear V:SALUT,~
~
      Thank you for your recent contribution of $V:CONTRIB to the Committee to
Re-elect President Lincoln. We are pleased that you think enough of the
President to participate in his campaign in this way.~

```

THE FNAME COMMAND

- A. If your text file includes an initial section, the machine does not open the data file until after it pauses at the TEXT command.
- B. Before the machine opens the file, you may change the name of the data file with the FNAME command.
- C. To do so, type **FNAME FILENAME**.
- D. You may use a variable to set the FILENAME by typing **FNAME :VARIABLE**.
- E. **Note:** The FILENAME in the FILE command may also be a variable.
- F. If you define the data file with a variable, you must first assign a value to the variable with a GET or SET command.
- G. If the machine cannot find the FILENAME that you originally specified, it prints an error message on the screen.
- H. When this happens, you may use the FNAME command to define the file.
- I. After the machine has opened the data file, it ignores any other FILE or FNAME command.

SYSTEM VARIABLES

- A. A system variable is one that the machine itself defines.
- B. A system variable is indicated by a percent sign (%) in front of the variable name.
- C. The system variables are:
 - %PAGE** The current page number
 - %PASS** The current pass number
 - %REC** The current record number
 - %LINE** The current line number
 - %LINES** The current number of lines from the bottom of the page
 - %COL** The current column number
 - %EOF** End of file for the data file
- D. The system variables help you keep track of where you are in the text.
- E. **Note:** You learn more about system variables in the last two lessons and in the notes.
 1. Following the TEXT command, embed a command to display on the screen the current pass number, record number and name of the person whose letter is being printed. (Type **\SHOW "Pass ", %PASS," Record ",%REC," Name: ", :NAME)**
 2. End EDIT.

```

\SETUP~
\LEN10,RM60~
\GET DATE="Today's Date?"
\FILE TLZ,SAMPLE1~
\DATA :NAME,ADDR1,ADDR2,CITY,STATE,SALUT,CONTRIB~
\TEXT~
\SHOW "Pass: ",%PASS~ Record ",%REC," Name: ",:NAME~
\RIGHT:,DATE~
\LEFT~
~
~
~
\NAME~
\ADDR1~
\ADDR2~
\CITY, :STATE~
~
\SP2~
Dear \SALUT,~

```

Thank you for your recent contribution of \$%CONTRIB to the Committee to Re-elect President Lincoln. We are pleased that you think enough of the

THE START AND STOP COMMANDS

- A. Type **START6**.
- B. The machine skips the first five records and starts printing the file with the sixth record.
- C. Type **STOP6**.
- D. The machine stops after having printed the sixth record.
- E. **Note:** The **START** and **STOP** commands have a different meaning if your text file does not access an external data file. (See Lesson Nine)
- F. Type **END**.
- G. The machine ends the current pass of the file and loads the next record from the data file.
- H. Type **QUIT**.
- I. The machine immediately terminates the current file and returns to CP/M.
 1. Begin to process **LESSON7**. (Type **PRINT LESSON7**)
 2. Change the name of the data file to **LESSON7D**. (When the machine pauses before opening the data file, type **FNAME LESSON7D**)
 3. Print only the records that you just entered. (Type **START6**)

```

Input File: LESSON7
Data File: SAMPLE4      Text Elements per Record: 12

Format: Left Flush      Printer: Your Printer      # Variables: 8
PL - Page Length: 66    Diskout:                   Ignore Char
TM - Top Margin: 0      Forms: Continuous        Cond. Hiden: 1
EM - Bottom Marg: 5    Lines/In: 6              Underscore: Solid
LM - Left Margin: 5     Chars/In: 10             Boldface: B Intensity: 1
RM - Right Margin: 60   Page: 1                   SSN (Above): <
IN - Indentation: 0     Line: 0                   SSN (Below): >
PI - Para Indent: 0    Column: 1                 Command Mark: \
SP - Spacing: 1        Source: Text

\FNAME LESSON7D
\START 6
\
Pass 1 Record 6 Name: Clara Barton
Pass 2 Record 7 Name: Jefferson Davis
End of Job
02_

```

REVIEW

Using everything you have learned in this lesson, convert the **LESSON6** file so that it accesses **LESSON7D** as a data file.

1. Remove all of the **GET** commands except for **GET DATE**.
2. Type **\FILE T12, LESSON7D**.
3. Define the data structure with the appropriate variables. (Type **DATA , NAME, ADDR1, ADDR2, CITY, STATE, SALUT**)
4. Remove the **COPY0** command.
5. Insert the **SETUP** and **TEXT** commands.
6. Insert the command to display on the screen the pass number, record number and name. (Type **SHOW "PASS ", %PASS," Record ",%REC," Name: ", :NAME**)
7. End **EDIT**.
8. Print **LESSON6**.

```

\SETUP-
\ Document revised for use with name-and-address file-
\M10.RM60-
\GET DATE-
\FILE T12, LESSON7D-
\DATA , NAME, ADDR1, ADDR2, CITY, STATE, SALUT-
\TEXT-
\SHOW "Pass ", %PASS, " Record ", %REC, " Name: ", :NAME-
\RIGHT: DATE-
\LEFT-
-
-
\ :NAME-
\ :ADDR1-
\ :ADDR2-
\ :CITY, \ :STATE-
-
Dear \ :SALUT, -

We are Pleased that you have taken time out of your busy day to write us
concerning our speech at Gettysburg. Although your interest is gratifying, we
are much more gratified by the leadership you give to our people. We could not

```

LESSON EIGHT

Lesson Eight deals with conditional commands, i.e., commands that are only executed if a condition that you establish is true. Don't be concerned with any jargon that we might use. The concepts are quite simple. Here are some examples of conditional statements from everyday life.

1. If the weather bureau is forecasting rain, bring your umbrella to work with you.
2. If your alarm clock does not go off, you oversleep.
3. If you eat more than you should, you gain weight.

Conditional commands give you a great deal of control over the text that you print, but sometimes they can be tricky. Consequently, before you give a conditional command, you should analyze exactly what you want to accomplish and how best to bring that about. We are a bit more wordy in this lesson than in the others because we want to go through the analyses with you.

TESTING FOR A BLANK VARIABLE

When you printed LESSON6 and LESSON7, you had two problems: 1) The machine asked you to enter the date on each pass of the file. 2) If you only had one address line, the machine left a blank line before the city and state. The first problem was corrected by putting GET DATE into the initial section, but we might have used a conditional statement instead.

- A. Problem: You want to execute the GET DATE command only if you have not previously entered a date, i.e., if the value of DATE is blank.
- B. Solution: Type **IF DATE=" "**, in front of GET DATE.
- C. The entire command line now reads **\IF DATE=" ",GET DATE.**
- D. When the machine encounters this command, it checks to see if it is true, i.e., if the current value of DATE is blank.
- E. If DATE is blank, the machine executes the GET command, i.e., it prints **Enter DATE** on the screen and waits for you to type in the date.
- F. If you have previously defined DATE, then DATE is *not* blank, and the machine skips the rest of the command line.
- G. Problem: You want to print the value of ADDR2 only if it is *not* blank.
- H. Solution: In front of :ADDR2, insert **IF NOT ADDR2=" "**,
- I. Behind :ADDR2, insert a new line (NL) command and delete the second back slash.
- J. The command line now reads **\IF NOT ADDR2=" ";:ADDR2,NL**
- K. When the machine encounters this command, it checks to see if ADDR2 is not blank.
- L. If ADDR2 is not blank, the machine prints ADDR2; otherwise, it ignores the remainder of the command line and processes the next line of the file.
- M. Note: You use the NL command whenever you cannot press RETURN, e.g., since you cannot press RETURN in the middle of a command line, the NL command is the only way to give a *conditional* carriage return.
 1. Edit LESSON6.
 2. Make the changes discussed above.
 3. End EDIT and print records 4 through 6 of LESSON6. (Type START4,STOP6)

```

\SETUP-
\LM10,RM40-
\* GET DATE removed from setup section for illustration-
\FILE T12,LESSON7-
\DATA :NAME,ADDR1,ADDR2,CITY,STATE,SALUT-
\TEXT-
\IF DATE=" " GET DATE-
\SHOW "Pass " *PASS," Record ".ZREC," Name: ":NAME-
\RIGHT :DATE\
\LEFT-
-
-
\NAME\
\ADDR1\
\IF NOT ADDR2=" ";:ADDR2,NL-
\CITY\ :STATE\
-
Dear :SALUT\
-
  We are pleased that you have taken time out of your busy day to write us
  concerning our speech at Gettysburg. Although your interest is gratifying, we
  are much more gratified by the leadership you give to our people. We could not

```

June 5, 1864

Lisa Jeffreys
Wilson School
Pa. Bend, Indiana

Dear Lisa,

We are pleased that you have taken time out of your busy day to write us concerning our speech at Gettysburg. Although your interest is gratifying, we are much more gratified by the leadership you give to our people. We could not hope for victory on the battlefield did we not have individuals like you who keep the home fires burning.

We hope you will like the facsimile of our speech which we enclose. Should you ever be in Washington, we hope you will call at the White House so that a member of our staff can personally guide you through our humble abode.

Very truly yours,

Abraham Lincoln

AL/mw
encl.

THE LENGTH OF A VARIABLE

- Another way to see if a variable is blank is to check the length of its current value, i.e., if its current length is zero, then it is blank.
- B. To get the current length of a variable, type **&VARIABLE**, e.g., **&DATE**.
 - C. **Note:** The length of a variable is the number of characters from the first character to the last non-blank character.
 1. Edit LESSON6.
 2. Change the conditional command so that the machine executes GET DATE only if the *length* of DATE is zero. (Type **\IF &DATE=0,GET DATE**)
 3. Change the conditional command so that the machine only prints a second address line when the length of ADDR2 is not zero. (Type **\IF NOT &ADDR2=0, :ADDR2,NL**)
 4. Have the machine display the current length of NAME (Type **\SHOW :NAME," is ",&NAME," characters long."**)
 5. End EDIT and print records 4 through 6. (Type **START4,STOP6**)

```

\SETUP~
\LN10,RM60~
\* GET DATE removed from setup section for illustration~
\FILE T12,LESSON70~
\DATA ,NAME,ADDR1,ADDR2,CITY,STATE,SALUT~
\TEXT~
\IF &DATE=0,GET DATE~
\SHOW "Pass ",&PASS," Record ",&REC," Name: ",&NAME~
\SHOW "NAME" is ",&NAME," characters long."~
\RIGHT:DATE~
\LEFT~
~
~
~
~
\NAME~
\ADDR1~
\IF NOT &ADDR2=0,ADDR2,NL~
\CITY, \STATE~
Dear \SALUT~
~
  We are pleased that you have taken time out of your busy day to write us
  concerning our speech at Gettysburg. Although your interest is gratifying, we

```

THE END COMMAND

- A. Type **\IF NOT CODE="A2",END**.
- B. When the machine encounters this command, it checks to see if the current value of CODE does not equal **A2**.
- C. If it does not equal **A2**, then the condition is true, and the machine executes the **END** command and reads the next record from the data file.
- D. If **CODE** equals **A2**, the condition is not true, and the machine does not end the pass.
- E. Using the **END** command in conjunction with a conditional statement makes it possible for you to print only selected records from a data file.
- F. **Note:** Be sure to place this conditional command following the **TEXT** command but before the first printable character.
 1. Edit LESSON6.
 2. Add **CODE** to the list of variables in the data statement. (Type **DATA ,NAME,ADDR1,ADDR2,CITY,STATE,SALUT,,CODE**)
 3. Embed a conditional command so that the machine does not print the file if **CODE** does not equal **A2**. (On the line following the **TEXT** command, insert **\IF NOT CODE="A2",END**)
 4. End LESSON6 and print LESSON6.

```

\SETUP~
\LN10,RM60~
\* GET DATE removed from setup section for illustration~
\FILE T12,LESSON70~
\DATA ,NAME,ADDR1,ADDR2,CITY,STATE,SALUT,,CODE~
\TEXT~
\IF NOT CODE="A2",END~
\IF &DATE=0,GET DATE~
\SHOW "Pass ",&PASS," Record ",&REC," Name: ",&NAME~
\SHOW "NAME" is ",&NAME," characters long."~
\RIGHT:DATE~
\LEFT~
~
~
~
~
\NAME~
\ADDR1~
\IF NOT &ADDR2=0,ADDR2,NL~
\CITY, \STATE~
Dear \SALUT~
~
  We are pleased that you have taken time out of your busy day to write u

```

NUMERIC VARIABLES

In lesson Six you learned how to use *string* variables, i.e., variables consisting of any series of characters. In this section you are introduced to *numeric* variables, i.e., variables that can be treated as numbers, and not just as characters.

- A. The machine treats a variable as a string (a series of characters) unless you tell the machine to treat it as a number.
- B. Do this by putting a number sign (#) in front of the variable name, i.e., #VAR.
- C. Only the numbers at the beginning of a variable may be treated numerically; the machine stops converting the string as soon as the first non-number is encountered, e.g., "201" becomes 201, "1234 Elm St." becomes 1234, "New York, NY 10010" becomes 0.
- D. Fractions are rounded down to the nearest whole number, e.g., 10.5" becomes 10, ".75" becomes 0.
- E. The machine does not recognize negative numbers. (Because the minus sign is not a number.)
- F. A number greater than 32767 results in an error message.
- G. You may add or subtract a number variable and/or numeric constant as long as the result is not less than zero or greater than 32767. (This is called a numeric expression)
- H. If the result of a numeric expression is less than zero or greater than 32767, the machine prints an error message
- I. All length variables and system variables are numeric, and you may use them in any way that you can use a number variable.
- J. You may use a numeric variable or expression in almost any command that requires a number, e.g., LM#VAR, TAB20-&VAR, etc.
 1. PRINT SAMPLE5 and follow the instructions it gives you.
 2. Section one — As the machine asks, enter the following values and notice what happens when the machine treats them numerically: 1234, 1234.56, 123 Oak Street, -123, SEVEN, 123456.
 3. Section two — As the machine asks, enter the following number pairs, 2 and 2, 100 and 1, 5 and 7.
 4. Section three — As the machine asks, enter the following number pairs and notice how the status screen reflects the changes: LM as 10 and RM as 60, LM as 20 and RM as 50, LM as 0 and RM as 45.

DOLLAR COMPARISONS

- A. Since numeric variables must be whole numbers between zero and 32767, they may not be used to compare very large, negative or fractional numbers.
- B. The dollar comparison overcomes these restrictions.
- C. To establish a dollar comparison, prefix the first variable name in the IF statement with a dollar sign.
- D. You may use the dollar comparison to see if variable is greater than ($>$), less than ($<$) or equal to ($=$) another variable or constant.
- E. You may test for any two of these relationships in one comparison, i.e., greater than or equal to ($>=$), less than or equal to ($<=$), and greater than or less than [not equal to] ($<>$).
- F. If you are comparing the dollar variable to a constant, put the constant between quotation marks.
- G. If you are comparing the dollar variable to another variable, prefix the second variable with a dollar sign as well.
- H. **Note:** Your number may have as many as twelve figures to the left of the decimal point and two to the right. You may compare negative numbers.
- I. You may not use the dollar comparison to compare a variable to a system or length variable or to a numeric expression.
- J. If you need to make this kind of comparison, you must use a number comparison, e.g., **IF #VAR, etc.**
 - 1. PRINT SAMPLE5 and follow the instructions it gives you.
 - 2. Section one: When the machine asks you, type in the following values and see the difference between a colon variable, a number variable and a dollar variable:
1234, 1234.56, \$1,234.56, 1000000., -12.45
 - 3. Section two: When the machine asks you, type in the same numbers as above and see the difference between a string comparison (without prefix), a number comparison and a dollar comparison.

THE SKIP COMMAND

- A. The SKIP command allows you to skip over portions of the text file.
- B. If you type **SKIP**, by itself, the machine skips over the next line of the text file, i.e., it skips until it reaches the next carriage return.
- C. If you follow the SKIP command with a carriage return instead of a back slash, the machine skips over that carriage return as well.
- D. If you type **SKIP2**, the machine skips over the next two lines from the file, i.e., two carriage returns.
- E. You may skip as many lines as you wish.
- F. **Note:** The machine counts the number of carriage returns; it does not care how many or what kind of characters precede them. This means that you can skip over command lines, text lines or blank lines.

THE SKIP TO COMMAND THE IGNORE CHARACTER

1. If you type **SKIP TO***, the machine passes over the file until it encounters the next asterisk.
2. You may skip to any character you choose except the command marker.
3. The machine prints the character that it skips to unless it is a recognition character, e.g., a Ghost Hyphen, underscore marker, etc.
4. If you do not wish to print the character, you may establish it as a special recognition character that is ignored when it is part of the text, i.e., the machine *recognizes* it as a character but does not *print* it.
5. To do this, type **IGNOREc**, e.g., **IGNORE***
6. You may define any character as the **IGNORE** character except the command marker.
7. **Note:** If you skip to a character in the middle of a command line, the machine prints the commands instead of executing them because it skipped over the command marker.
 1. Edit **SAMPLE6**.
 2. Print a copy with a line length of 65 characters. (Type **L65**; then type **P**.)
 3. Exit **EDIT** by typing **QUIT**.
 4. Notice in the **SAMPLE6** file, we have defined a carat sign (^) as the **IGNORE** character.
 5. Notice that we have used both dollar and number comparisons.
 6. **PRINT SAMPLE6**.
 7. Notice how the **SKIP** command allows you to make a form letter more personal.
 8. **Note:** The **LINE** command allows you to go directly to a particular line. (See the Notes for a full discussion)

```

\SETUP
\LM10,RM60
\GET DATE="Today's date?"
\FILE T12,LESSONTD
\DATA ,NAME,ADDR1,ADDR2,CITY,STATE,SALUT,CONTRIB,TIMES,,COMMENT
\
\TEXT
\IF #CONTRIB<=0,END
\IF #PASS=1,CLS
\SHOW RECORD *,#PASS,* *,#NAME,TAB 40,"CONTRIBUTION:"
*,#CONTRIB
\
\RIGHT,DATE\
\LEFT,LINES
\
\#NAME\
\#ADDR1\
\IF ADDR2<>**.,#ADDR2,NL
\#CITY\, \#STATE\
\LINE14
Dear \#SALUT\,

  Thank you \IF #TIMES>2,*"once *\IF #TIMES>1,*"again "\for
your \IF #CONTRIB>=100,*"generous "\contribution of
#\CONTRIB\ to the Committee to Re-elect President Lincoln.
\IF #CONTRIB>=25*,SKIP
  We know that \#\CONTRIB\ may not seem like very much, but
it is the small contributors like you, \#SALUT\, that make it
possible to run a campaign. President Lincoln has often said
that he would rather get ten small contributions than one large
one.
\IF #CONTRIB<=25*,SKIP
\IF #CONTRIB>=100*,SKIP
  We know that it took a considerable commitment on you part
to contribute as much as you did. President Lincoln has often
said that commitment should be matched with equal commitment.
We want you to know, \#SALUT\, that the committee is committed
to seeing that your \#\CONTRIB\ is spent wisely.
\IF #CONTRIB<=100*,SKIP
  It's people like you, \#SALUT\, that make it possible to
run the kind of campaign we need to insure success. So we hope
that you will consider this to be your campaign as well as ours.
We welcome any comments that you might have concerning
positions that President Lincoln has taken during the
campaign.\IF #CONTRIB<=1000*,NL,SKIP\ And if you are in
Washington, we hope you will call on us so that we can arrange a
personal meeting with President Lincoln.
  Again, we want to express our appreciation for your \IF
#TIMES=2,*"repeated *\IF #TIMES>2,*"continued "\generosity. If
we may be of service to you \IF #CONTRIB>=999,*"in any way*\IF
#CONTRIB>=999,*" (and we do mean any!)", we hope you won't
hesitate to write us.

```

```

\INFO\
Sincerely,

Joseph H. Miller
Finance Committee
\INFO\
JHM/sv
^

```

PUTTING COUNTERS IN THE FILE

- A. A counter is a variable that you use to keep track of how many times you have done something.
- B. A counter is created using a SET command to add (or subtract) numbers to the current value of a variable.
- C. The variable in the SET command must be a number variable, e.g.,
SET #VAR=#VAR+1
- D. When you have finished the file, you can have the machine print a report showing you what was processed.
- E. To do this, you need to use the %EOF system variable.
- F. When the machine has reached the end of the data file, %EOF is 1; at all other times it is zero.
- G. To get a report, put a COPY0 command in the SETUP section.
- H. Now, put a series of commands following the TEXT command to be executed only when %EOF=1.
- I. When the machine has completed the data file, it prints **READY TO PRINT?** on the screen.
- J. If you want a report press **Y**; otherwise, press **N** to quit processing.
 1. Edit SAMPLE6 and save as LESSON8. (EDIT SAMPLE6 LESSON8)
 2. Embed counters following the TEXT command to keep track of the size of the contributions and the number of repeat donations, e.g., **IF \$CONTRIB >="25", IF \$CONTRIB < "100", SET #CONT2=#CONT2+1** (See screen at right)
 3. Between the counters and the TEXT command, embed the conditional report section to be printed at end of file. (See screen at right)
 4. Put a COPY0 command in the SETUP section.
 5. End EDIT and PRINT LESSON8.

```

\DATA \NAME,ADDR1,ADDR2,CITY,STATE,SALUT,CONTRIB,TIMES,,COMMENT-
\COPY 0, ***** Allow printing of final totals at end of job-
\TEXT-
\***** Insert the following section to print totals-
\IF %EOF=0, SKIP 11,* Ignore this section until end of job-
\MP, LN 0, RM 80-
\CTR\Recap of Contribution Acknowledgments-
-
-
Seal1 (under $25) Medium ($25-99) Large ($100-999) Major ($1000 up)-
-
\=#CONT1, TAB 23,=#CONT2, TAB 42,=#CONT3, TAB 61,=#CONT4-
-
-
Total Letters Written — \=#CONT1+#CONT2+#CONT3+#CONT4,-
\FF,QUIT,* Force processing to stop after totals-
\***** End of totals section-
\IF #CONTRIB=0,END-

```

```

-
\IF #CONTRIB=0,END-
\***** Insert the following section to obtain totals-
-
\# Count this contributor in 1 category only-
\IF #CONTRIB<"25",SET #CONT1=#CONT1+1-
\IF #CONTRIB<"25", IF #CONTRIB<"100",SET #CONT2=#CONT2+1-
\IF #CONTRIB<"100", IF #CONTRIB<"1000",SET #CONT3=#CONT3+1-
\IF #CONTRIB<"1000",SET #CONT4=#CONT4+1-
-
\***** End of Counter Section-
-
\RIGHT,DATE1-
\LEFT,LINES-
-
\#PEV-
\#ADDR1-
\IF ADDR2**,:ADDR2.NL-
\#CITY, \#STATE1-
\#LINE14-
-
Dear \#SALUT,-

```

MULTIPLE LETTERS IN THE SAME FILE

- A. You can have more than one letter in a single text file.
- B. The simplest way to arrange the letters in this kind of file is sequentially, one letter following another.
- C. Use the SKIP and SKIP TO commands to select which letters to print.
- D. In many cases, this kind of file can take the place of boilerplating a letter in EDIT.
 1. PRINT SAMPLE7.
 2. Notice that this is REPLIES.GET converted to a PRINT file.
 3. Edit SAMPLE7 and save as LESSON8A. (EDIT SAMPLE7 LESSON8A)
 4. Embed counters to keep track of how many of each kind of letter is printed; have the machine print a report at the end of the data file. (See the screen at right)
 5. End EDIT and PRINT LESSON8A.

```

DATA ,NAME,ADDR1,ADDR2,CITY,STATE,SALUT,,,CODE-
^-
\COPY 2;**** Pass file 1 time after end of data-
^-
\TEXT-
^-
\***** Section to display counts on screen at END-
^-
\IF %EOF=0, SKIP 12-
\CLS: SHOW " Summary of Letters by Class",NL,NL-
\SHOW "CODE MEANING COUNT NL-
\IF #CNTA1>0, SHOW " A1 Influential Supporter - ",#CNTA1-
\IF #CNTA2>0, SHOW " A2 Important Bignitary - ",#CNTA2-
\IF #CNTA3>0, SHOW " A3 Friendly Citizen - ",#CNTA3-
\IF #CNTA4>0, SHOW " A4 Schoolchild - ",#CNTA4-
\IF #CNTZ1>0, SHOW " Z1 Important Opponent - ",#CNTZ1-
\IF #CNTZ2>0, SHOW " Z2 Unfriendly Citizen - ",#CNTZ2-
\IF #CNTZ3>0, SHOW " Z3 Threatening Citizen - ",#CNTZ3-
\SHOW NL," Total Letters " ,#CNT-
\END-
\***** End of count display-
^-
\IF CODE="X" END-

```

```

^-
\IF CODE="X" END-
\IF-
\WP-
^-
\***** Count letters by CODE class-
^-
\SET #CNT=#CNT+1, * Count all-
\IF CODE="A1", SET #CNTA1=#CNTA1+1-
\IF CODE="A2", SET #CNTA2=#CNTA2+1-
\IF CODE="A3", SET #CNTA3=#CNTA3+1-
\IF CODE="A4", SET #CNTA4=#CNTA4+1-
\IF CODE="Z1", SET #CNTZ1=#CNTZ1+1-
\IF CODE="Z2", SET #CNTZ2=#CNTZ2+1-
\IF CODE="Z3", SET #CNTZ3=#CNTZ3+1-
^-
\***** End of Counter Section-
^-
\RIGHT, :DATEY-
\LEFT, :LINE8-
^-
\NAMEY-
\ADDR1Y-

```

LESSON NINE

Lesson Nine shows you how to treat documents with more than one page. In it you learn how to set up a heading or a footing, how to print to disk instead of the printer, how to handle non-standard page lengths, etc. You also get a chance to use a few EDIT commands that were not covered in earlier lessons.

Basic handling of multi-page documents is simple and straightforward. As with most PRINT features, however, you can create some fairly complex functions if you push the commands to their limits. Do not try anything fancy until you are completely comfortable with the basic features.

BLOCK COPY

- A. Set Block Markers around the text to be copied. (See Lesson Two)
- B. Move the cursor to the place you wish to copy the block of text.
- C. Escape to the Command Screen and type **BC (Block Copy)**.
- D. The machine inserts a copy of the block at the cursor location.
- E. You may make as many copies as you want at one time by repeating the BC command.
- F. Return to the Text Screen.
- G. The cursor is positioned on the first character of the first copy.
 1. Edit **SAMPLE1** and save it as **LESSON9**. (Type **EDIT SAMPLE1 LESSON9**)
 2. Set Block Markers around the entire text.
 3. Leave the cursor immediately following the last marker.
 4. Make four copies of the file (**ESCAPE** to the Command Screen and type **BC** four times)
 5. **Note:** This gives you a file that takes up more than one page. Any long file will do for our purposes.
 6. End EDIT and print **LESSON9** without giving any print commands. (Type **PRINT LESSON9**)

It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from these honored dead we take increased devotion to that cause for which they gave their last full measure of devotion that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth."

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia."

I could go on like this all day, but let us be serious for a moment. Eightyseven years ago our ancestors brought forth on this continent a new nation, conceived and dedicated to the proposition that all men are made pretty much the same."

Now we are engaged in a real big civil war, testing whether that nation conceived and dedicated like that can endure for very long. We are meet on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this; and I am not at all sorry that I came."

PAGE LENGTH

- A. To set the number of lines in a page, type **PLn**, where *n* is any number between zero and 255.
 - B. If you do not give a PL command, the machine sets the page length at 66 lines per page, i.e., 11 inches at six lines per inch.
 - C. As the machine processes the file, it keeps track of the lines that have been printed so far on the page.
 - D. The machine continues to process the body of the text until it reaches the bottom margin. (See the following section)
 - E. The machine then processes the footing (if any) and form feeds to the top of the next page. (See below)
 - F. Most printers have a mechanical form feed that the machine triggers to get to the top of the next page.
 - G. The mechanical form feed assumes that the page is a full 66 lines long (unless you change the page setting on your printer).
 - H. If your current page length is not 66 lines, you must turn off the automatic form feed.
 - I. Do this by giving the command **FORMFEED OFF**.
 - J. Now the machine advances to the page line by line instead of using the mechanical form feed.
 - K. Note: This is usually almost as fast as the mechanical form feed.
 1. Set the page length at 40 lines per page and begin to print the file. (**PL40**)
 2. Allow the machine to continue until it begins the second page.
 3. If your printer advanced an entire 66 lines, then you know it is equipped with a mechanical form feed.
 4. Interrupt the processing of the file and inactivate the form feed. (Type **FORMFEED OFF**)
 5. Resume processing.
 6. Interrupt after advancing to the third page and change the page length to 30 lines. (**PL30**)
- Finish printing the file.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia.

I could go on like this all day, but let us be serious for a moment. Eightyseven years ago our ancestors brought forth on this continent a new notion, conceived and dedicated to the proposition that all men are made pretty much the same.

Now we are engaged in a real big civil war, testing whether that nation conceived and dedicated like that can endure for very long. We are meet on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives for that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came.

But, let's face it, we can not dedicate—we can not consecrate—we can not even hollow out—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or subtract, multiply or divide. The world can never forget what they did here.

It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion that this nation, under God, shall have a new birth of

freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia.

I could go on like this all day, but let us be serious for a moment. Eightyseven years ago our ancestors brought forth on this continent a new notion, conceived and dedicated to the proposition that all men are made pretty much the same.

Now we are engaged in a real big civil war, testing whether that nation conceived and dedicated like that can endure for very long. We are meet on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives for that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came.

But, let's face it, we can not dedicate—we can not consecrate—we can not even hollow out—this ground. The brave men, living or dead, who struggled here, have consecrated it, far above our poor power to add or subtract, multiply or divide. The world can never forget what they did here.

It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for

which they gave their last full measure of devotion that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia.

I could go on like this all day, but let us be serious for a moment. Eightyseven years ago our ancestors brought forth on this continent a new notion, conceived and dedicated to the proposition that all men are made pretty much the same.

Now we are engaged in a real big civil war, testing whether that nation conceived and dedicated like that can endure for very long. We are meet on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives for that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came.

TOP AND BOTTOM MARGINS

- A. Each page may be divided into three sections: the top margin, the body of the text and the bottom margin.
- B. The top margin is the section at the top of the page that the machine reserves for the heading and/or spacing.
- C. To set the top margin, type TM_n , where n is any number between zero and the current page length minus the bottom margin.
- D. If you do not give a TM command, the machine sets the top margin at zero.
- E. The bottom margin is the section at the bottom of the page that the machine reserves for the footing and/or spacing.
- F. To set the bottom margin, type BM_n , where n is any number between zero and the current page length minus the top margin.
- G. If you do not give a BM command, the machine sets the bottom margin at six lines.
- H. **Note:** The bottom margin is usually larger than the top margin because it extends to the first few physical lines of the next page.
- I. The machine prints the body of the text between the top margin and the bottom margin, e.g., if your page length is 66 lines and your top margin is three lines and your bottom margin is six lines, you have 57 lines reserved for the body of the text (66 minus 3 minus 6).
- J. In this case, the body of the text begins on line four and ends on line 60.
 - 1. To get a good idea of how these three sections relate to one another, PRINT SAMPLE8 and follow the instructions that it gives you.
 - 2. Continue until you feel completely comfortable with the page arrangement.
 - 3. **Note:** You do not need to do anything else with SAMPLE8, but it might be instructive to examine the file to see how some of the commands are handled.

```

TOP MARGIN LINE 1, 27 remaining
TOP MARGIN LINE 2, 26 remaining
TOP MARGIN LINE 3, 25 remaining
***** BODY OF THE TEXT ***** 24 available.
BODY OF THE TEXT LINE 5, 23 remaining
BODY OF THE TEXT LINE 6, 22 remaining
BODY OF THE TEXT LINE 7, 21 remaining
BODY OF THE TEXT LINE 8, 20 remaining
BODY OF THE TEXT LINE 9, 19 remaining
BODY OF THE TEXT LINE 10, 18 remaining
BODY OF THE TEXT LINE 11, 17 remaining
BODY OF THE TEXT LINE 12, 16 remaining
BODY OF THE TEXT LINE 13, 15 remaining
BODY OF THE TEXT LINE 14, 14 remaining
BODY OF THE TEXT LINE 15, 13 remaining
BODY OF THE TEXT LINE 16, 12 remaining
BODY OF THE TEXT LINE 17, 11 remaining
BODY OF THE TEXT LINE 18, 10 remaining
BODY OF THE TEXT LINE 19, 9 remaining
BODY OF THE TEXT LINE 20, 8 remaining
BODY OF THE TEXT LINE 21, 7 remaining
BODY OF THE TEXT LINE 22, 6 remaining
BODY OF THE TEXT LINE 23, 5 remaining
BODY OF THE TEXT LINE 24, 4 remaining
BODY OF THE TEXT LINE 25, 3 remaining
BODY OF THE TEXT LINE 26, 2 remaining
BODY OF THE TEXT LINE 27, 1 remaining
BOTTOM MARGIN LINE 28, 0 remaining
BOTTOM MARGIN LINE 30, 0 remaining
BOTTOM MARGIN LINE 31, 0 remaining
BOTTOM MARGIN LINE 32, 0 remaining
BOTTOM MARGIN LINE 33, 0 remaining
TOP MARGIN LINE 1, 27 remaining
TOP MARGIN LINE 2, 26 remaining
TOP MARGIN LINE 3, 25 remaining
***** BODY OF THE TEXT ***** 24 available.
BODY OF THE TEXT LINE 5, 23 remaining
BODY OF THE TEXT LINE 6, 22 remaining
BODY OF THE TEXT LINE 7, 21 remaining
BODY OF THE TEXT LINE 8, 20 remaining
BODY OF THE TEXT LINE 9, 19 remaining
BODY OF THE TEXT LINE 10, 18 remaining
BODY OF THE TEXT LINE 11, 17 remaining
BODY OF THE TEXT LINE 12, 16 remaining
BODY OF THE TEXT LINE 13, 15 remaining
BODY OF THE TEXT LINE 14, 14 remaining
BODY OF THE TEXT LINE 15, 13 remaining
BODY OF THE TEXT LINE 16, 12 remaining
BODY OF THE TEXT LINE 17, 11 remaining
BODY OF THE TEXT LINE 18, 10 remaining
BODY OF THE TEXT LINE 19, 9 remaining
BODY OF THE TEXT LINE 20, 8 remaining
BODY OF THE TEXT LINE 21, 7 remaining
BODY OF THE TEXT LINE 22, 6 remaining
BODY OF THE TEXT LINE 23, 5 remaining
BODY OF THE TEXT LINE 24, 4 remaining
BODY OF THE TEXT LINE 25, 3 remaining
BODY OF THE TEXT LINE 26, 2 remaining
BODY OF THE TEXT LINE 27, 1 remaining
BOTTOM MARGIN LINE 28, 0 remaining
BOTTOM MARGIN LINE 29, 0 remaining
BOTTOM MARGIN LINE 30, 0 remaining
BOTTOM MARGIN LINE 31, 0 remaining
BOTTOM MARGIN LINE 32, 0 remaining

```

```

BOTTOM MARGIN LINE 33, 0 remaining
TOP MARGIN LINE 1, 16 remaining
TOP MARGIN LINE 2, 15 remaining
TOP MARGIN LINE 3, 14 remaining
TOP MARGIN LINE 4, 13 remaining
TOP MARGIN LINE 5, 12 remaining
***** BODY OF THE TEXT ***** 12 available.
BODY OF THE TEXT LINE 7, 11 remaining
BODY OF THE TEXT LINE 8, 10 remaining
BODY OF THE TEXT LINE 9, 9 remaining
BODY OF THE TEXT LINE 10, 8 remaining
BODY OF THE TEXT LINE 11, 7 remaining
BODY OF THE TEXT LINE 12, 6 remaining
BODY OF THE TEXT LINE 13, 5 remaining
BODY OF THE TEXT LINE 14, 4 remaining
BODY OF THE TEXT LINE 15, 3 remaining
BODY OF THE TEXT LINE 16, 2 remaining
BODY OF THE TEXT LINE 17, 1 remaining
BOTTOM MARGIN LINE 18, 0 remaining
BOTTOM MARGIN LINE 19, 0 remaining
BOTTOM MARGIN LINE 20, 0 remaining
BOTTOM MARGIN LINE 21, 0 remaining
BOTTOM MARGIN LINE 22, 0 remaining
TOP MARGIN LINE 1, 39 remaining
TOP MARGIN LINE 2, 38 remaining
TOP MARGIN LINE 3, 37 remaining
TOP MARGIN LINE 4, 36 remaining
TOP MARGIN LINE 5, 35 remaining
***** BODY OF THE TEXT ***** 34 available.
BODY OF THE TEXT LINE 7, 33 remaining
BODY OF THE TEXT LINE 8, 32 remaining
BODY OF THE TEXT LINE 9, 31 remaining
BODY OF THE TEXT LINE 10, 30 remaining
BODY OF THE TEXT LINE 11, 29 remaining
BODY OF THE TEXT LINE 12, 28 remaining
BODY OF THE TEXT LINE 13, 27 remaining
BODY OF THE TEXT LINE 14, 26 remaining
BODY OF THE TEXT LINE 15, 25 remaining
BODY OF THE TEXT LINE 16, 24 remaining
BODY OF THE TEXT LINE 17, 23 remaining
BODY OF THE TEXT LINE 18, 22 remaining
BODY OF THE TEXT LINE 19, 21 remaining
BODY OF THE TEXT LINE 20, 20 remaining
BODY OF THE TEXT LINE 21, 19 remaining
BODY OF THE TEXT LINE 22, 18 remaining
BODY OF THE TEXT LINE 23, 17 remaining
BODY OF THE TEXT LINE 24, 16 remaining
BODY OF THE TEXT LINE 25, 15 remaining
BODY OF THE TEXT LINE 26, 14 remaining
BODY OF THE TEXT LINE 27, 13 remaining
BODY OF THE TEXT LINE 28, 12 remaining
BODY OF THE TEXT LINE 29, 11 remaining
BODY OF THE TEXT LINE 30, 10 remaining
BODY OF THE TEXT LINE 31, 9 remaining
BODY OF THE TEXT LINE 32, 8 remaining
BODY OF THE TEXT LINE 33, 7 remaining
BODY OF THE TEXT LINE 34, 6 remaining
BODY OF THE TEXT LINE 35, 5 remaining
BODY OF THE TEXT LINE 36, 4 remaining
BODY OF THE TEXT LINE 37, 3 remaining
BODY OF THE TEXT LINE 38, 2 remaining
BODY OF THE TEXT LINE 39, 1 remaining
BOTTOM MARGIN LINE 40, 0 remaining
BOTTOM MARGIN LINE 41, 0 remaining
BOTTOM MARGIN LINE 42, 0 remaining
BOTTOM MARGIN LINE 43, 0 remaining

```

NUMBERING OF PAGES

- A. The machine keeps track of the current page number with the system variable %PAGE.
- B. Before the machine begins to process the file, %PAGE is one.
- C. At the beginning of each new page, the machine increases %PAGE by one.
- D. %PAGE can be no larger than 32767.
- E. %PAGE is a numeric variable and may be used in numeric expressions, e.g., **SET #VAR=%PAGE+1.**
- F. You can also print %PAGE as part of the text, e.g., if the current value of %PAGE was 4 and you typed **\CTR\page\%PAGE**, the machine would print **page 4** centered on the line.
- G. You may set the value of %PAGE to any number from zero to 32767.
- H. To do so, type **PGn**, e.g., to set %PAGE to 20, type **PG20.**
- I. The machine now numbers the pages in the text beginning with page 20, 21, etc.

%LINES

- A. When the machine gets to the bottom margin, it breaks off processing the body of the text and does not resume until after the top margin of the next page.
- B. It is possible that the machine could break off processing at an undesirable place.
- C. You could print the file through once, and then go back and embed NP commands to force new pages where you needed them.
- D. Or you could take advantage of the %LINES system variable and give a conditional command.
- E. %LINES is the number of lines remaining on a page before the bottom margin, including the current line.
- F. For example, if you were on the next to last line before the bottom margin, the value of %LINES would be 2 (the current line plus the last line).
- G. To give a conditional NP command, type `\IF %LINES<n,NP` where *n* is the minimum number of lines you want remaining on the page.
 1. Edit LESSON9
 2. Embed commands to make sure that the title is never left on one page with the text beginning on another. (Immediately preceding the title, type `\IF %LINES<4,NP`) Note: This gives you room for the title, a blank line and two lines of text.
 3. Embed commands to make sure that no paragraph begins with only one line remaining on the page. (Type `\IF %LINES=1,NP`)
 4. Note: You can save yourself a lot of time if you insert a code, e.g., ZZ, where the command is to go instead of retyping the command each time. After all of the codes are in place, go back to the beginning of the file and perform a MULTIPLE SEARCH AND REPLACE to change the code into the command (`:ZZ:\IF %LINES=1,NP:`).
 5. End EDIT and print LESSON9

```
##-
      LINCOLN'S ADDRESS AT GETTYSBURG, 1863-
~
ZZ-
  It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the
  hospitality you have shown us during our stay. I wish that I could say that
  everyone loves it here. I met a soldier who told me that he would rather have
  died at Gettysburg than have to live in Philadelphia."
ZZ-
  I could go on like this all day, but let us be serious for a moment.
  Eightyseven years ago our ancestors brought forth on this continent a new
  nation, conceived and dedicated to the proposition that all men are made pretty
  much the same."
ZZ-
  Now we are engaged in a real big civil war, testing whether that nation
  conceived and dedicated like that can endure for very long. We are meet on a
  real big battle field of that war. We have come to dedicate some of that
  field, as a memento for those who here gave their lives that that nation might
  live. It is altogether fitting and proper that we should do this, and I am not
  at all sorry that I came."
ZZ-
  But, let's face it; we can not dedicate—we can not consecrate—we can not
```

```
##-
      LINCOLN'S ADDRESS AT GETTYSBURG, 1863-
~
\IF %LINES=1,NP-
  It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the
  hospitality you have shown us during our stay. I wish that I could say that
  everyone loves it here. I met a soldier who told me that he would rather have
  died at Gettysburg than have to live in Philadelphia."
\IF %LINES=1,NP-
  I could go on like this all day, but let us be serious for a moment.
  Eightyseven years ago our ancestors brought forth on this continent a new
  nation, conceived and dedicated to the proposition that all men are made pretty
  much the same."
\IF %LINES=1,NP-
  Now we are engaged in a real big civil war, testing whether that nation
  conceived and dedicated like that can endure for very long. We are meet on a
  real big battle field of that war. We have come to dedicate some of that
  field, as a memento for those who here gave their lives that that nation might
  live. It is altogether fitting and proper that we should do this, and I am not
  at all sorry that I came."
\IF %LINES=1,NP-
  But, let's face it; we can not dedicate—we can not consecrate—we can not
  ##:\IF %LINES=4,NP-
```

HEADINGS

- A. Immediately following the SETUP command, open up the text and type \HEAD
- B. Following the HEAD command, type the series of text and command lines that you want repeated each time you reach the top of a new page.
- C. You may take as many lines as you wish.
- D. **Note:** You can do anything in a heading that you can do in regular text.
- E. When you have finished, count the number of lines (carriage returns) in the heading not including the HEAD command.
- F. Go back and insert that number following HEAD.
- G. Example: if you typed three lines for your heading, you would go back and type HEAD3.
- H. Once you have defined a heading, the machine automatically processes the heading each time it reaches the top of a new page.
- I. The machine does not automatically print the heading on the first page of the file unless you tell it to do so.

If you want to print the heading on the first page, embed an NP command following the TEXT command but before the first true text character.

- K. **Note:** An NP command at this point in the file does not force a page feed or increase the value of %PAGE. If you are printing more than one pass of the file, the machine does force a page feed and increase the value of %PAGE on subsequent passes.
 1. Edit LESSON9.
 2. Insert a SETUP/TEXT section at the beginning of the file.
 3. Set a left margin of 10 and a right margin of 60.
 4. Set a page length of 33 lines. (Type PL33) **Note:** Remember to turn off the mechanical page feed. (Type FORMFEED OFF)
 5. Include a heading that prints the current page number flush right and underlined on the first line of the page. Type:
 \HEAD1
 \RIGHT_page \%PAGE\
 \
 6. Set a top margin of three lines (TM3) to leave room for the heading.
 7. Insert an NP command on the line following the TEXT command.
 8. End EDIT and print the file.
 9. Interrupt the printing and set the page number to 10 and finish printing. (Type PG10)

```

\SETUP~
\LM10,FM10~
\PL33,TM3~
\FORMFEED OFF~
\HEAD 1~
\RIGHT\_page \%PAGE\  

\TEXT~
\NP~
\NP
\NP \%LINES(4,MP~
      LINCOLN'S ADDRESS AT GETTYSBURG, 1863~
~
\NP \%LINES(1,MP~
      It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the
      hospitality you have shown us during our stay. I wish that I could say that
      everyone loves it here. I met a soldier who told me that he would rather have
      died at Gettysburg than have to live in Philadelphia.
      I could go on like this all day, but let us be serious for a moment.
      Eightyseven years ago our ancestors brought forth on this continent a new
      notion, conceived and dedicated to the proposition that all men are made pretty
      much the same.
  
```

page 1

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia. I could go on like this all day, but let us be serious for a moment. Eightyseven years ago our ancestors brought forth on this continent a new notion, conceived and dedicated to the proposition that all men are made pretty much the same.

Now we are engaged in a real big civil war, testing whether that nation conceived and dedicated like that can endure for very long. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a memento for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came.

But, let's face it, we can not dedicate—we can not consecrate—we can not even hollow out—this ground. The brave men, living or dead, who struggled here, have

page 2

consecrated it, far above our poor power to add or subtract, multiply or divide. The world can never forget what they did here.

It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia. I could go on like this all day, but let us be serious for a moment. Eightyseven years ago our ancestors brought forth on this continent a new notion, conceived and dedicated to the proposition that all men are made pretty

FOOTINGS

- A. A footing is just like a heading except it goes at the bottom of the page instead of at the top.
- B. To define a footing type FOOT n , where n is the number of lines to be processed as the footing.
- C. If you have defined a footing, the machine processes it when it reaches the bottom margin, beginning on the first line of the bottom margin.
- D. If you want any space left between the body of the text and the footing, you must include that in the footing itself.
- E. The machine does not increase the value of %PAGE until after it has processed the FOOT routine.
- F. You should make sure that you have left enough of a bottom margin to fit the footing on the page.
- G. The machine assumes you know what you are doing, so if the footing is larger than the bottom margin, it prints it anyway before skipping to the next page.
- H. If your footing goes beyond the bottom margin, the machine skips an entire page before processing the heading on the next page.
 1. Edit LESSON9.
 2. Following the heading, insert a footing that would print the current page number centered between dashes after skipping a line. Type:
 \FOOT1
 \NL,CTR\-\%PAGE\
 3. End EDIT and print the file.

```

\SETUP-
\LNLO,RM60-
\PL3,TH3-
\FORMFEED OFF-
\HEAD 1-
\RIGHT\page \%PAGE-
\FOOT 1-
\NL,CTR\-\%PAGE-
\TEXT-
\VP-
\
\IF %LINES<4,%P-
LINCOLN'S ADDRESS AT GETTYSBURG, 1863-

```

```

\IF %LINES=1,%P-
It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the
hospitality you have shown us during our stay. I wish that I could say that
everyone loves it here. I met a soldier who told me that he would rather have
died at Gettysburg than have to live in Philadelphia.
\IF %LINES=1,%P-
I could go on like this all day, but let us be serious for a moment.
Eightyseven years ago our ancestors brought forth on this continent a new

```

page 1

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia.

I could go on like this all day, but let us be serious for a moment. Eightyseven years ago our ancestors brought forth on this continent a new notion, conceived and dedicated to the proposition that all men are made pretty much the same.

Now we are engaged in a real big civil war, testing whether that nation conceived and dedicated like that can endure for very long. We are met on a real big battle field of that war. We have come to dedicate some of that field, as a moment for those who here gave their lives for that that nation might live. It is altogether fitting and proper that we should do this, and I am not at all sorry that I came.

But, let's face it, we can not dedicate—we can not consecrate—we can not even hallow out—this ground. The brave men, living or dead, who struggled here, have

-1-

page 2

consecrated it, far above our poor power to add or subtract, multiply or divide. The world can never forget what they did here.

It is rather for us to be here dedicated to the real big task before us. It is for us the living, rather, to be dedicated here to the incomplete work which they who fought here have thus far so nobly advanced—that from those honored dead we take increased devotion to that cause for which they gave their last full measure of devotion that this nation, under God, shall have a new birth of freedom—and that government of Americans, by Americans, for Americans, shall not perish from the earth.

LINCOLN'S ADDRESS AT GETTYSBURG, 1863

It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the hospitality you have shown us during our stay. I wish that I could say that everyone loves it here. I met a soldier who told me that he would rather have died at Gettysburg than have to live in Philadelphia.

I could go on like this all day, but let us be serious for a moment. Eightyseven years ago our ancestors brought forth on this continent a new notion, conceived and dedicated to the proposition that all men are made pretty

-2-

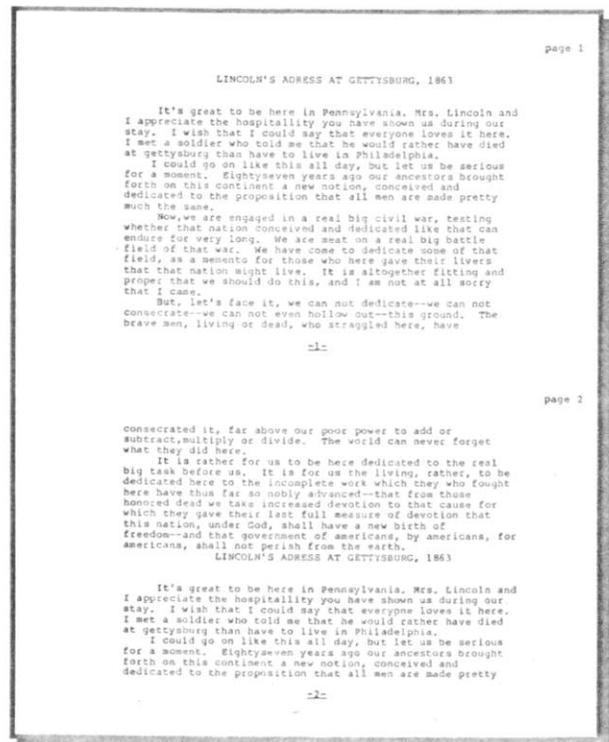
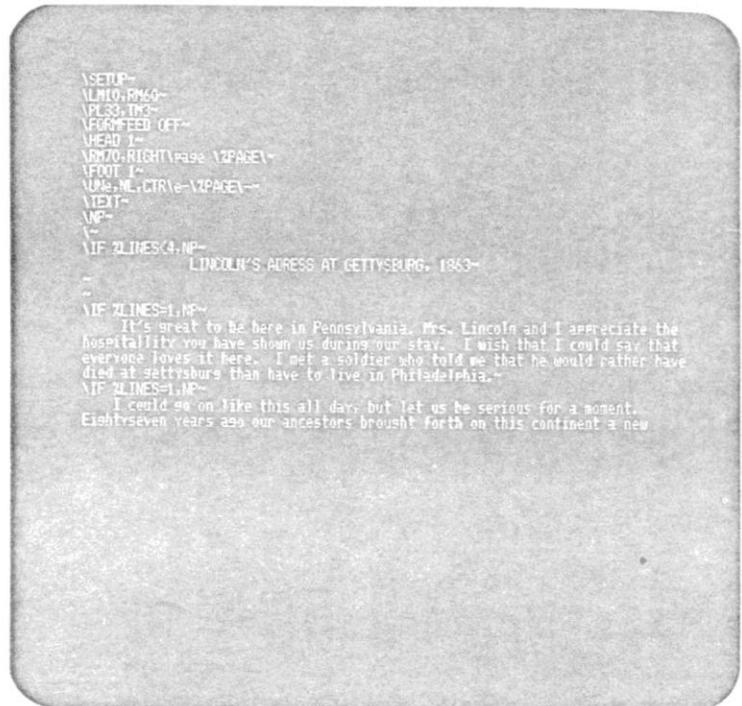
MORE ON HEADINGS

Note: This section applies equally well to footings.

- A. The heading remains the same until you change it.
- B. If you give another HEAD command, you erase the previous heading from memory and replace it with the new one.
- C. Note: To blank out the heading without replacing it, type HEAD0.
- D. The heading operates in its own environment separate from the body of the text or the footing, i.e., you can include shaping and formatting commands in the heading that apply only to the heading.
- E. Before the machine processes the heading, it stores the current shaping and formatting parameters for the body of the text in memory.
- F. When the heading is complete, the machine restores the shaping and formatting parameters to their former values.

Note: This is why the machine went back to LEFT format, even though you had set a RIGHT format for the heading.

- H. If you do not change a shaping or formatting parameter in the heading, the machine uses the current parameters coming from the body of the text.
- I. You may end a heading before the machine processes all of the lines by using the END command.
- J. The END command as part of the heading returns you immediately to the body of the text.
- K. Note: You cannot end a pass from a heading, although you can QUIT from a heading.
 1. Edit LESSON9.
 2. Change the heading so that it has a right margin of 70 with the date printed flush right.
(\RM70,RIGHT\page\%PAGE\)
 3. Change the footing so that lowercase e is the recognition character for underlining, and underline the page number.
(\UNe,NL,CTR\e-\%PAGE\ -)
 4. End EDIT and print LESSON9.
 5. Notice how the commands that you give in the heading do not affect the body of the text.



PRINTING TO DISK

- A. You may save the processed PRINT file on disk as well as or instead of printing it on the printer.
- B. To save the processed file on disk, type **DISK ON**.
- C. When you give this command, the machine opens a file on the same disk as your primary text file, with the same name as your text file but with a PRN file type, e.g., if your file is named LESSON9, the machine opens a file named LESSON9.PRN.
- D. If you want the file saved under a different name or on a different disk drive, type **DISK FILENAME** where FILENAME is any legitimate, unused name.
- E. To stop printing to disk, type **DISK OFF**.
- F. You may only print to one disk file per session; however, you may start and stop printing to the same file as many times as you wish.
- G. **Note:** After you have opened a disk file, the machine interprets a **DISK FILENAME** command as **DISK ON**.
- H. To stop printing on the printer, type **PRINT OFF**.
- I. To resume printing, type **PRINT ON**.
- J. You can use this ability to switch back and forth from the printer to the disk in various ways, some of which are discussed in the notes.
 1. Edit LESSON9.
 2. Add to the heading routine the command to SHOW the current value of %PAGE. (Type SHOW "Page ", %PAGE) **Note:** If you put this on a separate line, be sure to change the number of lines in the HEAD command.
 3. End EDIT and begin to print LESSON9.
 4. When the machine pauses before processing the text, turn off the printer and turn on the disk. (Type PRINT OFF, DISK ON)
 5. Process the file.
 6. Notice how fast the machine can process a file when it does not have to print it.

```

\SETUP-
\LM10,RM40-
\PL33,TM3-
\FORMFEED OFF-
\HEAD 2-
\RM70,RIGHT\page \ZPAGE\
\SHOW "Page ",ZPAGE-
\FDOT 1-
\LINE.NL.CTR\e-\ZPAGE\--
\TEXT-
\MP-
\
\IF ZLINES<4,MP-
      LINCOLN'S ADDRESS AT GETTYSBURG, 1863-
-
-
\IF ZLINES=1,MP-
  " It's great to be here in Pennsylvania. Mrs. Lincoln and I appreciate the
  hospitality you have shown us during our stay. I wish that I could say that
  everyone loves it here. I met a soldier who told me that he would rather have
  died at Gettysburg than have to live in Philadelphia."
\IF ZLINES=1,MP-
  I could go on like this all day, but let us be serious for a moment.

```

TM - Top Margin	3	Form	Continuous	Ignore Char	
BM - Bottom Margin	6	Lines/In	6	Cond. Hyphen &	
LM - Left Margin	10	Chars/In	10	Underscore	Solid
RM - Right Margin	60	Page	1	Boldface	Intensity
IN - Indentation	0	Line	0	SSA (Above)	<
PI - Para Indent	0	Column	1	SCB (Below)	>
SP - Spacing	1	Source	Text	Crossed Mark	\

Print off, disk on

A

```

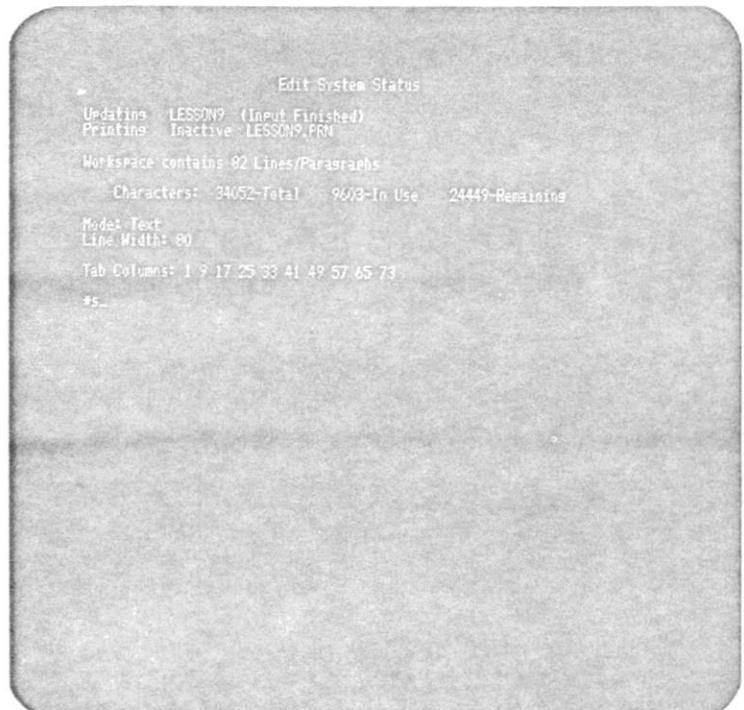
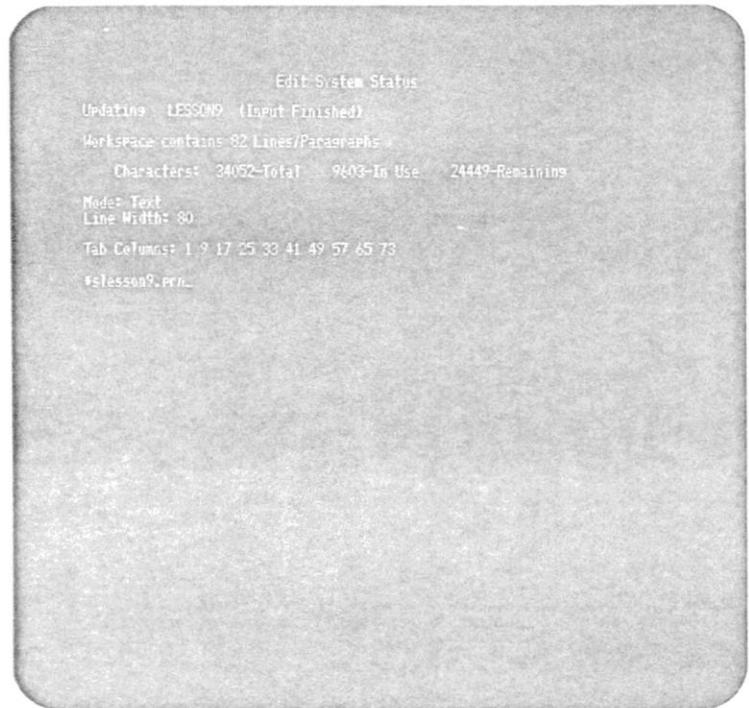
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
Page 8
End of Job

```

A.

PRINTING IN BACKGROUND

- A. You have just created a PRINT output file called LESSON9.PRN.
- B. If you wish, you can print that file while you edit another file.
- C. To do so, begin EDIT for the file you want to work on.
- D. When you reach the Command Screen, type **S**LESSON9.PRN.
- E. Note: The **S** is for **spool printing**, a computer term for printing a file without processing it.
- F. The machine may give you a message that LESSON9.PRN contains non-standard characters.
- G. If this happens, just press RETURN, since your printer can interpret the non-standard characters in a PRINT output file.
- H. The machine reprints the Command Screen with LESSON9.PRN listed as an inactive Printing file.
- I. To begin printing, type **S**.
- J. Edit the file as you normally would.
- K. To halt printing, type **SX**; to resume printing at the same spot, type **S** again.
- L. Note: While the print file is active, the keyboard will probably be considerably more sluggish than usual. Go slowly.
 1. Proceed with the next section in the lesson. (**EDIT LESSON9**)
 2. Once in the Command Screen, establish LESSON9.PRN as a background file. (Type **S**LESSON9.PRN)
 3. Begin printing LESSON9.PRN. (Type **S**)
 4. Continue with the next lesson.
 5. If the file stops printing before you have finished EDIT, start it again by repeating the same commands as above. (**S**LESSON9.PRN and **S**)



START/STOP COMMANDS

- A. If your text file does not access an external data file, the machine interprets the START and STOP commands as *page* numbers instead of *record* numbers.
- B. To start processing the text on a particular page, type **START n** , where n is the page on which you wish to start processing.
- C. To stop processing on a particular page, type **STOP n** .
- D. When you give the machine a START command, it processes the file as usual, but it does not output to either printer or disk until %PAGE matches the number in the START command.
- E. When the machine reaches the page that you designated, it begins to output the file, either to disk and/or to the printer, whichever you have previously specified.
- F. The machine stops processing as soon as %PAGE exceeds the number in the STOP command, i.e., as soon as it has processed the footing for the indicated page number.
- G. **Note:** If you have given a PG command so that your file does not begin on page one, the number in a START or STOP command must correspond to the value of %PAGE and not to the physical page number. Example: To print the tenth page of a file that begins on page eleven (PG11), you would type **START20**.
 1. Print the third through fifth pages of LESSON9. (**START3,STOP5**)
 2. Start numbering LESSON9 with page 10 and print the third through fifth pages of the file. (**PG10,START12,STOP14**)

```

Format: Left Flush      Printer: Your Printer
PL - Page Length 66    Diskout:
TM - Top Margin 3      Forms Continuous
BM - Bottom Marg 6     Lines/In 6
LM - Left Margin 10    Chars/In 10
RM - Right Margin 60   Page 1
IN - Indentation 0     Line 0
PI - Para Indent 0     Column 1
SP - Spacing 1         Source Text
# Variables 0
Ignore Char
Cond. Hraph &
Underscore Solid
Boldface Intensit 1
SSA (Above) <
SSB (Below) >
Command Mark \

```

```
Astart3,stop5
```

```

\
Page 1
Page 2
Page 3
Page 4
Page 5
End of Job

```

```
A_
```

```

Format: Left Flush      Printer: Your Printer
PL - Page Length 66    Diskout:
TM - Top Margin 3      Forms Continuous
BM - Bottom Marg 6     Lines/In 6
LM - Left Margin 10    Chars/In 10
RM - Right Margin 60   Page 1
IN - Indentation 0     Line 0
PI - Para Indent 0     Column 1
SP - Spacing 1         Source Text
# Variables 0
Ignore Char
Cond. Hraph &
Underscore Solid
Boldface Intensit 1
SSA (Above) <
SSB (Below) >
Command Mark \

```

```
Apg10,start12,stop14
```

```

\
Page 10
Page 11
Page 12
Page 13
Page 14
End of Job

```

```
A_
```

BOILERPLATING PRINT COMMANDS

- A. As you become familiar with the features of the **MAGIC WAND**, you will probably develop standard ways to do things.
- B. Once you have devised such an application, you can save it as part of a formatted **INCLUDE** file. (see **LESSON THREE**)
- C. You can use the **INCLUDE** sequence to *boilerplate* the commands into the text you are working on.
- D. This can save you a great deal of time.
- E. It can also insure that all of the commands are repeated in the correct sequence.
- F. You have on your disk such a file called **FORMATS**.
- G. In it you will find several initial sections, headings, footings, file statements, etc.
- H. We give you this file without additional comment.
- I. Try including some of them in the files you have created in the various lessons.
- J. Hopefully, they will give you some ideas about how the **MAGIC WAND** can be used to meet your special needs.

CP/M® FOR MAGIC WAND USERS

CP/M is the registered trademark of Digital Research Corp. for the operating system it developed for 8080, Z80 and 8085 microprocessors. It was designed primarily for programmers and can be a bit bewildering to the uninitiated. What follows is a brief introduction to CP/M for the convenience of our business users. It is not intended as an exhaustive guide, but it should help you get the most out of your **MAGIC WAND**.

CP/M communicates with the user in the most rudimentary way. You type in a command which the machine accepts or rejects. If CP/M rejects your command, the machine prints the command on the screen followed by a question mark. CP/M does not tell you what was wrong with your command.

CP/M does not tell you when a legitimate command has not been executed. For example, you want to erase the file named **RESPONSE.BAK**, but you type instead **ERA RESPONSE.BAL**. CP/M does not tell you that the command could not be carried out. Unless you check the directory on that disk, you don't know that **RESPONSE.BAK** is still there. Or, for example, you request a directory for **DRIVE B**, but you have not loaded a disk on that drive. CP/M attempts to go to **DRIVE B** and waits until you load a disk. And waits. And waits. But it does not tell you that you need to load a disk.

CP/M limits the size of an individual file. This should not bother you, since you are allowed as many as 256,000 characters per file. Should you have a file larger than that, you would be well-advised to break it into sub-files.

CP/M also limits the number of files on a single disk (usually to 64). If you have many small files on a disk, you may find that you are unable to add a file even though you still have room on your disk. You may overcome this limitation by erasing files you no longer need, combining files for storage on historical disks, etc.

BASIC DISK OPERATING SYSTEM (BDOS) ERROR MESSAGES

Even the bravest of programmers can tremble with fear when a BDOS Error Message comes on the screen. You need to know what they are so that you can avoid them if possible and recover from them if they occur.

BDOS ERROR ON (DISK DRIVE): R/O

1. R/O stands for Read Only, meaning you can read a file from the disk, but you cannot do

- anything that would change the disk, i.e., saving, copying, renaming or killing a file.
2. The message occurs when you try to write on a disk that is not properly mounted, e.g., you have changed disks on the drive to which you are trying to write without pressing **CONTROL C** or **RESET**.
3. A like message occurs when using a *write-protected* disk, i.e., a disk that you have protected against accidental erasures.
4. When you receive this error message, pressing any key on the keyboard causes the machine to reload CP/M from the current disk which effectively wipes out the memory.
5. If you are trying to save a file by ending **EDIT** when this error message occurs, you lose the file you are trying to save.
6. This error will not occur if you do not change disks after **EDIT** gives you the opportunity to do so.
7. **Note:** If you must change a disk after beginning **EDIT**, e.g., to access an **INCLUDE** or **DISPLAY** file, be sure that you do not load the new disk on the drive on which you plan to save your edited file.

BDOS ERROR ON (DISK DRIVE): BAD SECTOR

1. This indicates that the machine is having a hard time reading from the disk.
2. The problem may be in the hardware itself, or it may be caused by a worn, creased or otherwise damaged disk.
3. When you receive a Bad Sector message, you may tell the machine to try to read the sector again by pressing **R**.
4. If the machine still cannot read the sector, you will get another message.
5. There is no limit to the number of times you can press **R**; but there is no guarantee that the machine can ever read the problem sector.
6. Instead of retrying, you may tell the machine to read what it can and continue by pressing **RETURN**.
7. If you press **RETURN**, you lose some of your file (up to 128 characters), so you should not do this until you have retried and failed several times.
8. If you encounter a Bad Sector message at a non-crucial time, you may escape and start over by pressing **CONTROL C**.
9. If you receive a Bad Sector message on a disk that has previously given you no prob-

lem, it is a good idea to copy that disk as soon as possible before a small problem gets worse.

10. Often you can erase a disk that has had Bad Sector problems and reuse it without further difficulty.

BDOS ERROR ON (DISK DRIVE): SELECT

1. This message occurs when you have tried to read from or write to a non-existent disk drive, e.g., DRIVE C in a two-drive system.
2. There is no recovery from this error, since pressing any key causes the machine to reload CP/M from the current drive.

CP/M CONTROL KEYS

CONTROL C

1. Pressing CONTROL C ends the current CP/M function and reloads CP/M from the current drive. (See below.)
2. If the disk on the current drive does not have a copy of CP/M, the machine waits until you load a disk containing CP/M.
3. CONTROL C resets the disks currently on the drives so that you can write to them. (See BDOS Error Messages — R/O).

CONTROL X

1. Pressing CONTROL X voids the current command line and moves the cursor to the beginning of the next line, allowing you to retype your command.
2. If you make a mistake while typing a CP/M command, we advise you to press CONTROL X and retype the command from scratch rather than attempting to delete the mistake.
3. Trying to delete may be confusing because CP/M echoes the deleted characters on the screen. Example: You want to type STAT, but instead type STZT. If you delete the ZT and type in AT, the final command line reads STZTTZAT.

CONTROL S

1. Pressing CONTROL S halts any printing in progress whether the machine is printing on the screen or printer.
2. Pressing any key on the keyboard (including CONTROL S) restarts the printing process.
3. Since many printers print from a small internal memory, they may not stop imme-

diately when you press CONTROL S since they continue until their memory is empty.

4. As your disks become crowded, you may need to use CONTROL S as you examine the directory or status of a disk.

CONTROL P

1. Pressing CONTROL P causes the machine to print on the printer as well as the screen.
2. Pressing CONTROL P again turns off the printer.

CP/M COMMANDS

COPY: The COPY command is actually a small program. Using the COPY command is the only way to copy CP/M from one disk to another. It is also the fastest way to copy all of the files from one disk to another. The exact format for this command varies depending upon the manufacturer of your disk drive. Check with your dealer to find out how to use the COPY command for your system.

DIRECTORY: This command allows you to examine the names of the files on a disk. The format for the directory command is:

DIR DRIVE:

Example: You are getting ready to create a series of boilerplate letters using EDIT, and you want to make sure you have mounted the correct disk on DRIVE B. You type **DIR B:**

DRIVE: This command allow you to change the current drive. When you first load CP/M, the current drive is always DRIVE A, which means that when you do not indicate a drive, the machine assumes you are referring to DRIVE A. Under most circumstances, this should be quite sufficient. But if you want to change your current drive the format is:

DRIVE:

Example: You wish to switch your current drive to DRIVE B. You type **B:**. The machine prints **B>** on the screen indicating that DRIVE B is now the current drive.

ERASE: This command allows you to kill any file on disk. **Note:** Once you have killed a file there is no way to recover it. The format for the erase command is:

ERA Drive: FILENAME

Example 1: Your disk on DRIVE B is getting crowded and you want to make

more room by erasing the backup file for the Jackson brief. You type **ERA B:JACKSON.BAK.**

Example 2: You wish to erase all of the files on DRIVE A whose *File Name* begins with **REPLY**. You type **ERA REPLY???.***

PIP: This command allows you to copy any file from one disk to another, or to the same disk under a different name. Like **COPY**, **PIP** is a program rather than a function of **CP/M**, so you must have a copy of **PIP.COM** on one of your disks in order to use it. The format for the command is:

Drive: **PIP Object Drive: NEWNAME = Source Drive:OLDNAME**

If you want to copy all of the files from one disk to another, the format is:

Drive: **PIP Object Drive: = Source Drive:*.***

Example 1: You wish to set up a new disk with the **MAGIC WAND** programs (**EDIT.COM** and **PRINT.COM**) as well as other **CP/M** programs like **PIP.COM** and **STAT.COM**. You load the original disk on DRIVE A and the new disk on DRIVE B. Press **CONTROL C** to remount the disks. Type **PIP B:=A:*.COM.** (This tells the machine to copy all files with a **COM** *File Type* regardless of the *File Name*.) The machine prints the name of each program as it copies it, and when it is finished, it reloads **CP/M** from the current drive.

Example 2: You have a series of files on DRIVE B with the filename of **REPLIES**, distinguished by different filetypes, e.g., **REPLIES.ADV**, **REPLIES.MKG**, etc. You wish to copy all of them to a disk on DRIVE A that contains **PIP.COM**. Press **CONTROL C** to mount the disks. Type **PIP A:=B:REPLIES.*** (This tells the machine to copy all files with a **REPLIES** *File Name* regardless of the *File Type*.) The machine proceeds as in example 1.

Example 3: You have a series of files on DRIVE B named **SAMPLE 1** through **SAMPLE 12** which you wish to copy to DRIVE A. Your copy of **PIP.COM** is on DRIVE B. Press **CONTROL C** to mount the disks. Type **B:PIP A:=B:SAMPLE??.*** (This tells the machine to copy all files whose *File*

Name begins with **SAMPLE** regardless of how it ends or what *File Type* it has.) Proceed as in Example 1.

Example 4: You wish to copy a file named **RESPONSE.STD** from one disk to another, but neither disk has a copy of **PIP.COM** on it. Load the Object disk (i.e., the disk to which you want to copy) on DRIVE B and a disk containing **PIP.COM** on DRIVE A. Press **CONTROL C** to mount disks. Type **PIP.** The machine prints an asterisk and awaits your command. Remove the disk from DRIVE A and load the disk containing **RESPONSE.STD** on DRIVE A. Type **B:=A:RESPONSE.STD.** When an asterisk returns again, press **CONTROL C** to end **PIP.COM**. **Note:** You do not get a Read Only **BDOS Error** on DRIVE A because you are only reading **RESPONSE.STD** and not writing to the disk.

RENAME: This command allows you to rename any file on disk as any other unused legal file name. (See the section in the notes on Naming Files.) The format for the rename command is:

REN Drive: **NEWNAME = OLDNAME**

Example 1: You have a file on DRIVE A named **REPLIES.ADV** which is obsolete but which you wish to save for your historical files. You type **REN REPLYADV.HIS = REPLIES.ADV**

Example 2: You have a file **JACKSON.***** on DRIVE B which was saved by using **QUITX** to exit from **EDIT** while you were working on the Jackson brief. Now you need to rename it. You type **REN B:JACKSON.SAV = JACKSON.*****

STATUS: This command allows you to find out how much space you have on a disk, the names of the files on that disk and how much space each of them occupies. Like **COPY** and **PIP**, this is a small program rather than an integral function of **CP/M**, so you must have a copy of **STAT.COM** on a disk in use. The format for the **STAT** command is:

Drive: **STAT Drive: *.***

If you just need to know how much room you have on a disk but do not need any information about the files, the format is:

Drive: **STAT Drive:**

Example 1: You need to copy some files on DRIVE A to your historical disk on DRIVE B, and you want to find out whether there is enough room. Your copy of STAT.COM is on the historical disk. You type **B:STAT B:** to find out how much room you have left on your historical disk. Then you type **B:STAT *.*** to find out how much space the files occupy.

Example 2: You want to determine the number and size of the BAK files on DRIVE A. You type **STAT *.BAK.**

TYPE: This command allows you to display

the contents of a file on the screen. If you press **CONTROL P** in conjunction with this command, you may print the file on the printer. The format for the type command is:

TYPE Drive: FILENAME

Example: You have a formatted file created by **PRINT** called **STDREPLY.PRN** which you now want to print on the printer. You type **TYPE STDREPLY.PRN.** Before you press **RETURN** you press **CONTROL P** to turn on the printer.

NOTES ON EDIT

LOADING EDIT PROGRAM/TEXT FILE

1. The complete format for loading the EDIT program and the text file is:
(*Drive*) EDIT (*Drive* :) FILENAME
((*Drive* :) NEWNAME)
2. Indicate the drive if the disk is not on the current drive. (See Glossary under **current drive**.)
3. Include *NEWNAME* when you want to save the edited file under a name other than **FILENAME**.
4. To load from one drive and save on another, just type the new Drive followed by a colon, e.g., EDIT SAMPLE1 B: loads file SAMPLE1 from the current drive and saves the edited SAMPLE1 on DRIVE B.
Note: The SAMPLE1 on the current drive is unchanged.
5. After the machine has loaded EDIT, it prints the sign-on screen, which contains such information as the release number, the version and the serial number of the system you are using.
6. After printing the sign-on screen, the machine gives you the opportunity to change disks before continuing.
7. This means that you do not need to have a copy of EDIT mounted on a disk drive while you are editing.
8. When you have loaded the correct disks, press RETURN.
9. **Note:** If you want to abort EDIT at this point, e.g., you realize you have typed **FILENAME** incorrectly, press ESCAPE instead of RETURN; and the machine returns to CP/M.
10. After you press RETURN, the machine resets the disk drives so that they are no longer **Read Only**. (See *CP/M for MAGIC WAND Users*).
11. **Note:** After this point, *never* change the disk on the drive on which you plan to save your edited file, e.g., if your edited file is to be saved on DRIVE B, you may change the disks on DRIVE A as many times as you wish, but not on DRIVE B.
12. If **FILENAME** is not in the specified disk, the machine asks you if it is a new file.
13. If **FILENAME** is a new file, press **Y**, and the machine creates a place for it in the disk directory.
14. If **FILENAME** is *not* a new file, press **N**, and the machine aborts EDIT and returns to CP/M.

NAMING A FILE:

1. **FILENAME** is composed of a *File Name* and a *File Type*.
2. The *File Name* may be up to eight characters.
3. Any character may be part of the *File Name* except a blank, asterisk, question mark or period.
4. Since the *File Name* should be meaningful to you, you should not need to use other than uppercase letters, numbers, and a few punctuation marks such as slash(/), hyphen(-), parentheses and brackets.
5. Including a *File Type* is optional; however, it can help you organize your files.
6. The *File Type* may be up to three characters.
7. You can use the same characters for the *File Type* as for the *File Name*.
8. To include a *File Type*, follow the *File Name* with a period followed by the *File Type*, e.g., INQUIRY.LTR, JACKSON.BRF, ANUALRPT.STD.
9. The period is not necessary if you do not include a file type.
10. **BAK** and **\$\$\$** are reserved *File Types* and may not be used to designate an output file in EDIT.
11. If you wish to edit a **BAK** or **\$\$\$** file, you must save the edited file under another name, or rename the file before attempting to edit it.
12. If your output file has any of the following *File Types*, the machine begins in program mode (see section on Special Formatting): **ASC**, **ASM**, **BAS**, **COB**, **FOR**, **MAC** or **PRN**.
13. When referring to **FILENAME**, you must include the Drive if it is other than the current drive, e.g., B:FILENAME.

CURSOR MOVEMENT

1. You have Control Keys to move the cursor UP, DOWN, LEFT and RIGHT on the Text Screen.
2. When you press one of these keys, the cursor changes its position on the screen without altering the text.
3. Pressing a cursor movement Control Key moves the cursor one space in the indicated direction with the following exceptions:
 - a. Pressing the UP Control Key when the cursor is on the top line of the screen

- causes the text to shift backward one line.
- b. If the cursor is at the top of text, when this occurs, the machine rewrites the current screen but makes no changes to it.
 - c. Pressing the DOWN Control Key when the cursor is on the bottom line of the screen causes the text to shift forward one line.
 - d. You may move the cursor DOWN only *one line* beyond the end of the text. The machine ignores any attempt to move cursor beyond this point.
 - e. Pressing the RIGHT Control Key when the cursor is on the last position on a line causes the cursor to wrap around to the first position of the next line.
 - f. If the cursor is on the last line on the screen when this occurs, the machine also moves the text forward one line unless the cursor is already beyond the end of the text.
 - g. Pressing the LEFT Control Key when the cursor is on the first position of a line causes the cursor to wrap around to the last position of the preceding line.
 - h. If the cursor is on the top line on the screen when this occurs, the machine also moves the text backward one line unless the cursor is on first line of text.
4. The TAB and HOME Control Keys also help you position the cursor on the Text Screen.
 5. When you press the TAB Control Key, the cursor jumps forward to the next tab marker setting (see Section on Tab Setting)
 6. If the cursor is beyond the last tab marker on a line, pressing the TAB Control Key jumps the cursor to the first position of the next line.
 7. When you press the HOME Control Key, the cursor moves to the first position of the line it is on.
 8. If you press the HOME Control Key when the cursor is already at the first position of a line, the cursor moves to the first position of the top line on the screen.
 9. If you press the HOME Control Key when the cursor is already at the first position of the top line, the command has no effect.
2. Moving forward or backward through the text file is called *scrolling*, probably because you can roll the text forward and backward as if it were on a scroll.
 3. Scrolling changes the position of the text on the Text Screen without changing the location of the cursor on the screen.
 4. The FORWARD LINE SCROLL Control Key shifts the text currently on the screen **up** one line and writes the next line of the text file on the last line of the screen.
 5. If the cursor is beyond the end of text when you press the FORWARD LINE SCROLL Control Key, the command has no effect.
 6. The BACKWARD LINE SCROLL Control Key moves the text currently on the screen **down** one line adding a new line at the top of the screen.
 7. If you press the BACKWARD LINE SCROLL Control Key when the first line of the text is already on the screen, the machine rewrites the screen, but otherwise changes nothing.
 8. The FORWARD PAGE SCROLL Control Key clears the screen and rewrites the screen starting with the last line of text on the previous screen.
 9. If you press the FORWARD PAGE SCROLL Control Key with less than a full screen remaining in the text file, the machine rewrites the last screen with the last line of text on the same line as the cursor.
 10. The BACKWARD PAGE SCROLL Control Key clears the screen and rewrites the text so that the *first* line on the previous screen becomes the *last* line on the new screen.
 11. If you press the BACKWARD PAGE SCROLL Control Key with less than a full screen remaining at the beginning of the text file, the machine writes the first screenful of text from the file.
 12. **Note:** On many terminals when you press the BACKWARD LINE SCROLL, the machine must rewrite the entire screen. Often this is slow; and if you need to back up several lines, you may find it quicker to do a single BACKWARD PAGE SCROLL followed by several FORWARD LINE SCROLLS.
 13. Pressing the TOP Control Key prints the first screen of the text with the cursor at the first position of the first line.
 14. Pressing the BOTTOM Control Key prints the last screen of text with the cursor located on the line following the end of the text.

TEXT MOVEMENT (SCROLLING)

1. Since you can only change text that is on the Text Screen, you need to be able to move new text onto the screen.

INSERT (Character and Full)

1. You have two ways that you can insert characters into a text — CHARACTER INSERT, for inserting character by character, and FULL INSERT, for longer insertions.
2. To use either of the INSERT Control Keys, you must first move the cursor to the character in front of which you wish to make the insertion, e.g., to insert an **a** in **bet** to make **beat**, move the cursor to the **t**.
3. When the cursor is in the correct position, press the CHARACTER INSERT Control Key and type the character or characters you wish to insert.
4. Each time you press a key, the machine echoes the character on the screen at the location of the cursor, shifting all of the characters from the cursor to the end of the line one space to the right.
5. The cursor remains positioned over the same character as when you pressed the CHARACTER INSERT Control Key.
6. The machine continues to insert characters as you type them until you press any other Control Key — cursor movement, scrolling, backspacing, deletion, etc.
7. **Note:** As you insert characters, you may push words to the next line. If your terminal is the kind that prints out characters one at a time, this may be time consuming, and, if you are not careful, you may skip characters. Even if your terminal does not print characters one at a time, the response is sluggish compared to normal typing.
8. For this reason use the FULL INSERT Control Key when you need to make long insertions.
9. After positioning the cursor on the character in front of which you wish to insert, press the FULL INSERT Control Key.
10. The machine clears the screen from the cursor to the bottom of the screen, printing the line of text following the cursor on the message line at the bottom of the screen for reference.
11. **Note:** If the cursor is in void space (see glossary) when you press the FULL INSERT Control Key, the machine inserts an extra blank in the text.
12. You may now type the insertion as if you were at the end of the text, being able to use most of the Control Keys.
13. If you press the SEARCH, TOP, BOTTOM or ESCAPE Control Keys, the machine ends FULL INSERT before the command is executed.
14. You may not use the DOWN, FORWARD LINE SCROLL or FORWARD PAGE SCROLL Control Keys to move the cursor more than one line past the last line of the insertion.
15. When you finish making your insertion, press the END INSERT Control Key.
16. When you end FULL INSERT, the machine fills in the rest of the screen with the subsequent text.
17. If you insert more than the screen will hold, the text scrolls upward as it does at end of text. When this happens, the machine blanks out the reference line at the bottom of the screen to avoid confusion.
18. If you are in the middle of a FULL INSERT, pressing the FULL INSERT Control Key has no effect.

DELETION (Character and Line)

1. To delete a single character, move the cursor to that character and press the CHARACTER DELETE Control Key.
2. The machine deletes the character and moves all characters on that line to the right of the cursor one space to the left. (The cursor does not move.)
3. If this makes room for the first word of the following line, then the machine moves that word up and rewrites the next line, etc.
4. You may delete any character, including control characters, with the CHARACTER DELETE Control Key, except the carriage return.
5. A carriage return can only be deleted by pressing the LINE DELETE Control Key.
6. Pressing the LINE DELETE Control Key deletes the line from the cursor to the end of the line and moves the remainder of the text up to the cursor.
7. **Note:** The machine only deletes the line if you have pressed the LINE DELETE Control Key two consecutive times. This prevents deleting a line if you press the LINE DELETE Control Key by mistake.
8. If you press either of the delete Control Keys when the cursor is located to the right of a carriage return on a line, the command has no effect.
9. **Note:** If you delete the space between the last word on a line and the first word on the following line, they form a single word which may not fit on the line. If this happens, the machine rewrites the next line with the two words joined at the beginning.

SEARCH

1. Pressing the SEARCH Control Key allows you to look for the next occurrence of a string of characters following the current cursor position. (See glossary under *string*.)
2. When you press the SEARCH Control Key, the machine drops the cursor to the message line at the bottom of the screen, prints a colon (:) and waits for you to type your search string.
3. Type in the search string *exactly* and press RETURN.
4. The machine searches through the text until it finds the next occurrence of the search string and reprints the screen with the cursor on the first character of the string.
5. The machine prints the screen so that the cursor is on the same line as it was when you pressed the SEARCH Control Key, i.e., if the cursor was on the third line when the search began, the machine scrolls the text forward and prints the screen so that the next occurrence of the search string is on the third line.
6. The machine can only find a search string that is between the cursor and the end of the text. If it is located between the cursor and the beginning of text, the machine cannot find it.
7. If the machine does not find the search string, it prints **Cannot find SEARCH STRING** on the command line and returns the cursor to its original location on the Text Screen.
8. In addition to the characters on the keyboard, the string may contain the PAGE FEED, BLOCK MARKER and RETURN Control Characters.
9. Since pressing RETURN begins the search, the TAB Control Key substitutes for a RETURN as part of your search string, e.g., to search for the next carriage return, you press the TAB Control Key, followed by RETURN to activate the search.
10. If you need to correct your search string, you may move the cursor back using the BACK SPACE Control Key and retype the line.
11. After you have backed up the cursor, the machine ignores all characters to the right of the cursor, beginning with the cursor itself.
12. To abort a SEARCH command, press ESCAPE. (Note: This is the only time that pressing ESCAPE does not return you to the Command Screen.)
13. The machine makes no assumptions about the string for which you are searching, i.e., it does not assume that you are looking for a complete word or that an uppercase letter is the same as a lower case letter. **Example:** If you had **the** as your string, the machine would go to **there** and **other** but would not go to **The**.
14. It is up to you to make your string unique. In the above example the best way to designate the word **the** is " **the** " because the leading and trailing blanks set it off as a word.

SEARCH AND REPLACE

1. When you give the SEARCH AND REPLACE command, the machine looks between the cursor and the end of text for the next occurrence of the search string and replaces it with the replacement string.
2. Define your search string as described in the SEARCH section.
3. Instead of pressing RETURN after typing the SEARCH string, press the SEARCH Control Key again.
4. The machine prints a second colon behind the search string on the message line and waits for you to type the replacement string.
5. Type the replacement string exactly as you wish it to appear, following the rules for the search string outlined above.
6. If you wish to delete the search string, press RETURN without entering a replacement string.
7. If you back the cursor over the second colon, you inactivate the REPLACE command.
8. To reactivate the REPLACE Command, you must press the SEARCH Control Key again and reenter the replacement string.
9. When both the search and replacement strings are exactly as you wish them to be, press RETURN.
10. The machine reprints the screen as in the SEARCH command, but the cursor is located immediately following the replacement string.

MULTIPLE SEARCH AND REPLACE

1. When you give the MULTIPLE SEARCH AND REPLACE command, the machine looks between the cursor and the end of text for occurrences of the search string and replaces them with the replacement string.

2. You may either specify a number of occurrences to change or indicate that you want *all* occurrences to be changed.
3. To execute the MULTIPLE SEARCH AND REPLACE command, define your search and replacement strings as described above.
4. After typing the replacement string, instead of pressing RETURN, press the SEARCH Control Key once again.
5. The machine prints a third colon on the message line following the replacement string.
6. Type the number of times you wish the search string to be replaced and press RETURN, e.g., if you want to replace the next three occurrences of the search string, type 3.
7. If you want to replace all occurrences of the search string between the cursor and the end of text, press RETURN without typing a number.
8. The machine reprints the screen with the cursor immediately following the last replacement string.
9. If you replaced all occurrences of the search string, the machine prints on the message line the number of occurrences found.
10. The machine also prints the number of occurrences if it is less than the number specified.
11. **Note:** You may use MULTIPLE SEARCH AND REPLACE to determine the number of occurrences of a particular string in your text file. To do this, define the replacement string as exactly the same as the search string and search for all occurrences of the string. The machine executes the command the same as always, but since the replacement string is the same as the string it replaces, the text is not changed. When it has found all occurrences of the string, the machine prints the number of occurrences on the message line. You should be very careful when you use the command this way, because a typographical error could be disastrous.

REPEAT SEARCH

1. Pressing the REPEAT SEARCH Control key causes the machine to repeat the last SEARCH or SEARCH AND REPLACE command that you give it.
2. If that was a SEARCH command, the machine performs as described in the SEARCH section.
3. If that was a SEARCH AND REPLACE

command, the machine performs as described in the SEARCH AND REPLACE section.

4. The machine performs only one replacement when you press the REPEAT SEARCH Control Key, even if you had previously executed a MULTIPLE SEARCH AND REPLACE.
5. If you press the REPEAT SEARCH Control Key without having previously defined a search string, the machine searches for the first occurrence of an exclamation point (!).
6. By placing exclamation points in your text file, you may use the REPEAT SEARCH Control Key to help you write standardized letters. (See Lesson Three)

PAGE FEED AND LINE FEED

1. When you press the PAGE FEED Control Key, the machine prints a carat (^) on the screen. This is not actually a carat; it is a control character which is represented by a carat.
2. When the machine encounters this control character while printing the file from EDIT, it moves the printer forward to the first line of the next page.
3. When the machine encounters this control character while printing from PRINT, it treats it exactly like a New Page command (See the notes on PRINT).
4. You may use the PAGE FEED Control Key to format files as INCLUDE files. (See the section on the INCLUDE Command, below).
5. When you press the LINE FEED Control Key, the machine prints a vertical bar (|). Like the FORM FEED, this represents a control character.
6. You do not need the LINE FEED Control Key unless you are using EDIT to compose programs for Microsoft BASIC.
7. If this is the case, you may want to break a program line without terminating the line.
8. To do this, press the LINE FEED Control Key and follow immediately with a carriage return.
9. When you get a listing of the program, the carriage return acts as a carriage return usually acts.
10. However, when the machine executes the program, the LINE FEED control character tells the machine that you do not want to terminate the current program line.
11. Both the PAGE FEED and the LINE FEED

control characters may be deleted with the CHARACTER DELETE Control Key.

OVERFLOWING MEMORY

1. When you get within about 300 characters of filling memory completely, the machine sounds an audible alarm and notifies you on the message line that memory is almost full.
2. To continue editing you need to write some of the file to disk. (See the section on the WRITE command, below)
3. Every time that you move the cursor when memory is this close to being full, the machine repeats the alarm and the message.
4. The machine prevents you from using block manipulation or inclusion to overflow memory; however, if you can tolerate listening to the alarm for at least 250 times, you *can* manually overflow memory.
5. If this happens, you kill the current EDIT and the file in memory.
6. You cannot recover from a memory overflow; all you can do is press RESET and start over.

THE COMMAND SCREEN — GENERAL

1. EDIT begins on the Command Screen.
2. To get to the Text Screen from the Command Screen, press RETURN without entering a command.
3. To return to the Command Screen from the Text Screen, press ESCAPE.
4. To give a command from the Command Screen, press RETURN after typing the command.
5. Commands use uppercase letters; however, the machine treats lowercase letters the same as uppercase letters.
6. If you wish to correct a command that you have typed, you can back up the cursor using either the LEFT, CHARACTER DELETE or BACKSPACE Control Key.
7. To void an entire command line that you have typed, press ESCAPE.
8. The machine only accepts valid commands; if you mistype a command, the machine prompts you with the correct command.

THE COMMAND SCREEN — STATUS FEATURES

1. The Command Screen maintains the status of the file you are editing.
2. On the top lines, it lists the various files you are using and their status.
3. You may have as many as four different files listed on the Command Screen at any one time: a Reading file (the input file), a Writing file (the output file), a Printing file (for background printing) and an Include file (for merging into the file being edited).
4. The Command Screen lists the various FILENAMES and indicates whether they are active, inactive or finished.
5. Next the Command Screen lists the number of lines (carriage returns) in the workspace (i.e., in memory).
6. The Command Screen also keeps track of the number of characters of memory in your system.
7. It lists the total number of characters of available memory in your system (i.e., the amount remaining after CP/M and EDIT have been loaded).
8. Next it gives the current number of characters being used.
9. Finally, it tells you the number of characters in memory that can still be filled without writing some of the file to disk.
10. On the next line, the Command Screen gives you the current mode of the Text Screen — text, program or special. (See section on special formatting commands for a full explanation).
11. Below that is the current line length being used on the Text Screen.
12. Finally, the Command Screen gives you the current locations of the Tab markers.

READ COMMAND

1. When you load EDIT, the machine automatically reads in FILENAME into memory, reserving at least fifty percent of available memory for additions to the text.
2. If the file is larger than fifty percent of available memory, some of it does not load on the first pass.
3. When all of the file has loaded, the machine prints (input finished) next to the input

- file name on the Command Screen.
4. Type **R** (*Read*) if you want to load more of the file.
 5. The machine loads more of the file, up to fifty percent of remaining (twenty-five percent of available) memory.
 6. Each time the machine executes a **READ** command, it has less space left for additions.
 7. If you need more space than would be left after an additional **READ**, or if your text file is larger than available memory, you must use a combination of **READ** and **WRITE** commands. (See the section on the **WRITE** command below.)
 8. You may read a specific number of *logical lines* into memory by typing **Rn**, where *n* is any integer.
 9. After receiving the **Rn** command, the machine reads the file into memory until either the designated number of lines has been read or memory is full.
 10. **Note:** Actually, memory is not completely full. It still has room for about 250 more characters — enough for only very minor additions. (See section on *Overflowing Memory* above).
 11. **Example:** You need to make minor changes to a file that has 40,000 characters. After you have loaded CP/M and **EDIT**, you still have an available memory of 46,000 characters. If you just type **R**, you must repeat the command several times to read the entire file into memory. By typing **R9999** (assuming the file has fewer than 9999 logical lines), you load the entire file with one command.
7. If the cursor is in the middle of a line, the machine writes the file to the end of the preceding line.
 8. **Example:**
 - a. You have a forty page brief that requires extensive editing.
 - b. You load the beginning of the text file when you load **EDIT**.
 - c. The Command Screen indicates that the input of the file is not finished.
 - d. Enter the Text Screen and make corrections to this portion of file.
 - e. Return to the Command Screen and type **W**.
 - f. The machine writes this portion of text to disk and clears memory.
 - g. Type **R**.
 - h. The machine reads the next portion of text into memory.
 - i. Follow steps *d* through *h* until you have made all of your corrections.
 - j. Return to the Command Screen and type **END**.
 - k. The machine writes the text currently in memory to disk, transfers any text remaining in the input file to the output file, and ends **EDIT**.

WRITE COMMAND

1. When you give a **WRITE** command, the machine writes all or a portion of the text in memory to the output file on disk.
2. As the text is written to disk, it is cleared from memory.
3. Type **W** to write all of the file in memory to disk.
4. Type **Wn** to write only the first *n* number of logical lines of the file.
5. **Example:** To write the first ten lines of a text file of thirty lines, type **W10**. The text remaining in memory now begins with line eleven.
6. Type **WC** to write from the beginning of the text to the current cursor position.

LINE LENGTH

1. The **LINE LENGTH** command sets the maximum number of characters per line that may be printed while using **EDIT**, either on the screen or the printer.
2. The **LINE LENGTH** command is for your convenience while using **EDIT**; it has no effect on the file as it is saved on disk or later printed by the **PRINT** program.
3. If you do not give a **LINE LENGTH** command, the machine uses the maximum number of characters for your terminal (usually 80 characters).
4. To give the **LINE LENGTH** command, enter the Command Screen and type **Ln**, where *n* is an integer from two to the maximum for your terminal, e.g., to set a line length of 50 characters, type **L50**.
5. You may change the line length as many times as you wish without affecting the text file.
6. **Note:** When automatic wordsplitting is active, the last character of a line must be a blank, carriage return, page feed or line feed. Since **PRINT** strips off blanks at the

end of a line, a text with a line length of 60 characters may not look exactly the same when it is printed by PRINT as when it is printed by EDIT.

TAB SETTING

1. Tab markers help you move the cursor on the Text Screen and do not otherwise affect the text file.
2. You may designate as many as 16 tab markers.
3. Although you may set markers anywhere up to the maximum line length for your terminal, you may not tab beyond the current line length.
4. The Command Screen indicates at which columns the tab markers are set.
5. Initially these are *1 9 17 25 33 41 49 57 65 73*. (Note: An interval of eight is standard for CP/M.)
6. To change the interval between the markers, type **Tn**, where *n* is an integer between 1 and 80, e.g., **T5** would set markers at columns *1 6 11 16*, etc.
7. To set markers at specific columns, type **Tn, n1, n2, nx, 0**, where *n* through *nx* are integer values. Example, **T5, 20, 33, 50, 0** would set markers at columns *1 5 20 33 50*.
8. The left margin is column one; if you want to set markers five spaces and twenty spaces from the left margin, type **T6, 21, 0**.
9. Note: The machine interprets the final number in a TAB command as an interval rather than a column number. If you had not put a zero in the last position, the previous command would have set markers at columns *1 5 22 43 64*.
10. If you want to set markers at specific columns near the beginning of the line but at intervals thereafter, type **Tn, n1, . . . , nx**, as above, where *nx* is less than or equal to the previous number.
11. Example: To set markers in columns 6 and 21, and then at intervals of ten spaces, type **T6, 21, 10**. The markers are now set at *1 6 21 31 41 51 61 71*.

BLOCK MARKERS

1. Block markers () identify a block of text to be manipulated.
2. Although block markers *look* the same as an underline character, the machine interprets them differently.
3. To manipulate a block of text, you must have *exactly two* block markers in the text.
4. If you have more or less than two markers

in the text when you attempt to manipulate a block of text, the machine prints an error message on the screen.

5. To set block markers, move the cursor to the first character of the block and press the BLOCK MARKER Control Key. Then move the cursor to the first character following the block and press the BLOCK MARKER Control Key again.
6. If the block ends with a carriage return that you wish to include in the block, place the marker at the first character of next line following the carriage return.
7. You may delete individual block markers with the CHARACTER DELETE Control Key.
8. To delete all block markers in the file, enter the Command Screen and type **BK** (Block Kill).
9. Block markers are not saved with the file; when the machine writes a file to disk, it destroys any block markers in the file.

BLOCK COPY

1. With the BLOCK COPY command, you may copy a block of text to another location in the file, leaving two copies of the same block.
2. Mark the block you wish to copy with **BLOCK MARKERS**.
3. Move the cursor to the place you want the copy to go.
4. Note: You may not copy the block to a point within the block. If the cursor is within the block, the machine prints an error message on the screen.
5. Enter the Command Screen and type **BC** (Block Copy).
6. When you return to the Text Screen, the cursor is positioned on the first character of the copy.
7. You may make as many copies of a block as you wish, within the limits of available memory.
8. When you no longer need to mark the original block, delete the markers using the BK command.

BLOCK MOVE

1. With the BLOCK MOVE command, you may move a block of text to another location in the file.
2. Mark the block you wish to move with **BLOCK MARKERS**.
3. Move the cursor to the place you wish to move the block.

4. **Note:** You may not move the block to a point within the block; if the cursor is between the markers, the machine prints as error message on the screen.
5. Enter the Command Screen and type **BM** (Block Move).
6. When you return to the Text Screen, the cursor is positioned on the first character of the moved block.
7. The move destroys the markers, so you do not need to delete them.

BLOCK DELETE

1. With the BLOCK DELETE command, you may delete a block of text from the file.
2. Mark the block you wish to delete with BLOCK MARKERS.
3. Make sure the cursor is not left within the block.
4. Enter the Command Screen and type **BD** (Block Delete).
5. The machine prints the number of characters in the block and requests confirmation that you wish to perform a BLOCK DELETE. (This prevents accidental deletions.)
6. If you wish to delete the block, type **Y**; otherwise, type **N**.
7. When you return to the Text Screen, the block and BLOCK MARKERS have been deleted.
8. You may use the BLOCK DELETE sequence to determine the number of characters in a block of text. Simply type **N** when the machine requests confirmation.

DISPLAY COMMAND

1. With the DISPLAY command, you may examine the contents of any pure Ascii file on disk without exiting EDIT or affecting the file you are editing. (See the glossary for Ascii.)
2. To display a file, enter the Command Screen and type **DFILENAME**.
3. FILENAME must be the exact name of a file on the disk. (See section on Naming Files.)
4. If the disk is not on the current drive, type **D Drive: FILENAME**.
5. Example: You want to display a file named REPLIES.STD which is on the disk mounted on Drive B. Type **DB:REPLIES.STD**. If Drive B is the current drive, type instead **DREPLIES.STD**.
6. If the file is not pure Ascii, the machine prints an error message on the screen.

7. **Note:** If you attempt to display a file that is not pure Ascii, the machine may print some very strange things on the screen.
8. If the file is found *and* it is in pure Ascii, then the machine prints the first screenful of text.
9. When displaying a file, the machine ignores the current line length and prints a line using the maximum line length for your terminal.
10. Automatic wordsplitting does not operate when you display a file, so lines may end in the middle of words.
11. On the message line at the bottom of the screen, the machine prompts with the FILENAME, and a reminder to press RETURN to print the next page of text or ESCAPE to return to the Command Screen.
12. When the machine reaches the end of the file, pressing RETURN returns you to the Command Screen.

INCLUDE COMMAND

1. With the INCLUDE command, you may incorporate all or part of any Ascii file on disk into the text you are editing.
2. **Establishing an INCLUDE file.**
 - a. Type **IFILENAME**. (See section on Naming Files.)
 - b. The machine reprints the Command Screen indicating that FILENAME is an active INCLUDE file.
 - c. Only one file at a time may be an INCLUDE file.
 - d. FILENAME remains an INCLUDE file until you establish another INCLUDE file, or you exit EDIT.
3. **How to include**
 - a. Move the cursor to the place in the current edit file you wish to include the additional text.
 - b. Enter the Command Screen and type **I** to activate the INCLUDE sequence.
 - c. The machine prints the first screenful of text from the INCLUDE file.
 - d. As in the DISPLAY command, the machine does not recognize the current line length or automatic wordsplitting when printing the INCLUDE file.
 - e. On the message line the machine prompts you to press **Y** to include the current screen or RETURN if you do not wish to include it.
 - f. The machine then prompts you to press

- RETURN to continue to the next screen of text, ESCAPE to return to the Command Screen, or type in a section name followed by RETURN (See Below).
- g. The machine repeats this procedure until you press ESCAPE or until it reaches the end of the INCLUDE file, at which point it returns to the Command Screen.
 - h. The amount of text you may include is limited only by the available memory.
 - i. Before a section is included, the machine fits it into memory on a trial basis.
 - j. If it does not fit into memory, the machine notifies you.
 - k. If memory is full, you must write some of your file to disk before continuing the INCLUDE Sequence.
4. **Resetting the INCLUDE file**
 - a. If the portion of the INCLUDE file that you wish to include occurs before the section on the screen, you must reset file to the beginning.
 - b. To do this, enter the Command Screen and type **I@**.
 - c. The machine resets the file to the beginning and is ready for another INCLUDE Sequence.
 5. **Including multiple sections at one time**
 - a. You may include more than one screenful of text in an INCLUDE Sequence.
 - b. The sections are placed in the text file in the order that you include them, with the first section beginning at the cursor and each subsequent section following immediately thereafter.
 - c. If you interrupt the INCLUDE Sequence, e.g., to reset the file, the next section that you include is written at the cursor location *in front of* the text included in the last Include Sequence.
 - d. If you want this section to follow the last section you included, you must enter the Text Screen and move the cursor.
 - e. Example: You have an INCLUDE file consisting of ten sections. You wish to include sections eight and three in that order. After including section eight (as above), reset the INCLUDE file by typing **I@**. Enter the Text Screen and move the cursor so that it follows section eight. Then include section three in the usual way. If you do not first move the cursor, section three is included in front of section eight.
 - f. If you include text in the wrong order, you can easily correct your file using the BLOCK MOVE command.
6. **Formatting files by paragraphs or sections**
 - a. Use the PAGE FEED Control Key to help format INCLUDE files for separation into discreet screens.
 - b. During the INCLUDE Sequence the machine treats a PAGE FEED as a marker for a section, so that each screen, other than the first screen, begins with a PAGE FEED and ends when it encounters the next PAGE FEED.
 - c. If a section is more than one screenful, the machine prints only the first screenful of that section.
 - d. You may put a banner on the first line of each section for identification and reference.
 - e. The banner begins with a PAGE FEED command and ends with a carriage return.
 - f. During the INCLUDE Sequence, you may specify a banner, and the machine searches until it finds a PAGE FEED followed by that banner.
 - g. You do not need to type the entire banner, as long as the portion that you type is sufficient to identify the section.
 - h. You may specify a banner when you begin the INCLUDE Sequence by typing **I@banner**.
 - i. When you do this, the machine goes directly to the section that you specify.
 - j. If you need to reset your file, you must first type **I@** by itself before you type the command to search for a particular banner.
 - k. If you have searched for a particular section by name, the machine does not include the banner when it includes the section.
 - l. Example: You have established a file of standard responses using a system of identifying codes: A1, A2, A3, B1, B2, etc. Your banners look like this:
^A1 — Description of Aluminum Widgets. You receive an inquiry concerning aluminum widgets. After starting an edit of your standard response letter and filling in the variable infor-

mation, you establish your formatted response file as in INCLUDE file. Type **I@A1**. The machine searches through the file until it reaches **^A1** and displays that section of text on the screen. Press **Y** to accept the section. Then press RETURN to reach the Command Screen.

PRINTING FROM EDIT

1. While still in EDIT, you may print a hard copy of all or a portion of the file.
2. The machine prints the text in memory exactly as it would appear on the screen except that PAGE FEED commands are executed and carriage returns are not visible.
3. The machine prints any commands you have embedded for PRINT but does not execute them.
4. Type **P**, to print all of the file in memory.
5. To print a block of text, set BLOCK MARKERS around the block (See section on Block Manipulation); enter the Command Screen and type **PB** (PrintBlock).
6. Press CONTROL S to interrupt printing in progress.
7. Press any key to resume printing.
8. Press ESCAPE to stop printing completely.
9. **Note:** Many printers have a small internal memory from which they print. If your printer is like this, it may take several seconds for it to stop printing since it empties its memory before it responds to the stop command.

PRINTING IN BACKGROUND

1. While you are editing a file, you may, if you wish, print another file *in the background*. (This is also called *spool* printing.)
2. To do this, enter Command Screen and type **SFILENAME** (Spool).
3. If the file is not on the disk or is not pure Ascii, the machine prints an error message on the screen and gives you a chance to change your mind.
4. **Note:** If the background file is a file formatted by PRINT, it may have non-Ascii characters. This does not create a problem. Just press RETURN to continue. Trying to print any other file containing non-Ascii characters may not be possible.
5. The machine rewrites the Command Screen indicating that printing is active for that file.

6. Type **S** to begin print.
7. Enter the Text Screen to continue work on the file you are editing.
8. To halt printing before the end of the file is reached, enter the Command Screen and type **SX**.
9. Type **S** to resume printing at the point you halted.
10. Although you may only have one active printing file at any given time, you may establish more than one file as a background file during a single session.
11. Printing in the background has several limitations that may lessen your ability to use it.
 - a. The machine does not format the file being printed, so it must contain its own line breaks, e.g., a program listing or an output file produced by PRINT. *You cannot print a regular text file in the background.*
 - b. Since the printer and the terminal compete for computer time, the keyboard may be sluggish in responding, and you may find that you drop characters if you try to make a lengthy addition. **Note:** Your dealer may be able to make a minor change to your CP/M which greatly increases the responsiveness of the terminal during background printing.
 - c. You cannot exit from the Editor while background printing is in progress without terminating the printing.

FILES

1. With the FILES command, you may examine the directory of a disk without exiting EDIT.
2. To give the FILES command, enter the Command Screen and type **F DRIVE** (Files).
3. If you indicate a valid drive, the machine prints out a list of the files on that disk.
4. Press RETURN to return to the Command Screen.
5. If no disk is mounted on the indicated drive, the machine hangs up until you put a disk on that drive.
6. Example: You wish to include a section from a file containing stock responses, and you need to find out if the file is on the disk mounted on DRIVE B. Enter the Command Screen and type **FB**. If DRIVE B is the current drive, type **F**. The

machine prints the directory but the file you need is not there. Replace the disk with the correct disk, making sure that it is correct by using the FILES command again. Proceed with the INCLUDE Sequence.

SPECIAL FORMATTING FEATURES

1. EDIT has three modes of formatting the screen: Text, Program and Special. **Note:** If you plan to use EDIT solely for the creation of text files, you probably do not need the information in this section.
2. The differences between the modes are a few formatting features: Automatic truncation of lines at the end of words (wordsplitting), blank packing and the display of the carriage return symbol (~) on the screen.
3. In text mode, all three features are active.
4. In program mode, automatic wordsplitting and the visible carriage return are inactive.
5. Any other combination of features is considered special, e.g., automatic wordsplitting inactive but the carriage return displayed.
6. EDIT begins in text mode unless the *File Type* of the output file is ASC, ASM, BAS, COB, FOR, MAC or PRN, in which case it begins in program mode.
7. To change to program mode, enter the Command Screen and type **MP (Mode Program)**.
8. To change to text mode, enter the Command Screen and type **MT (Mode Text)**.
9. When automatic wordsplitting is inactive, the machine only breaks a line when it exceeds the specified line length or encounters a carriage return.
10. To inactivate automatic line truncation (wordsplitting), enter the Command Screen and type **MAN (Mode Automatic truncation, No)**. To activate this function, type **MAY**.
11. The blank packing feature takes a series of blanks and replaces them with a tab symbol if they cross a CPM tab point (1,9,17,etc.). This may save considerable disk space.
12. You should only need to inactivate this feature if you are writing programs for Microsoft BASIC which has different tab points or when you are creating a fixed length data file.

13. To inactivate blank packing, enter the Command Screen and type **MBN (Mode Blank packing, No)**. To reactivate, type **MBY**.
14. To inactivate the visible carriage return, enter the Command Screen and type **MCN (Mode Carriage return, No)**. To activate this feature, type **MCY**.

EXITING THE EDITOR

1. Normally you exit EDIT by entering the Command Screen and typing **END**.
2. The machine saves the edited file under the name you specified when you loaded EDIT.
3. If you have not saved the edited file under a new name or on a different disk, the original file is renamed as a BAKfile type.
4. If there is an existing BAK file when this happens, it is erased.
5. **Note:** The machine does not distinguish between BAK files for different files with the same *File Name*, e.g., REPLIES.PRO, REPLIES.CON. Consequently, if REPLIES.PRO had a BAK file when you edited REPLIES.CON, it would be erased and replaced with the new REPLIES.BAK.
6. To exit EDIT without saving the edited file, enter the Command Screen and type **QUIT**.
7. The machine ends EDIT without renaming the original file or its backup, if any.
8. Finally, you may wish to save the current file as edited without changing the original file or its backup, e.g., you did not specify a new name for the output file when you initialized EDIT but wish to do so.
9. To save the edited file with a \$\$\$ File Type (the designation of an output file), enter the Command Screen and type **W** (to write the edited file to disk). When the machine has completed that command, type **QUITX**.
10. **Note:** The **QUITX** command leaves the output (\$\$\$) file exactly as it is when you give the command. Only that portion of the file that has been previously written to disk, is saved.

RECOVERING FROM A DISK FULL ERROR

1. If the machine runs out of disk space while executing a WRITE or END command, it prints **DISK IS FULL** on the screen.
2. To save the remainder of your edited file, you need to make room for it on the disk by

erasing a file or files that you no longer need.

3. Type **KFILENAME** (Kill), to erase a file from the disk.
4. If machine does not find the specified file, it informs you; otherwise, it requests confirmation that you wish to kill **FILENAME**.
5. If you respond **Y**, then the **KILL** command is carried out. If you respond **N**, the **KILL** command is aborted.
6. After you have made room for the file that you are saving, resume by giving another **WRITE** or **END** command.
7. Example: You are editing a very long document by using a series of **READ** and **WRITE** commands. This means that you now have at least three files on the disk — the origi-

nal and its backup, both of which are very large, and the new output (\$\$\$) file which is getting larger each time you write to it. As the machine executes another **WRITE** command, it runs out of disk space. You have two choices. You can keep the disk as it is by typing **QUITX**, exiting **EDIT** and preserving your current output file. This way you preserve your **BAK** file, but you lose any changes that have not yet been saved. This is no good, so you decide to make room on the disk. First, you type **F** to find out which files you have on the disk. You decide to kill the **BAK** file. You give the **KILL** command. Afterwards, you resume the **WRITE** command that was in progress at the time the disk became full.

NOTES ON PRINT

LOADING PRINT PROGRAM/TEXT FILE

1. The complete format for loading the PRINT program and text file is: *(Drive:)*PRINT *(Drive:)*FILENAME. (See Section on Naming of Files, above)
2. You must indicate the drive if the disk is not on the current drive. (See Glossary under **current drive**.)
3. After printing the sign-on screen, the machine pauses so that you may change disks and/or enter a command prior to processing the file.
4. When you press RETURN, the machine resets the disk drives so that you can write to any disk. (See section on Writing to Disk, below)
5. After resetting the disks, the machine searches for the file you have indicated.
6. If the file that you indicated is not on the indicated disk (or if you did not previously enter a name) the machine prints a message on the screen and allows you to enter a different FILENAME.
7. You may change disks before entering the new names if you wish, since the machine resets the disks as soon as you press RETURN.
8. After the machine locates the file on the disk, it looks to see if you entered a command prior to pressing RETURN.
9. If you did enter a command, the machine executes the command, prints a back slash (\) on the screen and awaits another command before processing the file.
10. If you did not give a command, the machine begins to process the file without pausing.

GIVING COMMANDS
FROM THE KEYBOARD

1. You may enter almost any print command directly from the keyboard at run time. (Those that may not be given from the keyboard are indicated as they occur.)
2. The machine allows you to enter commands from the keyboard under four different circumstances:
 - a. when you first load PRINT (see preceding section);
 - b. when the machine encounters a TEXT command (See section on the TEXT command, below);
 - c. when you interrupt the processing of a file (either manually or by embedding a WAIT command);
 - d. when the machine encounters an improper command.
3. To interrupt the processing of the file manually, press any key on the keyboard (except the CONTROL or SHIFT key).
4. As soon as that the machine goes to the file to get a character, it recognizes that you wish to interrupt.
5. When this happens, the machine prints a back slash on the screen, halts the processing of the file, and waits for you to enter a command.
6. To enter a command from the keyboard, type in the command and press RETURN.
7. **Note:** Although commands are comprised of only uppercase letters, the machine converts any lowercase letters to uppercase before processing the command.
8. The machine processes commands in the order that you enter them.
9. If the machine accepts the command, it prints another back slash on the screen (indicating that it is ready for another command).
10. If the machine rejects the command for any reason, it prints an error message on the screen prior to printing the back slash.
11. **Note:** Whenever you give an incorrect command, whether from the keyboard or the text, the machine prints an error message which explains why it was rejected.
12. The machine ignores any command following an incorrect command on the same command line.
13. You must reenter the commands that follow an incorrect command as well as correcting the error.
14. You may give as many commands as you wish at one time.
15. You may separate commands with commas, and place more than one command on a command line, e.g., LM10, RM60, LEFT, SP+1.
16. **Note:** When you enter commands from the keyboard, the command line may not be longer than the width on the screen (usually 80 characters).
17. When you have given all of the commands that you want to give, press RETURN without entering any command.
18. The machine resumes processing the file where it left off.
19. You may interrupt the machine as many times as you wish during the processing of the file.
20. **Note:** Entering a formatting command from the keyboard overrides the existing

parameters; however, an embedded command can change the parameters back again if it occurs after your runtime command. The only way to change a command permanently is to do so in the text.

EMBEDDING COMMANDS IN A TEXT FILE

1. You may embed any print command directly in the text file.
2. Indicate a command by typing a back slash (the command marker).
3. Embedded commands have exactly the same format as those that you enter from the keyboard.
4. You may use upper or lower case, as you prefer, because the machine interprets all letters as uppercase.
5. **Note:** You may find that your file is more readable if your commands contrast with the main body of the text, i.e., if most of your text is in lowercase, uppercase commands are more likely to stand out.
6. You may include more than one command in a command line by separating the commands with commas.
7. A single command line may contain no more than 256 characters. (**Note:** this is considerably more than commands given from the keyboard.)
8. When you have finished a command line, press RETURN.
9. If you wish to include a command as part of a text line, you may end the command by typing a second back slash instead of the carriage return.
10. **Note:** *There is no difference as far as the machine is concerned whether you end a command line with a carriage return or a second back slash; you should base your decision on how to end a command line by what makes sense as far as the layout of the text is concerned.*
11. The following examples process and print exactly the same:

```
\LM10      \LM10,RM60\Lincoln's ...
\RM60
Lincoln's ...
```
12. **Note:** If you end a command line with a back slash followed by a carriage return, the carriage return becomes part of the text and is executed by the printer.
13. Some commands must appear on separate command lines or at the end of a command line.
14. The sections on specific commands indicate when this is the case.

PROCESSING EMBEDDED COMMANDS

1. When the machine encounters a back slash in the text file, it interprets anything that follows it as a command until it encounters another back slash or a carriage return, whichever comes first.
2. Embedded commands are completely invisible to the printer, i.e., although the machine processes each command as it encounters it, the command itself is not printed and occupies no space on the printed page.
3. For example: If you were using the KERN command (see below) to tighten up the printing of a particular word, you might have something that looked like this:

```
\K-2\W\K-3\A\K-3\IT
```

but it would print as **WAIT**.
4. The machine processes commands as it receives them, so if you have commands that must be executed in a specific order (e.g., **IN5,PI-5**), you must be sure to give them in the correct order. *
5. If the machine encounters an improper command, it interrupts the processing of the file, prints an error message on the screen and gives you the opportunity to enter the correct command from the keyboard.
6. Enter corrections as you would enter any commands from the keyboard.
7. After you have made the corrections and given the command to resume, the machine continues as if nothing had happened.
8. **Note:** When the machine interrupts the text with an error message, you may give any command that you wish. You are not limited to merely correcting the error.

WHAT IS A SPECIALTY PRINTER

1. The main difference between a standard printer and a specialty printer is how they move.
2. A standard printer can only move horizontally by whole spaces and vertically by whole lines.
3. A specialty printer moves both horizontally and vertically by *increments*, usually 120 increments to the horizontal inch and 48 increments to the vertical inch.
4. Consequently, if you were printing ten characters per inch on a specialty printer, the machine would be moving twelve *increments* per character. If you were printing twelve characters per inch, the machine would move ten *increments* per character, etc.
5. You have at your disposal a number of commands that utilize incremental movement

and which, consequently, may only be performed on a specialty printer.

6. **Note:** If you have a file that was written for a specialty printer, you may print it on a standard printer if you give a **DRAFT** command. (See section on the **DRAFT** command, below)

TEXT POSITIONING COMMANDS

LM — Left Margin

1. The LM command establishes the number of characters that the machine offsets the printed line from the extreme left of your printer.
2. The format for the LM command is **LM n** , where n is an integer between zero and the maximum page width for your printer. For example, **LM10** offsets the line ten characters.
3. The LM command affects only the position of the line on the page; it does not affect the length of the line.
4. The machine positions the line on the page by inserting blanks at the beginning of the line, e.g., if you type **LM10**, the machine inserts ten blanks.
5. It may take noticeably longer to print a page with a large left margin, because the machine must output a blank for each character of the offset.
6. The LM command establishes the minimum left margin; if you wish to establish a temporary left margin to the right of the minimum left margin, use the **IN** command. (See below)
7. If you type **LM $+n$** or **LM $-n$** , the machine increases or decreases the size of the offset by n characters.
8. For example, if you have a left margin offset of ten characters and you type **LM+5**, your offset is now fifteen characters. (**Note:** You could also have typed **LM15**.) Typing **LM-5** (or **LM10**) changes the offset back to ten characters.
- 9.*The size of the offset is dependent upon the pitch in effect when you give the LM command, e.g., **LM10** in ten pitch is one inch whereas **LM10** in twelve pitch is 5/6 of an inch. (See sections on Movement and Spacing below)

RM — Right Margin (Default-60)

1. The RM command establishes the number of characters in a line, and therefore, its length.
2. The format for the RM command is **RM n** where n is a whole number, e.g., **RM60**

establishes a maximum line length of sixty characters from the left margin.

3. You may not set a line length wider than the maximum width of your printer less the offset for the left margin, e.g., if your printer has a maximum width of eighty characters and you establish a left margin of ten characters, the maximum number for the **RM** command is seventy.
4. You may not set a line less than the indentation (**IN**), e.g., if you have established a temporary left margin of twenty characters (**IN20**), you may not change the line length to less than 20 characters.
5. You may increase or decrease the line length by typing the new length or by adding or subtracting characters from the current length, e.g., to change a 60 character line length to 55 characters, you may type **RM55** or **RM-5**.
- 6.*The actual length of the line depends upon the pitch in effect when the command is given. For example, if you establish a line length of sixty characters (**RM60**) when the machine is set at ten pitch, the line is six inches long. If you then change to twelve pitch, the line is still six inches long, but now it holds 72 characters.

IN — Indentation

1. The **IN** command establishes a temporary left margin, an offset from the offset.
2. The format for the **IN** command is **IN n** , where n is an integer between zero and the current line length, e.g., if **RM** is 60, the maximum possible indentation is 60.
3. As with the **LM** and **RM** commands, you may change **IN** by typing in the new indentation or the change from the previous indentation. (**IN $+n$** or **IN $-n$**)
4. **IN0** returns the left margin to that established by the **LM** command.

PI — Paragraph Indentation

1. The **PI** command establishes the indentation that immediately follows a carriage return, i.e., the beginning of a logical line or paragraph.
2. The indentation may be either positive or negative.
3. A paragraph that has negative indentation is called a *hanging paragraph*.
4. The format for the **PI** command is **PI n** , for a positive indentation, and **PI $-n$** , for a negative indentation
5. Positive indentation is limited by the current line length less any temporary inden

*for specialty printers only

- tation; negative indentation can be no larger than the current temporary indentation.
- For example, if IN is five characters, PI-5 is the maximum negative indentation permissible.
 - Since the PI command may have a negative value, you must type in the actual indentation, e.g., the only way to change the paragraph indentation from -5 to -10 is to type **PI-10**.
 - If you have manually indented your text file, the machine adds (or subtracts) the paragraph indentation to the indentation in the file.
 - The machine only recognizes paragraph indentation if you are printing the line in flush left or justified format. (See below)

PL — Page Length (Default-66)

- The PL command establishes the number of single-spaced lines per printed page.
- The format for the PL command is **PLn**, where *n* is any integer between zero and 255.
- The page length must be greater than or equal to the top margin plus the bottom margin.
- You may increase or decrease the page length by typing **PL+n** or **PL-n**.
- If you set a page length other than your usual page length, you may need to turn off the automatic form feed feature. (See Section of **FORMFEED OFF** command below)
- *The length of the page depends upon the number of lines per inch when the PL command is given, e.g., PL66 at six lines per inch is eleven inches.

TM — Top Margin

- The TM command establishes the number of lines that the machine reserves at the top of the page as a margin before printing the body of the text.
- The format for the TM command is **TMn**, where *n* is any integer from zero to current page length minus the current bottom margin.
- You may increase or decrease the top margin by typing **TM+n** or **TM-n**.
- Use the TM command to give the proper spacing between your heading (if any) and the body of text.

BM — Bottom Margin (Default-6)

- The BM command establishes the number of lines from the bottom of the page that the

- machine stops printing the body of the text.
- The format for the BM command is **BMn**, where *n* is any number between zero and the current page length minus the current top margin.
 - You may increase or decrease the bottom margin by typing **BM+n** or **BM-n**.
 - Note:** The bottom margin is usually larger than the top margin because the bottom margin often contains the first few lines at the top of the physical page.

FORMATTING COMMANDS

LEFT — Flush Left

- Flush left is the standard typewriter format; the left margin is flush and the right margin is uneven.
- Flush left is the default format for PRINT.
- To change to flush left format, type **LEFT**.
- In flush left format, the machine only ends a line with a complete word or with a hyphen. (See section on Ghost Hyphens, below)
- The machine strips off blanks at the end of the line so that you may fit a full sixty characters on a sixty character line. (As opposed to the way EDIT breaks lines. See the section on **LINE LENGTH** in the notes on EDIT)
- If a single word is longer than the current maximum line length, the machine breaks the word in the middle, i.e., if you try to print a 20-character word on a 15-character line, the machine prints the first 15 characters on the line and the last 5 characters on the next line.

RIGHT — Flush Right

- Flush right is the reverse of flush left, i.e., the right margin is flush and the left margin is uneven.
- To change to flush right, type **RIGHT**.
- When the machine is printing flush right, it inserts at the beginning of the line the number of blanks necessary to move the line to the right margin.
- The most frequent use for flush right format is heading and footings, e.g., printing the date or page number.

JUST — Justified

- A text is *justified* when both the left and right margins are flush.
- The machine creates even margins by inserting blanks into the text line to pad out the line to the current maximum line length (RM minus IN minus PI).

*for specialty printers only

3. On a standard printer the machine does this by inserting character-sized blanks between the words on the line.
4. Since inserting large spaces into a line may create *holes* in the text, the machine alternates between inserting spaces on the left, right and middle parts of a text line.
5. The machine distributes the spaces as evenly as possible on the line.
6. If a line is less than the maximum current line length because it ends with a carriage return, the machine prints that line flush left without inserting any spaces.
- 7.*If you type **JUST** on a specialty printer, the machine divides the additional space evenly among all of the spaces between the words, e.g., if you have five spaces to insert into a line that has eleven words in it (*10 spaces between the words*), the machine inserts an additional half space between every word.
8. **Note:** The fewer spaces that you need to insert into a justified line of text the better. To give your justified text a more even appearance, you need to use a certain amount of hyphenation. (See section on Ghost Hyphens, below)
9. **Note:** Although justified text looks processed and very nice, recent research has indicated that justified text is harder to read than text printed flush left. For brief documents like letters it does not make much difference; but with longer documents, these studies show that the uneven right column helps the reader keep track of his or her place in the text.

*JUSTC — Justified

1. If you type **JUSTC** instead of **JUST**, the machine inserts very small spaces between the characters themselves (we call this character spreading).
2. This gives a much more even appearance to the text; however, it may not be the easiest format to read.
3. **Note:** Spreading out the letters in a word adds a subtle emphasis to that word. If you use character spreading, you may be emphasizing words unintentionally. You need to decide what looks best to you.

CENTER — Centered Format

1. **CENTER** format prints each text line in the center of the current maximum line length (LM less IN).
2. Typing **CENTER**, causes all subsequent lines to be centered until you change the format.

3. The machine does not break lines at the ends of words while in **CENTER** format.
4. For the machine to center a line, it must be shorter than the current maximum line length.
5. The machine prints any line that is longer than the current maximum line length in literal format (see below).

CTR — Centered Line

1. To center a single line, type **CTR** at the beginning of that line (or on the command line immediately preceding it).
2. The machine prints the next line in center format.
3. The machine prints subsequent lines in the format in effect when you gave the **CTR** command, e.g., if you were in flush left format and you gave the command to center a line, as soon as the machine had centered that line, it would return to flush left format.
4. All of the rules that apply to **CENTER** format apply to the **CTR** command as well.

LIT — Literal

When in literal format, the machine prints the text exactly as it finds it, breaking a line only when the maximum line length has been exceeded.

2. While in literal format, the machine executes all of the shaping commands except the **PI** command.
3. Normally, program listings need to be printed in literal mode.
4. **Note:** If you plan to use **PRINT** for program listing, be sure to inactivate the various recognition characters (see below) before beginning the listing. You may give these commands from the keyboard or from commands embedded in **REM** statements or their equivalent.

MOVEMENT AND SPACING

*CPI — Character Per Inch (Default-10)

1. To change the characters per inch (*pitch*), type **CPI n** , where n is a whole number greater than or equal to one.
2. The machine calculates to the nearest whole increment the number of increments necessary to print at that pitch.
3. Fractional increments are rounded down.
4. Example: Your printer moves 120 increments per inch and you type **CPI9**. Nine characters per inch would be $13 \frac{1}{3}$ increment per character which would round

*for specialty printers only

quently, you would actually be typing 9 3/13 characters per inch.

5. When you give a positioning command, the current pitch determines how the machine interprets the command, e.g., an LM10 command sets a left margin of one inch at CPI10 and 5/6 inch at CPI12.
6. **Note:** If you are varying your pitch, the order in which you give commands is critical, e.g., LM10, RM60, CPI12 is *not* the same as CPI12, LM10, RM60.
7. Once a positioning command has been executed, changing the pitch does not affect it, e.g., a line length of six inches at ten pitch remains six inches when you change to twelve pitch.
8. When you change the pitch, the machine recalculates the number of characters needed to carry out a positioning command.
9. You may increase or decrease the number of characters per inch by typing **CPI+n** or **CPI-n**.
10. **Note:** You may change the pitch anywhere in the text, in the middle of a line, even in the middle of a word.

*H — Horizontal Movement by Increments

1. When you give a CPI command, the machine calculates the number of increments per character.
2. You may instead directly specify the number increments per character by typing **Hn**, where *n* is a number between zero and the number of increments per inch for your printer.
3. This command gives you greater control over the horizontal movement of your printer, and is the only way to print some sizes. For example, if your printer has 120 increments per character and CPI12 has 10 increments per character. However, CPI11 does not print 11 increments per inch because the machine rounds the number of increments per character to 10, the same as twelve pitch. The only way to print 11 increments per inch is by typing **H11**.
4. You may increase or decrease the number of characters per inch by typing **H+n** or **H-n**.
5. If you type **H0**, you inactivate any forward movement by the printer.

*K — Kerning

1. Kerning is a typographical term referring to the way you reduce space between some letters to make them fit together better.

2. With the **K** command, you may increase or decrease the horizontal movement for the character immediately following the command, e.g., **K3** increases by 3 the number of increments the printer moves when it prints the next character; **K-3** decrease the number of increments by 3.
3. You may use the **K** command to overstrike characters. For example, if you wanted to accent the phrase *maitre d'hotel* that you were printing at ten pitch (twelve increments per character), you would type `ma\k-12\i ^ tre d'h\k-12\o ^ tel` which would then be printed as *mâitre d'hôtel*.
4. **Note:** The printer strikes the character and then moves. Consequently, if you want to affect how much the printer moves after striking a character, you must give the command in front of the character.

*LPI — Lines Per Inch (Default-6)

1. The LPI command sets the number of single-spaced lines that your printer allows in a vertical inch.
2. The format for the LPI command is **LPI*n***, where *n* is an integer greater than zero and less than or equal to the number of increments per vertical inch allowed on your printer (usually 48).
3. **LPI0** is not a legitimate command; if you wish to inactivate the line feed, use either **SP0** or **V0** (see below).
4. The LPI command affects the PL, TM and BM commands in the same way that CPI command affects LM, RM, IN and PI, i.e., the lines per inch in effect when you give a vertical length command determines how the machine interprets the command.
5. **Note:** Do not confuse the lines per inch with the spacing (SP) command. The SP command does not affect the number of lines on a page; the LPI command does.

SP — Spacing

1. The SP command sets the spacing of the printed line.
2. The format for the SP command is **SP*n***, where *n* in an integer between zero and six, e.g., **SP2** sets the machine for double spacing.
3. **SP0** inactivates your printer's automatic line feed, allowing you to print special formats. For example, if you wanted to print a heading for a letter with the customer's name flush left and the date flush right, you would give the following commands:
`\SP0,LEFT\American Widget Co.
\SP1,RIGHT\October 23, 1927`

4. Note: You must restore or change spacing *prior* to a carriage return if you want to affect the spacing at the end of that line.
5. *To set your machine for half spacing, type $SP+n$, e.g., $SP+1$ would be one-and-a-half spacing, $SP+2$ would be two-and-a-half spacing, etc.

*V — Vertical Movement by Increments

1. The **V** command is the equivalent of the **H** command for vertical movement.
2. The **V** command allows you to specify the exact number of increments per vertical line that you want the machine to move.
3. **V0** inactivates the line feed (similar to **SP0**).
4. The format for the **V** command is Vn , where n is a number from zero to the number of increments per vertical inch allowed by your printer.
5. You may increase or decrease the increments by typing $V+n$ or $V-n$.

*PROPORTIONAL PRINTING

1. Some specialty printers have proportional print wheels available.
2. The characters on a proportional print wheel are designed to take up varying amounts of room on the page, e.g., an uppercase "M" takes up a lot more room than a lowercase "i".
3. When printing proportionally, the machine must calculate the number of increments to move for each character that it prints.
4. Proportional print wheels are not arranged the same as non-proportional wheels, because it is necessary to alternate wide and narrow characters.
5. Consequently, if you try to type with a proportional wheel on a non-proportional printer, none of the characters will be in the right spot.
6. Check with the dealer from whom you got the **MAGIC WAND** to see if your specialty printer can be used for proportional printing using **PRINT**.
7. If it can, we strongly recommend that you purchase a proportional print wheel so that you can take advantage of this feature of **PRINT**.
8. Your dealer can supply you with the necessary information on how to use **PRINT** for proportional printing.
9. Using a proportional print wheel is almost as easy as using a non-proportional wheel, and it looks much, much better.

*for specialty printers only

RECOGNITION CHARACTERS

1. A *recognition character* is a character in the text that triggers an action by the machine, i.e., the machine *recognizes* it but does not print it.
2. A recognition character only functions as part of the text. It does not act as a recognition character when it is part of a command line or the value of a variable.
3. You may have as many as seven recognition characters at any one time: the command marker, ghost hyphen, underscore, bold face, sub/superscripting, and **IGNORE** characters.
4. You may define any character as a recognition character.
5. You may define only one recognition character for a given function; as soon as you define a new character for that function, the previous character is undefined, i.e., it is no longer a recognition character.
6. A character may be the recognition character for only one function at a time.
7. For example, if you define an at-sign (@) as the recognition character for underscoring, not only do you undefine the current underscore character, but you undefine the at-sign for bold face as well.
8. **PRINT** has default values for all of the recognition characters except the **IGNORE** character.
9. You should not change these characters unless you need to print one of them.
10. Note: This recommendation is for your convenience only, so that you do not have to keep track of the current recognition characters.

CMD — The Command Marker (Default \)

1. Typing $CMDc$, where c is any character, changes the command marker.
2. The command may be given either in the text or from the keyboard.
3. If given in the text, you must first mark the command with the current command marker.
4. If you want to end the command line with a command marker instead of a carriage return, you must use the old command marker rather than the new one. (This is because the command marker is not changed until the command has been executed, and machine does not execute the command until the command line is concluded.)
5. Example: If you wish to change from a back

slash to a open bracket, you would type `\CMD[\`.

6. Typing `CMD` by itself means that the machine does not recognize any character as a command marker. This means that you may not embed any commands in the text, but you may still interrupt to give commands from the keyboard.
7. **Note:** The machine prints the current command marker on the screen when it is ready for you to give a command from the keyboard. If you have undefined the command marker, nothing is printed on the screen, but you may still give commands as usual.
8. **Note:** If you *must* change the command marker, you should probably change it at the beginning of the text and keep it the same throughout to avoid confusion.
9. In a multi-pass file changing the command marker creates confusion, because on the second pass the machine does not recognize the original command that changed the command marker (because `\` is no longer the command marker).
10. If you change the command marker with an embedded command in a multi-pass file, you must change the marker back to a back slash at the end of the file.
11. As an alternative, you may change the command marker from the keyboard before you begin processing the file. This way the command to change the command marker is not part of the file.

HY — Ghost Hyphen (Default &)

1. The default Ghost Hyphen is the ampersand (&).
2. To define a new Ghost Hyphen type `HYc`, where *c* is any character.
3. Put the Ghost Hyphen recognition character in a place where you would allow a word to be hyphenated if it could not otherwise fit on a line.
4. When a word does not fit on a line, the machine looks to see if the word contains a Ghost Hyphen.
5. If the word does contain a Ghost Hyphen, the machine checks to see if the word fragment marked by the recognition character will fit on the line.
6. If the fragment fits on the line, the machine prints the fragment, replacing the Ghost Hyphen with a true hyphen, and moving the remainder of the word to the beginning of the next line.
7. If the word contains more than one Ghost Hyphen, the machine prints the largest

fragment that will fit on the line.

8. The Ghost Hyphen is never printed on the screen; if a word does not need to be hyphenated, the machine passes over the recognition character; if a word needs to be hyphenated, the machine replaces the recognition character with a true hyphen.
9. The machine breaks words only at Ghost Hyphens; if you have a hyphenated word which you would allow to be broken if necessary, you must first insert Ghost Hyphens, e.g., *mother-in-law* would become *mother-&in-&law*.
10. If a Ghost Hyphen follows a true hyphen, the machine prints only one hyphen.
11. Since a Ghost Hyphen is ignored except during hyphenation decisions, you may use the Ghost Hyphen recognition character as a second IGNORE character as long as you place it at the end of and between words. (See the section on the IGNORE character below).

UN — Underscoring (Default _ Solid)

1. You mark characters for underscoring with the underscore recognition character.
2. You define an underscore recognition character by typing `UNc`, where *c* is any character.
3. The default underscore recognition character is the underline character itself.
4. Place a recognition character immediately in front of the first character and immediately following the last character you wish to underscore.
5. If the characters you wish to underscore are immediately followed by a carriage return, you do not need to type in the second recognition character.
6. After printing a line, the machine goes back and underscores the characters you have marked for underscoring.
7. You may underscore any character at any point in the text automatically, no matter what else you might be doing, e.g., boldface, sub/superscripting, hyphenating, centering, justifying, etc.
8. If you have set the pitch at greater than CPI10, the underscoring may not be completely joined together.
9. You may choose either *solid* or *broken* underscoring.
10. *Solid* underscores every character between the two markers; *broken* underscores letters and numbers, but not blanks or punctuation marks.
11. To specify solid underscoring, type `UNS`; for

broken underscoring type UNB.

12. To print a solid line you may either undefine the underscore character and type in the line as you would on a typewriter, or you may place two underscore markers around the appropriate number of blanks, making sure that the machine is set for solid underscoring.
(\UN_____\UN_ or \UNS_____\UNB\)

*BF — Bold Face (Default @ Intensity 1)

1. You mark characters for bold face with the bold face recognition character, which you define by typing BFc.
2. The default recognition character for bold face is the at-sign (@).
3. Mark the text to be bold faced the same way you would mark it to be underscored (See above).
4. The machine boldfaces a character by striking it once, moving the print head one horizontal increment (usually 1/120 of an inch) and striking the character again, after which it moves the regular movement for that character.
5. You may set the machine to strike from one to nine additional times by typing BF n , where n is a number from one to nine.
6. You cannot set the intensity higher than nine; if you attempt to do so, the machine interprets one of the characters as the recognition character, e.g., if you type BF10, the machine sets the intensity at 1 and the bold face recognition character as 0.
7. The higher the intensity number, the more times you strike a character, the darker it becomes, the more space it occupies and the longer it takes to print.
8. **Note:** You can fit fewer bold faced characters on a line than non-boldfaced because of the extra increments that the printer moves for the additional strikes.
9. Since typing in bold face may be time consuming, you may wish to turn off bold face when printing rough drafts.
10. To turn off bold face, type BF0.

*SSA,SSB — Super/Subscripting (Default <, >)

1. To change the superscripting recognition character, type SSA c ; to change the subscripting recognition character, type SSB c .
2. Super/subscripting moves the carriage up or down one half line.
3. The movement is based on the current LPI.
4. The movement caused by the super/subscripting recognition characters is permanent, i.e., it stays in effect until you change it with the opposite command.

5. Superscripting is the only way you can back up on the page.
6. If your paper is not held firmly in place on the printer, it may slip when attempting to superscript.
7. Attempting to superscript when the number of vertical increments per line is large may also cause the paper to slip.

IGNORE

1. By typing IGNORE c , you make c a character that the machine passes over without printing, in fact without doing anything.
2. The IGNORE character was designed to be used with the SKIP TO command, to allow you to skip to the middle of a line of text or to the beginning of a command line.
3. You may also use the IGNORE command to inactivate other recognition characters.
4. Example: You have a text with a number of embedded underscore commands that you wish the machine to pass over. If you typed UN, you would undefine the recognition character which *would* inactivate underscoring, but the machine would now print the previous recognition character. The solution is to set the previous recognition character as an IGNORE character. Now when the machine encounters it, it passes over it as if it were not there.

OUT

1. The machine drives your printer by sending it a series of numbers which tell it what character to print, how much movement to make, etc.
2. These numbers are generated by the machine based on the Ascii value of the text characters and certain protocols of your printer.
3. The OUT command allows you to output directly to your printer a number or series of numbers that you yourself generate.
4. This means that you may be able to take advantage of special features on your printer that are not supported by PRINT.
5. The format for the OUT command is OUT $n1, n2, \dots, nx$, where numbers are between zero and 255.
6. The machine only checks to see that n is an acceptable number; it is up to you to see that it is meaningful.
7. **Note:** The machine transmits the numbers to the printer as soon as it encounters the command. Unlike recognition characters, it

does not wait until the text characters are transmitted to the printer.

8. **Note:** This command is designed to be used primarily by programmers. Do not attempt to use it unless you are completely familiar with the workings of your printer.

MULTI-PAGE AND MULTI-PASS COMMANDS

DRAFT

1. The DRAFT command allows you to turn off or ignore any command associated with specialty printers (boldface, variable pitch, super/subscripting, etc.)
2. This allows you to print on a faster standard printer a file prepared for a specialty printer.
3. **Note:** Attempting to print such a file on a standard printer without first giving the DRAFT command, gives undesirable results.
4. You must give the DRAFT command before any of the file has been printed.
5. The command may be embedded in the file, or you may give it from the keyboard at run time.
6. You may not change from DRAFT mode after you have given the command.

FORM — (Default Continuous)

1. Any multi-page or multi-pass file requires more than one page of paper.
2. If you are not using continuous form paper, you must pause between pages to insert a new sheet of paper.
3. The command that tells the machine to pause is FORMS (FORM Single).
4. Once you have given this command, the machine pauses after each page to give you the opportunity to load in a new sheet of paper.
5. When you have loaded the printer, press RETURN to resume printing.
6. If you wish to terminate PRINT at the end of a page, press ESCAPE instead of RETURN.
7. To change back to continuous form paper, type FORMC (FORM Continuous).

LINE

1. The LINE command allows you to move to a particular line on a page.
2. The format for the line command is LINE n , where n is a number between the current line and the current maximum line number.
3. The machine ignores any command to move to a line less than the current line or greater than the current maximum.
4. The LINE command is the equivalent for

vertical movement of the TAB command (See below).

5. It allows you to move to a specific line number, independent of the number of lines you have already printed. For example, if you wanted a letter to begin on line 10 but you did not know ahead of time how many lines would be in the address, you would type LINE10 following the address lines.
6. The LINE command also allows you to move to a particular line independent of the spacing you are using, e.g., if you were double-spacing a document starting on line one, you would normally be unable to get to line ten. By typing LINE10 you would be able to skip to that line.

TAB

1. The TAB command allows you to skip to a particular column.
2. The format for the TAB command is TAB n , where n is a number between the current and maximum column number.
3. If you attempt to tab to a column to the left of the current column, the machine ignores the command.
4. If you attempt to tab to a column to the right of the current maximum, the machine refuses the command and prints an error message on the screen.
5. The TAB command is the only way that you can line up even columns when you are bold facing or using a proportional print wheel.
6. To print character while tabbing to a column, type TAB" c " n , where c is any character.
7. For example, to print periods while tabbing to column 40, type TAB"."40.

NP — New Page

1. The NP command forces a new page.
2. If you have any headings or footings in the file, the machine executes them following the NP command.
3. The machine treats a FORM FEED character that you embedded in EDIT exactly the same as an NP command.
4. An NP command given prior to the beginning of the text does not physically force a new page or change the page counter, but it does cause any headings in the file to be printed on the first page. (See section on Headings)
5. An NP command given at any other time, forces a physical page feed, increases the PAGE counter and resets the line counter to zero.
6. Part of an NP command is a carriage return, so if you typed the command

sequence `\CTR,NP\`, the next line would not be centered because the carriage return in the NP command turns off the centering switch. (`\NP,CTR\` does work)

7. **Note:** The NP command does *not* work in an initial section between the SETUP and TEXT commands.

NL — New Line

1. The NL command is the equivalent of pressing RETURN.
2. Use the NL command when you need to force a new line but cannot use a carriage return, e.g., from the keyboard at run time or in conditional print statements.
3. **Example 1:** You have a WAIT command embedded in your text to allow you to input comments into a letter. Without the NL command, you would be unable to put the comments into a separate paragraph.
4. **Example 2:** You are printing a paragraph. In the middle of the paragraph you have a conditional statement. If the statement is true, you end the paragraph at that point and skip the remainder (NL,SKIP). If the statement is false, you print the rest of the paragraph, ending it normally with a carriage return.

FF — Form Feed

1. To force a form feed without updating the page or line counters, type FF.
2. Typing FF is exactly the same as pressing the form feed button on your printer.
3. One of the main uses of the FF command is to reset your paper from the console prior to printing.
4. **Note:** Since the FF command does not update the page or line counters, using it in the middle of a text file may create confusion.
5. *FF and NP are not equivalent commands.* If you want to go to a new page while printing the text file, you should probably use the NP command.

FORMFEED — (Default ON)

1. When the machine reaches the end of a page or an NP command, it takes advantage of the form feed hardware in your printer.
2. If you are printing a file with pages other than the length set on your printer, you do not want this to happen. (Example: If your printer is set for a page of 66 lines and your file calls for a page length of 22 lines, your printer skips an additional 44 lines at the end of a page.)
3. To turn off the hardware form feed, type

FORMFEED OFF.

4. Now the machine moves to the new page line by line.
5. **Note:** You can probably adjust your printer to a different line length, but this is time consuming, and you must remember to reset it when you are finished.
6. Should you want to turn the form feed back on, type **FORMFEED ON**.
7. **Note:** On many printers you cannot tell the difference between a mechanical form feed and one done line by line.

PG — Setting the Page Number

1. The PG command allows you to change the current value of the %PAGE system variable (See section on system variables, below).
2. The format for the command is **PGn**, where *n* is any number between zero and 32767.
3. This is the only way you can change the value of %PAGE.
4. **Note:** The PG command only changes the way the pages are numbered. It does not affect the printing of the file.
5. **Example 1:** You are getting ready to print the second part of a report. The first part ended on page twenty and you want this part to begin on page twenty-one. Type **PG21**.
6. **Example 2:** You are printing multiple copies of a four page document, and you need to reset the page counter with each pass. Embed **PG1** at the beginning of the file following the TEXT command.

COPY — (Default 1)

1. If you wish multiple copies of a file, type **COPYn**, where *n* is any number.
2. When the machine reaches the end of the file, it automatically begins the next copy (or pass) of the file.
3. The machine does not automatically go to a new page or reset the page counter; if you wish this to be done, you must embed the commands to do so (See the sections on the NP and PG commands)
4. The machine continues to print passes of the file until the number you have specified has been printed.
5. Typing **COPY0** is a special case; at the end of each pass of the file, the machine pauses to ask you if you want a new copy.
6. Answering **Y** goes to the next copy; answering **N** discontinues the processing of the file.

SETUP/TEXT

1. The SETUP and TEXT commands mark off

- a section of the file that is only executed on the first pass through the file.
2. The SETUP command should be one of the very first commands in the text.
 3. Include in the SETUP section the definition of any external data file, the header and footer routines, first time only functions (e.g., getting the date) and anything else that only needs to be done the first time through a file.
 4. The machine executes all of the commands in this initial section on the first pass of the file.
 5. On subsequent passes, the machine passes over this section.
 6. This initial section is for commands only; the machine ignores any text characters in this section.
 7. Since the NP command generates text (i.e., a carriage return) the machine ignores any NP command in the initial SETUP section.
 8. With or without an accompanying SETUP command, the TEXT command marks the beginning of the text and should appear before any text is generated for printing in that file.
 9. The TEXT command must be the only command in the command line.
 10. On the first pass, when the machine encounters a TEXT command following a SETUP command, it halts processing the file, displays the status screen, and gives you a chance to give commands from the keyboard.
 11. The machine does not pause on subsequent passes.
 12. If you have defined a data file in the initial section, the machine does not open it until after the pause at the TEXT command.
 13. **Note:** If you do not have a TEXT command in the file, the machine prints the file without stopping at the beginning of text.

START/STOP

1. The START and STOP commands allow you to specify the record from an external data file at which you wish to start or stop processing.
2. To start processing at a particular record, type `START n` , where n is the number of the record.
3. To STOP processing at a particular record, type `STOP n` .
4. The START and STOP commands are independent of each other.
5. If you give a START command without a STOP command, the machine starts with

the record you indicate and continues until it reaches the End of File.

6. If you give a STOP command without a START command, the machine starts with the first record and continues until it has printed the STOP record, at which point it terminates the processing.
7. If your file does not have an external data file, the START/STOP commands determine which *page* to begin or end processing.
8. For example, if you were printing a 20 page document, but had to interrupt it after ten pages, you could begin printing at page eleven by typing `START11` from the keyboard.

END

1. The END command allows you to end a particular pass of the text file.
2. It is normally used to force a new record from a data file.
3. The END command must be the last command on a command line.
4. The END command is primarily a time-saving device, allowing you to skip the remainder of a file rather than processing it without printing it.
5. If you give an END command while processing a heading or footing, the machine ends the heading or footing and returns to the body of the text.

QUIT

1. The QUIT command unconditionally and immediately terminates all processing of the text file and ends PRINT.
2. You may embed a QUIT command in a file, but more likely you will use it to abort the current PRINT program from the keyboard.
3. As with the END command, the QUIT command must be the last command on a line.

VARIABLES

GENERAL

1. A variable is something that may have more than one value.
2. The opposite of a variable is a constant, which always has the same value.
3. A variable has a *name*, which you define, and a *value*.
4. The name of the variable may be up to seven characters.
5. The name may be any combination of numbers and uppercase letters as long as the first character in the name is a letter.
6. A variable is treated as if it were a com-

mand and may only appear in command lines. (In fact, it *is* a command, namely, substitute the current value of the variable for the variable name.)

7. As with other commands, the machine treats lowercase letters in a variable name as if they were uppercase.
8. The machine rejects any improper variable name and prints an error message on the screen.
9. The value of a variable is a string of up to fifty-five characters. (See glossary for String)
10. Any character may be included in the string, including lowercase letters, punctuation marks, blanks, etc.
11. You do not need to redefine a recognition character to include it in the string. As with other commands, recognition characters do not function when they are part of the value of a variable.
12. The machine shortens strings with more than 55 characters to the first 55 characters.
13. If you wish to refer to a specific number of characters at the beginning of the string, type $V(n)$, where V is any variable name and n is the number of characters between one and 55.
14. **Note:** This does not shorten the value that is stored in memory. You can only shorten that value if you set a limit when you define the value. (See sections below)
15. When referring to a variable, you must use its exact name, e.g., **ADDRESS** and **ADRESS** are two different variable names.
16. **Note:** If the machine should ever substitute blank or incorrect values for a variable in a text file, you should check to make sure that you have typed the correct variable name.
17. You may have no more than 128 variables in one file.
18. The default number of variables permitted in a file is 32; however, you may increase or decrease this number with the **VSIZE** command. (See the section on changing the memory buffer below.)
19. Any reference in a command line to a variable creates that variable and reserves a spot in memory for it.
20. Although you can create a variable by referring to it as a variable or as part of an **IF** statement, you can only assign a value to it with one of three commands — the **GET** command, the **SET** command and the **DATA** command.
21. Once you have created a variable name, you

cannot *uncreate* it. Variables exist as long as the print session lasts even if you do not use them.

GET — Assigning values at run time

1. The **GET** command allows you to define the value of a variable from the keyboard at run time.
2. The format for the **GET** command is **GET V**, where V is any legitimate variable name.
3. When the machine encounters the **GET** command, it halts processing the file and prints **Enter V** on the screen, e.g., if you type **GET NAME**, the machine prints on the screen, **Enter NAME**.
4. Type in the value of the variable and press **RETURN**.
5. The machine assigns what you have typed (up to the first fifty-five characters) as the value of the variable and resumes processing the file.
6. If you wish to limit the length of the variable to less than 55 characters, type **GET V(n)**, e.g., **GET NAME (25)** would shorten any name longer than 25 characters to 25 characters.
7. **Note:** Since you have set the limit prior to defining the variable, the machine stores in memory only the number of characters that you indicated.
8. Pressing **RETURN** without entering anything leaves the value of the variable unchanged.
9. If you wish to blank out the value of the variable, press the space bar before pressing **RETURN**.
10. You may, if you wish, define a prompt statement to be printed on the screen instead of the variable name.
11. To do this, type **GET V="prompt"**, e.g., **GET NAME="Full name (last name first)?"**.
12. The machine now prints the prompt [**Full name (last name first)?**] instead of the variable name [**Enter NAME**].
13. You may give a **GET** command directly from the keyboard as a means of assigning the value of a variable; however, you can accomplish the same thing more directly with the **SET** command.

SET — Assigning values directly

1. The **SET** command allows you to assign a value to a variable directly in the text without going to the screen.
2. The **SET** command also allows you to assign a value from the keyboard without using a

GET statement.

3. As with the GET command, you may use the SET command to assign a string value to a variable.
4. To do this, type `SET V="s"`.
5. This command assigns (sets) the string *s* as the value of the variable *V*, e.g., `SET STATE="Texas"` or `SET STATUS="In Progress"`.
6. Unlike the GET command, the SET command can set the value of one variable as the value of another variable.
7. To do this, type `SETV=V1`.
8. This command sets value of *V* equal to the value of *V1*, e.g., `SET NAME1=:NAME`.
9. This command does not affect the value of the second variable.
10. **Note:** Use this command when you need to store the current value of a variable that you are about to change to something else.
11. You can also use the SET command to store the truncated value of a variable, e.g., `SET NAME1(25)=:NAME`. (Again, this does not affect the value of the second variable.)
12. You may also use the SET command to increment or decrement (add to or subtract from) a numeric variable, or to add or subtract two or more numeric variables. (This is covered in detail in the section on Numeric Variables, below)
13. You may not use the SET command to add strings together, e.g., `SET FULLNAM=:FRSTNAME+:LASTNAM` and `SET SALUTE="Mr.":LASTNAM` are not valid commands.

FILE — The FILE Statement

1. You may also assign a value to a variable from an external data file, i.e., from a file other than the one being processed.
2. A data file is divided into records.
3. The machine reads one complete record from the data file at the beginning of each pass through the text file.
4. A record is divided into elements.
5. You may assign an element as the value of a variable.
6. You may access two kinds of data files: a *text* file and a *fixed* file.
7. An element in a text file is an indefinite number of characters ending in a carriage return.
8. Each record in a text file contains a fixed number of elements (carriage returns).
9. In a fixed file, each record, and each element in the record, contains a fixed number of characters.
10. If you want to define a text data file, type `FILE Tn`, where *n* is the number of elements (carriage returns) in each record, e.g., if you have a file with twelve elements per record, type `FILE T12`.
11. To define a fixed data file, type `FILE Fn`, where *n* is the number of characters in each record.
12. You may include the name of the data file in the FILE statement, by typing a comma followed by the complete name of the file, e.g., `FILE T10,CUSTOMER.INQ` or `FILE F128,B:NAMELIST`. (See section on Naming Files at the beginning of the Notes)
13. You can use a variable to supply the name of the file.
14. Example: Type `GET FILNAME,FILE T12,:FILNAME` (**Note:** The colon indicates that `FILNAME` is a variable. Without it, the machine would try to find a file named "FILNAME".)
15. If you did not name a file in the text or if the machine does not find the file you did name, it prints an error message on the screen.
16. You may define or change the name of the data file from the keyboard using the `FNAME` command.
17. The format for the `FNAME` command is `FNAME,FILENAME`.
18. You may use the `FNAME` command to change the name of the data file any time before the file has been opened.
19. The machine does not open the data file until it is ready to read the first record from the file.
20. You should always put the FILE statement in the initial section of your file between the `SETUP` and `TEXT` commands. (See section on `SETUP/TEXT` above)
21. When it reaches the `TEXT` command, the machine displays the current status of the system, and allows you to enter commands from the keyboard. This is your last chance to use the `FNAME` command, because the machine opens the file as soon as you give it the go ahead to process.
22. You may access no more than one data file in any given processing of a text file.
23. If you have already opened a data file, the machine ignores all additional `FILE` or `FNAME` commands.
24. The `FILE` statement only tells the machine how much of the data file is included in each record; it does not create any variables or assign values to them.
25. Use the `DATA` statement to create and define the file variables. (See the following

section)

26. **Note:** Since the FILE statement has multiple parts, it must be the last, preferably the only, command on a line.

DATA — The DATA Statement

1. The DATA statement determines how each element of a record is assigned to a variable.
2. If the data file is a text file, the format of the DATA statement is `DATA V1,V2,V3`, etc., where *V1*, *V2* and *V3* are names of variables.
3. Following this command, the machine assigns the first element of the record to the first variable, the second element to the second variable, etc.
4. If the data file is a fixed file, the format of the DATA statement is `DATA V1(n1),V2(n2),etc.`, where *V1* and *V2* are variable names and *n1* and *n2* are the numbers of characters in the variable.
5. Following this command, the machine assigns the first *n1* characters in the record to *V1*, the next *n2* characters to *V2*, etc.
6. **Note:** If you want to limit the length of a variable coming from a text file, you may include the length in the DATA statement following the variable name.
7. You do not need to assign every element in a record to a variable, but you must indicate the location of each element in the DATA statement.
8. An unassigned element from a text data file is represented in the DATA statement by an extra comma.
9. For example, if you have a six-element record and you want to assign the second element as NAME, the fourth element as PHONE and the fifth element as REGION, type `DATA ,NAME,,PHONE,REGION,`
10. **Note:** The final comma is optional, since the machine ignores elements left over after all the variables have been defined.
11. If you wish to skip over part of a fixed record, you must indicate the number of characters you are skipping, e.g., `DATA NAME(30),15,ADDR1(20),ADDR2(20),5,SALUTE(15)`
12. The machine ignores any unassigned characters left over at the end of the record.
13. You may use more than one DATA statement to assign the elements in a record, as long as you do not assign more elements or characters than you established in the FILE statement.
14. If you attempt to assign more elements or characters than defined for a record, the

machine rejects the command and prints an error message on the screen.

15. The machine treats each DATA statement as a continuation of the first DATA statement. For example, the following commands are equivalent:
`\DATA NAME`
`\DATA ADDR1`
`\DATA ADDR2`
or
`\DATA NAME,ADDR1,ADDR2`
16. You can only assign to variables elements that contain pure Ascii characters.
17. An unassigned element from a data file may contain non-Ascii characters, e.g., if you have a fixed file generated by a BASIC program that contains numbers stored in non-Ascii format, you may still use it as a data file as long as you pass over all the non-Ascii characters.
18. A text data record may contain no more than 128 elements.
19. An element in a text data record may contain any number of characters; however, you can assign only the first 55 characters of an element to a variable.
20. A fixed data record may not have more than 32,768 characters.
21. You may not assign to a variable more than 55 characters from a fixed data record.
22. You may pass over up to 255 characters in a fixed record in one jump.
23. **Note:** Since you may divide a fixed record any way you choose, you may assign more than 55 characters in a row by assigning them to separate variables. Similarly, you may pass over more than 255 characters by skipping over consecutive elements.

USING VARIABLES

1. You may assign values to a variable with a GET, SET or DATA command.
2. You may create a variable without assigning a value to it with a variable or IF statement.
3. Variables must be part of a command line.
4. To indicate a variable, prefix the variable name with one of several symbols: a colon (:), an equals sign (=), a dollar sign (\$), a number sign (#), an ampersand (&) or a percent sign (%).
5. In addition to indicating a variable name, these symbols tell the machine how the variable should be interpreted. (See below)
6. If you do not indicate that you are using a variable, the machine treats the variable

- name as if it were a command.
- When this happens, the machine usually prints an error message on the screen; however, if the variable name is the same as a command (and it could be), the machine executes the command.
 - Note:** You do not need to prefix the variable name in a GET, SET, DATA or IF command, unless you want the variable to be interpreted in a special way.

STRING VARIABLES

:VARIABLE (colon variable)

- The machine allocates 55 characters of memory for the current value of each variable that you create.
- Although the machine may interpret the value of a variable in different ways, it always stores the value in memory as a string of characters.
- If the value of a variable has less than 55 characters (or whatever shorter length you may have defined), the machine may add blanks following the last character.
- The colon in front of the variable name indicates that you wish to use the value with any trailing blanks stripped away.
- To print the value of a variable as part of the text, type `\:V\`, where *V* is the name of any variable.
- When the machine encounters this command, it prints the current value of *V* with any trailing blanks stripped away.
- If you wish to print a variable on a line by itself, you must either follow the variable with a command marker prior to the carriage return or follow the variable with an NL command, e.g., `\:NAME\~` or `\:NAME,NL`
- If you followed the variable directly with a carriage return, the machine would interpret the carriage return as the end of the command line, and not as the end of the text line.
- Note:** In most situations, you should use the colon variable whenever you print a variable or refer to one as part of a command.

=VARIABLE (equals variable)

- Placing an equals sign in front of the variable name returns the *untruncated* value of the variable, i.e., the value with any trailing blanks that it might include.
- When you assign a value to a variable with a GET, SET or DATA command without specifying the length of the variable, the

- machine stores the actual length of the value in memory.
- In this situation, the colon variable and the equals variable are equivalent since the machine does not store any trailing blanks.
 - If you specify a length while assigning the value to a variable, the machine stores that length as the length of the value and shortens or adds blanks to the value so that it has the specified length.
 - Example: If you type `SET DATE (30)="May 1, 1950"`, the machine adds 19 blanks behind the 11 characters you typed in to make exactly 30 characters.
 - For most purposes, you should use a colon variable instead of an equals variable, since an untruncated string could put unwanted blanks in the text.
 - Although you use the equals variable infrequently, a variation of the equals format is very important and common, namely, the `"string"`, which is used to print literal (string) statements.
 - The `"` (equals quote) command allows you to insert text into the printed document from the keyboard at run time.
 - It also allows you to print text as a part of a command line, usually only part of a conditional statement. (See section on Conditional Commands, above)
 - If you are inserting text from the keyboard, the maximum length of your string is the width of your screen, e.g., if your screen is eighty characters wide, your string may be no more than 78 characters (two characters being used by `"`).
 - If you exceed the maximum length while entering the string, the machine refuses to accept any additional characters and immediately prints what you have typed thus far.
 - If the text that you want to insert exceeds one screen line, you may continue where you were interrupted by giving additional equals quote commands.
 - The machine prints each successive equals quote command immediately following the previous text without a break.
 - Use the NL command to insert a carriage return/line feed into the text while using the equals quote command.
 - Note:** If what you have entered does not fill a line on the printed page, the machine stores the string in memory until the remainder of the line is filled in from the file or from the screen or until it receives a carriage return or NL command.

16. If you are using the equals quote command as part of a command line embedded in the text, the string must not cause the command line to exceed the 256 character maximum, e.g., if you preceded the equals quote command with 30 characters of commands, the string may be up to 226 characters long.
17. **Note:** As with all other commands and variables, characters that you include as part of your string cannot function as recognition characters; consequently, you cannot bold-face, underscore, etc. in the middle of a line of text entered from the keyboard.

\$VARIABLE (Dollar Variable)

1. The dollar sign in front of the variable name indicates that you want the value of the variable formatted with commas and a decimal point.
2. You may have as many as twelve places to the left of the decimal and two places to the right of the decimal, i.e.,
999,999,999,999.99.
3. The primary purpose of the dollar variable is preparing raw numbers coming from a data file to be printed as dollar amounts.
4. The dollar sign in front of the variable name does not add a dollar sign in front of the printed number. If you want a dollar sign to appear in front of the value when it is printed, you must put one in the text in addition to the one in front of the variable name.
5. When converting the value of a variable to dollar format, the machine ignores all characters except numbers, periods, plus and minus signs.
6. This means that numbers that are already formatted can be treated just like those that have no commas or dollar signs.
7. The machine treats the first period it finds in the value of the variable as the decimal point and ignores any subsequent periods.
8. If there is no period in the string value, the machine prints whole numbers only without a decimal or two decimal places.
9. If the machine finds a period, it prints exactly two characters on the right hand side of the period.
10. If the machine finds less than two numbers following a period, it fills in with zeros.
11. If the machine finds more than two characters following a period, it prints only the first two.
12. Examples: 10000 becomes 10,000
10000. becomes 10,000.00
10000.1 becomes 10,000.10
10000.102 becomes 10,000.10
13. The presence of a minus sign (or dash) in the value of the variable causes the value to be treated as a negative number (unless for some reason there is a plus sign later in the string).
14. When the machine prints a negative value, the minus sign follows the last character, i.e., XXXXX-.
15. You may put a dollar sign in front of any variable without getting an error message.
16. If the string value does not contain any numbers, the dollar value is zero.
17. **Note:** You may use the dollar variable in conditional (IF) commands to compare two numbers. This is the only way to compare very large, fractional or negative numbers. (See section on conditional commands, below)
18. If you want to change to international format from american format, i.e., periods in place of the commas and a comma for the decimal point, type **DECIMAL C**. To change back to american format, type **DECIMAL P**.

NUMERIC VARIABLES

1. A numeric variable is one whose value is treated as a number and not just a string of characters.
2. The main difference between a number and a string is that a number can be used in an arithmetic function, whereas a string can only be printed or compared as is.
3. Any string value can also have a numeric value (even if that value is zero).
4. Numeric variables (or a numeric expression) may be substituted in place of numbers in any command that takes a number, e.g., left margin, page length, etc.
5. **Note:** The only command that does not take a variable in place of a number is **BF**, which only allows BF1 through BF9.
6. You may add or subtract numeric variables to constant numbers or to other numeric variables by using the SET command, e.g.,
SET #TIMES = #TIMES + 1, SET #LEFT = #LEFT - 1, SET #TOTAL = #PAID ÷ #FREE - #NOSHOW. (See Section on Number Variables, below)
7. A numeric variable must be a whole number.
8. A number containing a fraction is rounded down to the nearest whole number, e.g., 10.5 becomes 10, 1.99 becomes 1.
9. A numeric variable may not be larger than 32767 or less than zero.
10. Numbers less than zero are treated as zero.

11. The machine rejects and prints an error message for numbers greater than 32767 or numeric expressions evaluating to less than zero or greater than 32767.
12. **Note:** Although you may add and subtract numeric variables and constants for the purpose of setting up counters and doing comparisons, the machine can perform only the most basic arithmetic functions. You should do any complex calculations prior to including a number in a text or data file.

#VARIABLE (Number Variable)

1. Putting a number sign in front of a variable causes the machine to treat the string value of the variable as a numeric value.
2. When converting a string to a number, the machine reads the string character by character, treating all characters 1 through 9 as part of the number until it encounters a character other than 1 through 9.
3. **Note:** Decimal points and minus signs are not numbers which is why the machine does not recognize fractions or negative numbers.
4. If the number exceeds 32767, the machine rejects it.
5. If the value of the variable is blank or if it begins with a non-numeric character, the machine treats it as zero.
6. Examples: The figures on the left are string values; the figures on the right are those strings converted to numbers:

"12345"	12345	"123,456"	123
"123.45"	123	"123 Main"	123
"-12345"	0	"123456"	ERROR
7. A number variable may be added to or subtracted from any numeric variable or constant. This is called a numeric expression.
8. A numeric expression may have in it any number of numeric variables or constants.
9. A numeric expression is treated as if it were a single number, the result of the calculation being the value of the expression.
10. If the value of an expression is greater than 32767 or less than zero, the machine rejects it and prints an error message.
11. You may use a number variable or expression in any command that takes a numeric parameter.
12. Example: If you were going to print a file but could not know ahead of time what the starting page number would be, you could put this series of commands in the SETUP section of the file:


```
\GET PAGE ="Starting page number?"
\PG #PAGE
```
13. You may not use a number variable or expression to set the intensity of bold face.
14. If you wish to give a negative value to a number variable as part of a command, you must place a minus sign in front of the numeric variable, e.g., `PI-#V`, `K-#V`.
15. You may also compare a number variable to any other numeric variable, expression or constant in an IF statement. (See Section on Numeric Comparisons, below)
16. You may use the SET statement to set the number value of a variable to any other numeric variable, expression or constant.
17. **Note:** When setting number variables, do not put quotation marks around the value since quotation marks are used to indicate strings and not numbers.
18. Example: To set a variable equal to a constant number (e.g., three), type `SET #V=3`
19. You may use the SET command to establish *counters* in the file to keep track of how many times a particular function is performed.
20. Example: To set up a variable to keep track of how many times you print a particular portion of a text file, you might type `SET #TIMES=#TIMES+1`
21. **Note:** This command increases the value of #TIMES by one, and it works because the machine does not change the value of #Times until after the command is executed.
22. You may also use the SET command to store the current value of some other numeric variable.
23. Example: To set a variable equal to the current page number, type `SET #PAGE=%PAGE`.
24. **Note:** All length (&) variables and system (%) variables are numeric variables. You may not set a string value equal to a numeric variable, e.g., `SET PAGE=%PAGE` would be rejected.
25. Even though you assign the value to the variable as a number, the machine stores the value as a string which may be treated like any other string value.
26. Since the value is stored as a string, you must always put the number sign in front of the variable name when you wish to use its numeric value, even if you originally assigned it a numeric value.
27. **Note:** Although the numeric value of a blank variable is interpreted as zero, you must use the SET command if you actually wish to assign a value of zero to a variable.

The difference between the two is that the blank value has no length, whereas the value of zero has a length of one.

&VARIABLE (Length Variable)

1. When you place an ampersand in front of a variable name, the machine returns the length of the string value of the variable, truncated to the last non-blank character, i.e., the length of the colon variable.
2. The length variable has a numeric value and may be used like any numeric variable, e.g., in numeric expressions, command parameters, etc.
3. An important use of the length variable is to test the length of a string variable to make sure that it fits in a particular spot.
4. Example: The following series of commands gets a name from the keyboard, tests its length and, if it is longer than 30 characters, truncates it to 30 characters, notifies you what has happened and gives you a chance to re-enter the name.

```
\GET NAME
\IF &NAME<=30,SKIP3
\SET NAME =NAME(30)
\SHOW "Name must be shortened
to ",:NAME
\GET NAME ="If OK press RETURN,
else re-enter name"
```
5. Another important use of the length variable is lining up variables in columns, a tricky task because you do not know ahead of time how long they are, e.g.,

```
\TAB20 -&NAME,:NAME\
```

would print the value of NAME with its last character on column twenty.

%VARIABLES (System Variables)

1. A system variable is a numeric variable maintained automatically by the machine to keep track of where you are in the processing of a file.
2. A system variable may be used any place that you can use a numeric variable.
3. You may not change a system variable with a GET, SET or DATA command.
4. The system variables are:
 %PAGE, %PASS, %REC, %LINE,
 %LINES, %COL, %EOF.
5. Only a system variable can take a percent sign. If you put a percent sign in front of any other variable name, the machine prints an error message.
6. You may have a variable with the same name as a system variable, e.g., PAGE or REC; however, its value bears no relationship to its system counterpart.

%PAGE

1. %PAGE represents the current page number of the file you are printing.
2. Each time the machine goes from one page to another, between the footing of one page and the heading of the next, it adds one to the value of %PAGE.
3. Unlike other system variables, you may change the value of %PAGE.
4. To do so, use the PG command. (See the section on the PG command, above)
5. Note: The machine does not reset %PAGE at the end of a pass of the text file. If you want the page counter reset to one (or any other number) you must use the PG command.

%PASS

1. %PASS tells you the current pass number, i.e., how many times you have gone through the file including the current time.
2. Note: Actually, it keeps track of how many times you *begin* the file, since you may end it prematurely with the END command.
3. %PASS is used most often when printing files from external data files.

%REC

1. %REC tells you the current record number being processed from an external data file.
2. If you start at record one, %REC and %PASS are the same.
3. However, if you start at a record other than one, %REC and %PASS are not the same, e.g., if you start at record 5 (type START5), %REC is 5 while %PASS is still 1.

%LINE

1. %LINE tells you the current line number.
2. The line number is counted from the top of the page, including the top margin, e.g., if you have a top margin of three lines (TM3), the body of the text begins on line four.
3. A fractional line is rounded down to the nearest whole line.
4. Each time the machine goes to a new page, it resets %LINE to zero.

%LINES

1. %LINES tells you how many more lines are left in the body of the text for the current page.
2. The current line is included in the number of lines remaining.
3. Example: If you are on line 50 of a 66 line page and you have a 6 line bottom margin,

`%LINES` is 11 (50 through 60 is eleven lines).

- Use `%LINES` when you do not want to break certain lines of text at the end of the page.
- Example: You have a table of figures that takes up eight lines, and you want to make sure that the machine does not end a page in the middle of it. On a command line immediately preceding the table, type `\IF %LINES < 8, NP`. The machine now makes sure that it has room for the table before it starts to print, and if it does not have room, it forces a new page.

`%COL`

- `%COL` tells you the current column location on the print line.
- `%COL` can help you return to a specific column which could not be determined before processing the file.
- Example: If you were printing a file and wanted to store certain information on disk without losing your place in the file, you would turn off the printer, turn on the disk, set the spacing to zero, and assign `%COL` to a variable. When you were finished writing to disk, you would turn the printer back on, tab out to the column you had saved previously, and resume printing of the file where you left off.

`%EOF`

- `%EOF` tells you if you have processed the last record in your data file.
- When you reach the end of the data file, `%EOF` equals 1; at all other times it equals 0.
- When the machine reaches the end of the data file, it checks to see if there are any special commands in the FOOT routine before it terminates processing.
- This means that you may use the FOOT routine to create text that is only processed when the end of the file is reached.
- Example: You have a file containing three different letters which are printed depending upon variables in the data file. You have embedded in the file number variables as counters to keep track of how many times you print each kind of letter. The counters are named LTR1, LTR2 and LTR3. You also have a counter for records that had no letter printed called LTR0. After the last record has been processed, you want the machine to skip to a new page and print a recap showing how many of each letter were

printed. To do this, you would create a footer record as follows:

```
\FOOT7
\IF %EOF=0,END
\FF,CTR\Recap of processing for
\ :DATE\,NL,NL
Letter one\TAB20,:LTR1\ records,NL
Letter two\TAB20,:LTR2\ records,NL
Letter three\TAB20,:LTR3\ records,NL
No letter\TAB20,:LTR0\ Records,NL
\TAB10\TOTAL\TAB20,%PASS\
records processed
```

- Note: How to format other footings is covered in the section on the Foot Command below.

CONDITIONAL COMMANDS

IF STATEMENT

- A conditional command is one that is executed only if a condition that you establish is true.
- The condition must be a comparison between two expressions.
- An expression may be a variable, a constant or a combination of the two.
- Note: The first expression in the comparison must contain a variable, since you already know the outcome of a comparison of constants.
- You may test to see if one expression is greater than (>), less than (<) or equal to (=) the other expression.
- You may test for any two relationships at one time, i.e., greater than or equal to (>=), less than or equal to (<=), greater than or less than, i.e., *not* equal to (<>).
- You establish the condition with the IF statement.
- The format of the IF statement is: *IF 1st Expression Comparison 2nd Expression*.
- Examples:
`IF CODE="Y"` (If the value of CODE equals Y);
`IF #TIMES >= 5` (If the numeric value of TIMES is greater than or equal to 5);
`IF ADDR1 <> ""` (If the value of ADDR1 is not blank).
- You may, if you wish, use NOT in conjunction with IF to test if a condition is not true, e.g., `IF NOT #TIMES < 5` (If the numeric value of TIMES is *not* less than 5).
- Note: If the name of variable begins with "NOT", you must put a colon in front of it, or the machine interprets it as IF NOT.
- Any legitimate command or series of com-

mands may follow an IF statement separated from it by a comma.

13. When a condition is true the machine executes the rest of the commands in the command line.
14. When a condition is not true, the machine passes over the remaining commands in the command line.

COMPARING STRINGS

1. You may compare a string variable to any other literal string or string variable.
2. You do not need to prefix the first variable name with a colon or an equals sign (although you may if you wish), because the machine assumes that a variable is a string unless you specify otherwise.
3. If you are comparing a variable to a literal string, you must enclose the literal in quotation marks, either single (' ') or double (" ").
4. The format for comparing a string variable to a constant is: **IF** [or **IF NOT**] *VARIABLE* = "Constant".
5. **Note:** The second quotation mark is *essential*, otherwise the machine thinks that the following commands are part of the constant.
6. The format for comparing the value of one string variable to another is: **IF** [or **IF NOT**] *VARIABLE1* = :*VARIABLE2*
7. The machine considers two strings to be equal if they contain the same characters in the same order.
8. **Note:** Upper and lowercase letters do *not* compare as equal.
9. The machine does not count trailing blanks when testing for identity, i.e., if two strings are identical except for trailing blanks, the machine considers them equal.
10. **Note:** For this reason, a variable that has no characters assigned to it is equal to a variable or string made up entirely of blanks.
11. The machine does consider *leading* blanks when testing for identity, e.g., " Mr. Brown" is not equal to "Mr. Brown".
12. When the machine compares strings, it compares them character by character, i.e., the first character of one string is compared to the first character of the other string, the second character to the second character, etc.
13. For this reason you should not test to see if string values are greater than or less than each other except in a few limited cases.
14. Example: The expression **IF "300" > "1000"** would be true, because the three in 300 is larger than the one in 1000.
15. **Note:** To compare numbers, you must use either a numeric or dollar comparison (See following).
16. You may test to see if one string of numbers is greater than another string of numbers if they have the same number of characters, e.g. a Zip Code.
17. **Note:** Since a Zip Code could exceed 32767, you could not rely on a numeric comparison, and would need to perform a string comparison.
18. You may also test strings for alphabetical order using a less than or greater than comparison.
19. For example, if you wanted to test to see if a name came before **M** in the alphabet, you would type **IF NAME < "M"**. (Remember, only the *first* character is important for this comparison.)
20. **Note:** The machine compares based on the Ascii value of the character. This means that a blank is less than a number which is less than an uppercase letter which is less than a lowercase letter. The various punctuation marks are scattered among the other characters.
21. If you wish, you may compare only a portion of a variable by enclosing the number of characters you wish to compare in parentheses following the variable name.
22. Example: If you want to test the first character of a name to see if it is **M**, you would type **IF NAME(1) = "M"**. If the first character is **M**, the expression is true, no matter what characters follow it.
23. **Note:** If you wish to compare a fixed number of characters from two variables, you must include the number of characters with each variable name, e.g., **IF NAME(10) = NAME1(10)**.

NUMERIC COMPARISONS

1. You may make numeric comparisons when you have numeric variables as the first part of the IF statement. (See section on Numeric Variables, above)
2. You may only include numeric variables, expressions and constants in a numeric comparison.
3. Do not put constants in a numeric comparison in quotation marks.
4. The same restrictions that apply to numeric variables apply to numeric comparisons as well, i.e., the numbers must be positive whole numbers between zero and 32767.
5. Numbers greater than 32767 or less than zero are treated as zero.

6. Either side of the comparison may be a numeric expression instead of a single variable or constant, e.g.,
`IF #TIMES+3 >%PAGE,`
`IF &NAME < #LENGTH -5, etc.`
7. You must use a numeric comparison to compare a length variable (&VARIABLE) or a system variable (%VARIABLE), both of which are always numeric.
8. If you wish to add or subtract numbers as part of a comparison, you must use a numeric comparison.
9. If you are simply comparing one number to another, you should probably use the dollar comparison (also called decimal comparison), to avoid the restrictions of numeric variables.

DOLLAR COMPARISON (Decimal Comparison)

1. Use the dollar comparison to compare the numeric value of a variable to another variable or constant.
2. **Note:** The value of the variable does not have to be actually in dollars, as long as it is a number.
3. In a dollar comparison, the expression immediately following IF must be a dollar variable (See section on Dollar Variable, above).
4. You are allowed up to twelve places to the left and two places to the right of the decimal point.
5. You may use either positive or negative numbers.
6. You may not use numeric expressions, e.g.,
`%PAGE +2, #TIMES -1, etc.`
7. If you are comparing the variable to a constant, place the constant in quotation marks.
8. You may include as part of the constant a plus or minus sign, commas and a decimal point and a dollar sign, or you can just use a series of numbers.
9. If you are comparing one variable to another, you may put the second variable in dollar format as well.
10. When performing a dollar comparison, the machine automatically puts both sides of the comparison in full dollar format, filling in with zeroes as necessary.
11. For example, `10000.1` becomes `000,000,010,000.10`, `-303,030.30` becomes `000,000,303,030.30 -`, etc.
12. After placing both sides in dollar format, the machine performs a string comparison of the two sides.
13. Since the dollar format places the numbers in a fixed format, the problems associated with comparing strings have been eliminated; consequently, you may use the dollar comparison to test any relationship, greater than and less than as well as equals.
14. **Note:** The dollar comparison is the only way to compare numbers larger than 32767, less than zero or fractions.

THE SKIP COMMAND

1. The SKIP command enables you to pass over a portion of the file without processing it.
2. By itself, the SKIP command is rarely used, but in conjunction with an IF statement, it becomes a powerful means of arranging a file.
3. The format of the SKIP command is `SKIPn`, where *n* is the number of carriage returns in the text that you wish to skip.
4. When the machine encounters the SKIP command, it passes over the file, ignoring commands as well as text, until it reaches the *n*th carriage return.
5. All carriage returns are counted toward *n*, whether they end a text line, a command line or blank line.
6. The machine begins processing immediately following the *n*th carriage return.
7. If the command line containing the SKIP command ends with a carriage return, that one carriage return does not count toward *n*.
8. If the command line ends with a command marker, the next carriage return does count as one of the lines being skipped.
9. If no number is specified in the SKIP command, the machine skips one carriage return.
10. Instead of skipping a number of carriage returns, you may skip to the next occurrence of a particular character.
11. To do this, type `SKIP TO c`, where *c* is any character except the command marker.
12. When the machine encounters the SKIP TO command, it passes over the text file until it reaches the next occurrence of *c*.
13. The machine begins processing immediately with *c*.
14. If you do not wish to print *c*, you should establish *c* as the IGNORE character. (See section on IGNORE character, above)
15. **Note:** The machine skips to the first occurrence of *c* whether it is part of a command line or part of the text. Consequently, you

- should be careful that the character you skip does not occur before you expect it.
16. If *c* is part of a command line, the machine prints the command line instead of executing it, because it passed over the command marker.
 17. If you wish to skip to a command line, you must place *c* immediately in front of the command marker.
 18. Both the SKIP and SKIP TO commands must be the last command on a command line, since any command following them is passed over.

MULTIPLE CONDITIONS

1. When you test more than one condition at a time, you are establishing a multiple condition.
2. There are two kinds of multiple conditions: the AND condition and the OR condition.
3. The AND condition requires that *all* of the conditions must be true for the command to be carried out.
4. The OR condition requires that *at least one* of the conditions must be true for the command to be carried out.
5. Although the machine does not recognize the words AND or OR, you can easily establish multiple conditions.
6. To establish an AND condition, put all of the IF statements in the same command line in front of the conditional command.
7. If any *one* of them is not true, the machine skips the remainder of the command line.
8. Consequently, *all* of them must be true for the command to be carried out, and the AND condition is established.
9. The OR condition is trickier to establish than the AND condition, but you have two ways to go about it.
10. You can put each condition in separate command lines with the conditional command repeated at the end of each.
11. Example: You want to force a new page if a variable named CODE has a value of A or B; you should type:

```
\IF CODE="A", NP
\IF CODE="B", NP
```
12. Since CODE cannot be both A and B at the same time, the machine can only execute the NP command once.
13. It is possible to establish an OR condition in which more than one of the conditions can be true at the same time.
14. When this happens, you must prevent the machine from executing the command more than once.

15. Example: You want to force a new page if CODE equals either A or B or the current page number is less than two; you should type:

```
\IF CODE="A",NP,SKIP2
\IF CODE="B",NP,SKIP
\IF %PAGE<2,NP
```
16. **Note:** If you do not include the SKIP commands, then you would force two new pages if CODE equals A or B *and* the current page number is 1.
17. Instead of writing different command lines, you can combine the conditions in one AND statement.
18. To do this, reverse the conditions by following IF with NOT followed by a SKIP command.
19. Example: To rephrase the commands given above, you would type:

```
\IF NOT CODE="A", IF NOT
CODE="B", IF NOT %PAGE<2,SKIP
\NP
```
20. **Note:** This does the same as line 15 above, because if CODE equals either A or B or %PAGE equals 1, then the condition is false and the machine goes immediately to the line containing the NP command. If none of those conditions is met, the machine executes the SKIP command and passes over the NP command.

HEADINGS

1. The heading is a series of text and/or command lines that is processed at the beginning of each page in a multi-page document and at the beginning of each pass in a multipass document.
2. A heading does not necessarily contain anything to be printed on the page.
3. You may use the heading section to print information on the screen, print to disk, update a counter or perform any other non-print functions.
4. The machine does not process the heading section on the first page of a document or first pass of a file unless you give an NP command immediately following the TEXT command.
5. **Note:** If you do give the NP command at the beginning of the file, the machine does not skip to a new physical page and does not increase the page counter.
6. You designate a heading by typing `\HEADn`, where *n* is the number of lines (ending in carriage returns) following

the HEAD command that are part of the heading.

7. The HEAD command must appear on a line by itself. The carriage return at the end of that line does not count toward n .
8. For example, the following is a two line heading:

```
\HEAD2
\LEFT,SPO,:COMPANY\
\RIGHT\Page \%PAGE\
```
9. The machine prints the heading starting on the first line of the page.
10. Be sure to adjust your top margin to leave sufficient room for the heading.
11. If the top margin is not sufficient for the heading, the machine takes as many lines as necessary to print the heading and begins printing the body of the text immediately following the heading.
12. The print parameters in the heading do not need to be the same as those in the body of the text or the footing.
13. You must include as part of the heading section any of the print parameters that may differ in the heading as opposed to the body of the text.
14. The machine only changes those parameters that you specify; if you have any doubt about whether you need to include a command, you should include it.
15. When the heading is completed, the machine automatically returns to the parameters that were in effect when the end of the previous page was reached.
16. If you want to end the heading before processing all of the lines and return to the body of the text, type **END**.
17. You may change a heading at any point in the text by giving another HEAD command.
18. Any time you change a heading, you erase the previous heading.
19. The heading is stored in memory.
20. Normally, you have 1000 characters of memory allocated for the heading.
21. You may change the amount of memory that the machine reserves for the heading with the HSIZE command. (See section on Adjusting Internal Buffers, below)

FOOTINGS

1. A footing is just like a heading except that it goes at the bottom of the page.
2. All of the rules that apply to headings apply to footings as well, except those indicated below.
3. You designate a footing by typing **FOOT n** ,

where n is the number of lines (carriage returns) in the footing.

4. The machine goes to the footing routine when it reaches the bottom margin of any page (including the first page) or when you give an NP command.
5. If the footing generates any text to be printed, the machine begins printing it on the first line of the bottom margin, immediately following the body of the text.
6. If you want to leave any space between the text and the footing, you must include additional lines as part of the footing.
7. **Note:** Since the bottom margin often extends past the physical bottom of the page, you must be sure that you have enough room for your footing.
8. The machine also goes to the footing routine whenever it reaches the end of the file.
9. The end of file could be the end of a single pass text file or the end of an external data file.
10. Since this is the case, you may place in your footing routine a series of lines that are only executed at the end of the file. (See Section on %EOF under System Variables, above)

PRINTING TO DISK

1. You may print all or a portion of a processed file to disk instead of or in addition to printing it on the printer.
2. To begin printing to disk type **DISK ON**; to halt printing to disk, type **DISK OFF**.
3. When the machine receives the DISK ON command, it immediately opens a file on the same disk as the text file with the same File Name and a PRN File Type, e.g., if your text file is REPLIES.GET, the output file is called REPLIES.PRN.
4. You may open an output file under a different name or on a different disk by typing **DISK FILENAME** instead of DISK ON.
5. The machine now opens a file called FILENAME on the disk specified.
6. If the machine finds that a file with the designated name already exists, it prints an error message on the screen and gives you the opportunity to rename the file.
7. You may open no more than one disk output file per processing of a given text file.
8. Any attempt to open a second file is treated as DISK ON.
9. You may close and reopen the same file any number of times.
10. If you close and reopen the disk output file, the machine resumes printing to the file

where it left off.

11. If you wish, you may print to disk without printing on the printer.
12. To do this, you must first turn off the printer by typing **PRINT OFF**.
13. You may resume printing by typing **PRINT ON**.
14. The files that you print to disk are fully formatted for printing, and may be printed in background while editing another file. (See section on Printing in Background, in the Notes on EDIT, above)
15. If the disk output file does not contain non-Ascii characters, you may also edit it with the Editor.
16. Even if the printer is turned off, the machine treats the text being generated as part of the print file, and updates the line and page counters.
17. If you do not want the portion of the file that you are printing to disk to count toward filling the printed page, you must inactivate the line feed by typing **SPO** prior to turning the disk on and restore the line feed after you turn the disk off.
18. If you want to resume printing at the same point on the line as when you turned the printer off, you must store the column number (**%COL**) where you left off and then **TAB** out to that column. (See section of **%COL**, above)
19. By selectively turning the disk output file on and off, you can create a variety of files, e.g., a list of addresses for labeling envelopes, an index of subjects by page number, etc.

ADJUSTING INTERNAL BUFFERS

1. After the machine loads **PRINT**, it reserves some of the remaining memory for three different buffers: the variable buffer, the heading buffer and the footing buffer.
2. After the machine has loaded **CP/M** and **PRINT** and reserved space for these three buffers, the remaining memory may be used to store the text file.
3. **Note:** Use the **DB** command to determine the current size of the three buffers and the amount of memory remaining for the text file. (See section on the **DB** command, below)
4. On the first pass as the machine processes the text file, it loads the file into memory a record at a time.
5. **Note:** A record contains 128 characters. (Also called a *sector*)

6. If the entire file fits in memory, the machine does not need to read from the disk on subsequent passes.
7. This could save you a considerable amount of time.
8. If the machine does not have enough memory for the entire file, it loads as much as it can.
9. On subsequent passes, the machine processes the portion of the file in memory first and switches to the disk when that runs out.
10. If it is important to try to fit the entire text file in memory, you may try to create enough room by reducing the size of the other buffers.
11. Since the machine can process the file even if none of it fits in memory, it is by no means necessary to reduce the size of the buffers just to fit the file into memory.
12. **Note:** You do not need to reserve any memory for the text file, so if you need to increase the size of the other buffers, you may do so until the capacity of the machine is exhausted.
13. You may adjust the size of the buffers only once in a text file, and you should do so immediately following the **SETUP** command at the beginning of the file.

VSIZE

1. When **PRINT** is initialized, the machine allows you to define 32 variables.
2. The machine reserves 64 characters for each variable that it allows (7 for the name, 55 for the value, and 1 each for the actual lengths of the name and value).
3. This means that the machine initially reserves more than 2000 characters in memory for variables.
4. You may use the **VSIZE** command to increase or decrease the number of variables allowed (and, consequently, the size of the variable buffer as well).
5. The format of this command is **VSIZE n** , where n is a number between zero and 128.
6. The machine rejects any attempt to define more than the maximum number of variables.
7. If you attempt to do so, the machine prints an error message on the screen.
8. You must give the **VSIZE** command before attempting to define more than 32 variables, preferably at the very beginning of the file.

HSIZE, FSIZE

1. When you create a heading or a footing with

John Doe and the current record number is 5. If you type `SHOW "Record no.", %REC," ";:NAME` the machine prints
Record no. 5 John Doe.

CLS (Clear Screen)

1. The CLS command clears the screen and positions the cursor at the first position of the first line.
2. Use the CLS command in conjunction with the NOTE, WAIT and SHOW commands to simplify the screen.

DS (Display Status)

1. When you give the DS command, the machine clears the screen and prints the status screen.
2. The status screen gives you the current values for all of the major shaping and formatting parameters.
3. It also tells you where you are in the text — the pass, page, line and column.
4. It shows you which files are being used — text, data and disk output.
5. It indicates the number of variables that have been defined so far in the text
6. It gives the current recognition characters.
7. It indicates the source of the current line, i.e., whether it is from the text, the heading or the footing.
8. Use the DS command whenever you need to check to make sure that everything is set the way you want it to be.

DV (Display Variables)

1. When you give the DV command, the machine clears the screen and prints the names, current lengths and values of all of the variables that have been defined in the text.
2. The DV command displays only the string values of the variables.
3. If you want to display the value in a different format, or if you want to display a

system variable, you must use the show command. (See above)

4. **Note:** If you think that you may have mistyped a variable name, the DV command can help you find the error immediately because you will have a variable name with no value attached to it.

DF (Display File Variables)

1. When you give the DF command, the machine clears the screen and prints the names, current lengths and values of all of the variables defined in the DATA statement.
2. It also indicates the record number currently being processed.
3. **Note:** Embedding the DF command in the file is the easiest way you have to keep track of the current record being processed from the data file.

DB (Display Buffers)

1. When you give the DB command, the machine prints on the screen the current sizes of the variable, heading and footing buffers, and how much room is left over for the text file.
2. The machine indicates the number of variables allowed, the size of the heading and footing buffers in characters and the number of records of the text file that will fit in memory.
3. To calculate the number of characters in the variable buffer, multiply the number of variables by 64.
4. To calculate the number of characters from the text file that will fit in memory, multiply the number of records by 128, e.g., if you had room for 100 records in memory, you would have room for a text file containing 12,800 characters.
5. The machine does not indicate how much of these buffers is actually being used.

SOME NOTES FOR PROGRAMMERS

FILE DEFINITION

- A. EDIT supports pure Ascii files following CP/M conventions, so you may use it to maintain data files and source programs. The following notes apply primarily to these non-text applications.
- B. EDIT uses Ascii characters 32-126 (blank through tilde), Carriage Return (13), Line Feed (10), Form Feed (12), and Tab (9). It determines end of file by a control-Z (26), or by the physical end of the file.
- C. The machine replaces tab characters upon input with one or more blanks up to the next 'eighth' column, according to CP/M standard usage.
- D. When saving a file from EDIT, the machine compresses blanks into tabs wherever possible to conserve disk space.
- E. The use of tab characters (or *blank packing*) is independent of the Tab Key and Tab Settings in EDIT which you use purely to help you position the cursor.
- F. The PRINT module ignores linefeeds. EDIT tolerates them, consistent with standard usage.
- G. EDIT strips the LF from each CRLF pair as it inputs a file and restores them when it outputs the edited file.
- H. EDIT accepts a linefeed character as an Ascii character of no special significance.
- I. EDIT allows a linefeed followed by a carriage return to accommodate Microsoft Basic conventions (See Section on Microsoft Basic, below).
- J. The LFCR pair suppresses the trailing linefeed normally added at output for each carriage return.
- K. EDIT ignores the high-order bit of each input byte as well as any nulls in an input file. It replaces any other non-standard character with a blank.
- L. Any time you open a file in EDIT, whether for input, display, include or printing, the machine scans the first sector for any non-standard characters other than nulls. If it finds any non-standard characters, it issues a warning message. (This prevents loading garbage while allowing you to salvage partially damaged files.)

MODE SETTINGS

- A. EDIT has three processing options that define the operating Mode.
- B. You may set or reset them at any time dur-

- ing an edit, although you should not usually need to change the mode.
- C. The two most frequent combinations of these options are designated Text Mode and Program Mode. Any other is considered *Special Mode*. (See the Section on Special Formatting Features, in the EDIT notes.)
- D. EDIT selects the initial mode setting based upon the file type of the *output* file.
- E. File types ASM, MAC, COB, FOR, BAS, and PRN invoke Program Mode automatically.
- F. File type ASC invokes the Special Mode with all options off (See the Section on Microsoft Basic, following.)
- G. If the file type is not one of these, the machine begins in text mode.
- H. **Note:** You may set the line size and tab markers to conform to any special formats required by your language processor.

MICROSOFT BASIC

- A. Most languages compatible with CP/M accept standard text files directly from EDIT, but Microsoft Basic requires some special considerations since it predates CP/M conventions.
- B. You should create most of your programs within Basic if you wish to take advantage of the automatic line numbering and renumbering features.
- C. To edit a program created in Basic, you must first save it in Ascii format (e.g., SAVE "PROGNAME.ASC",A).
- D. **Note:** The file type ASC allows you to take advantage of EDIT's assumption of Special Mode and to distinguish the file from the compressed BAS file type. The Special Mode invoked by file type ASC is Program Mode with the Blank Packing option turned off.
- E. You should avoid tab characters while creating the file since Basic assumes tabbing to every tenth, rather than eighth, column.
- F. The difference in tab handling should not affect the running of a program, but it may cause erratic spacing in program listings.
- G. If you inadvertently edit a Basic file with tabs in the output file, you can run a dummy edit with blank packing turned off to convert the tabs back to blanks.
- H. In Basic, a linefeed entered by itself is stored as an LFCR to distinguish it from a CRLF, which indicates the end of a logical line. The Line Feed control key is recog-

nized by EDIT for this situation.

- I. **Note:** You must be sure to follow the linefeed character immediately with a carriage return to remain compatible with Basic usage.
- J. EDIT does not prevent you from including duplicate, out-of-sequence or unnumbered lines; and you could make these errors. If you do, the program may not load properly. If this happens, list the code that loaded successfully to find the area in which the error occurred, then re-enter EDIT to make corrections.

DATA FILES

- A. You may use EDIT to create or modify data files in text format (such as Basic INPUT/PRINT) as you would any other text files.
- B. You may also edit files with fixed-length records (such as Basic GET/PUT requires) if they contain only the Ascii characters described above.
- C. For fixed-length files, you should set the line width to an even divisor of the record length, e.g., 128-byte records occupy two lines of 64 characters (L64).
- D. Mode settings for fixed length files should be the same as for file type ASC — MAN, MBN, and MCN. (Blank Packing *must* be suppressed.)
- E. Since EDIT does not limit the length of a line of text, you may create a fixed-length file without a single carriage return.
- F. **Note:** If you include carriage returns, remember that the machine inserts a linefeed character following the CR (unless you precede it with an LF) and consequently occupies *two* spaces in the fixed record.

MISCELLANEOUS CONSIDERATIONS

- A. If you want to locate a string at the start of a line, e.g., a line number or variable name, precede the string with a carriage return (i.e., by pressing the TAB key).
- B. The machine now goes directly to the line that begins with the string without encountering every reference to that string within the source program.
- C. Example: The sequence (SEARCH Key, TAB Key, A, B, C, RETURN) locates the program line that defines ABC, and the cursor is positioned at the start of the line, rather than at the carriage return on the preceding line.
- D. You may use the Include command to merge part of one program with another.
- E. If you have not formatted your program file as an INCLUDE file, you must *page* through the file screen by screen until you reach the section you wish to include.
- F. You may need to include some additional lines from the program in order to get all that you need.
- G. You can easily delete the extra lines with either the line or block delete functions.
- H. If you have routines that you frequently include in programs, you should probably create a library file of the routines by inserting banners at the beginning of each routine, so that you may include routines by name just as in boilerplate text applications. (See the Section on the INCLUDE command, in the EDIT notes.)

GLOSSARY

ASCII — (ASK-key). American Standard Code for Information Interchange, the standard character set used by your terminal and printer. It includes letters, numbers, punctuation, and some other useful symbols, as well as a few special "characters", like the carriage return and line feed, that are found only in computer environments. Like the alphabet, the Ascii character set has a standard order that applies when comparing characters in an IF statement. A blank is a separate character and is the first, or lowest, displayable character in the set. The digits zero through nine, the upper case alphabet, and the lowercase alphabet follow in sections of consecutive characters, with the remaining symbols and punctuation marks interspersed between those sections.

BUFFER — Memory reserved for a specific purpose. Although both EDIT and PRINT use many buffers internally, PRINT has three buffers that you can change, i.e., the heading, footing and variable buffers. (See section on Adjusting Internal Buffers in the notes on PRINT)

CONSTANT — A numeric value that does not change, i.e., a number. You may assign constants as the current value of numeric variables, use them as part of a numeric expression, or compare them to numeric variables or expressions.

CURRENT DRIVE — The default disk drive when you do not include the drive as part of the FILENAME. Usually, DRIVE A is the current drive. You may change the current drive while in CP/M. (See CP/M for MAGIC WAND Users)

DEFAULT VALUE — The value that the machine assumes if you do not specify a value. For example, the default value for the RM command is 60, i.e., if you do not issue a RM command, the machine establishes a right margin of 60 characters.

ELEMENT — A single piece of information from a data file record. If the element is from a text file, it is a logical line, i.e., any number of characters ending in a carriage return. If the element is from a fixed file, it has a fixed number of characters as defined in the DATA statement. With a DATA statement you assign an element as the value of a variable, or you can skip it without assigning it to any variable.

FIXED DATA FILE — A data file in which each record contains a fixed number of characters, rather than a number of text lines. Each element within a record must be a fixed size

also, instead of a text line whose length can vary. You can create a fixed data file with EDIT, but it is more often the product of a separate program or system. A fixed file is more likely to contain elements composed of non-Ascii characters which cannot be processed by either EDIT or PRINT. To define such a file, you must know the exact size, location, and format of each element to be used in the document.

INCREMENT — The minimum physical movement that a given printer can perform. For a standard printer (or a specialty printer operating in DRAFT mode), the increment is fixed horizontally at the width of one character (typically 10 or 12 per inch) and vertically at 6 or 8 lines per inch. Specialty printers allow more refined movements for greater control and higher print quality. Typically, they allow a horizontal increment of 1/120th of an inch, and a vertical increment of 1/48th of an inch.

LIBRARY FILE — A file containing standard text or routines for inclusion in other files. Normally, you would use a library file with the INCLUDE function in EDIT to create boilerplate documents. (See Lesson Three)

LITERAL EXPRESSION — A string value that does not change. Always enclose a literal expression in quotation marks. You may use either single or double quotation marks (' ' or " "), as long as the second quotation mark is the same as the first. This allows you to include a double quotation mark in a literal expression. You may assign a literal expression as the current value of a string variable or print it on the screen (in a SHOW command) or on the printer (in an equals quote command).

NUMERIC VARIABLE — A variable whose value is treated as a number. You may use a numeric variable any place that you can use a constant. You must use a numeric variable when comparing or assigning a numeric variable, constant or expression.

PASS — One processing of a file. You process most of your files in just one pass. If you are printing form letters using a data file, the machine processes, or "passes", the file for each record in the data file. If you are printing multiple copies of a document, each copy is processed in a separate pass. The %PASS system variable gives you the current pass number.

RECOGNITION CHARACTER — A single character designated as a marker to represent some processing function. For example, the at-

sign (@) is the default recognition character for the boldface function. When the machine encounters an at-sign in the text, it treats it as a signal to begin (or stop) printing in boldface. The machine does not print the at-sign as text. Because it is sometimes necessary to print a character that is normally used for a special function, all recognition characters may be undefined or redefined.

RECORD — The primary subdivision of a data file. A record is made up of elements. If the data file is in text format, a record consists of a specific number of lines of text defined in the FILE statement. Each line, or element, may contain any number of characters ending in a carriage return. If the file is in fixed format, each record contains the number of characters specified in the FILE statement. Each element in a fixed record is defined in the DATA statement as a specific number of characters at a fixed location in the record. The %REC system variable stores the current record number.

SPECIALTY PRINTER — Printers that can move horizontally and vertically by small steps, or increments, of less than a full character or line width. Although generally slower and more expensive than standard printers, the incremental movement makes possible boldface, sub/superscripting, more even justification of text, and variable and/or proportional pitch.

SPOOL PRINTING — Printing a file while the computer is performing another function. While in EDIT, you can spool print any formatted print file while continuing to edit another file. (Also called printing in background)

STANDARD PRINTER — Virtually any printer that can be connected to a small computer can function as a standard printer. It needs only to print the Ascii character set at a fixed number of characters and lines per inch, and to recognize a carriage return and line feed as two independent characters. Standard printers, or specialty printers operating in DRAFT mode, cannot perform the boldface, sub/superscripting, and variable pitch functions.

STRING — A series of Ascii characters without any inherent meaning. A string may be of any length containing any Ascii characters in any order. Letters, numbers, spaces, punctuation and special characters contribute equally to the length of a string. When the machine looks at a string, it only sees a series of characters. If there is some significance in the string, you must supply it yourself. For one string to be identical to another string, it must contain exactly the same characters in exactly the same order.

TEXT DATA FILE — A data file consisting of lines of Ascii characters. A line is any number of characters (or none) followed by a carriage return. A record consists of the specific number of consecutive lines defined in the FILE statement. Each line is an element in that record. Although a line may be any length, the machine can assign only the first 55 characters as the value of an element. A text data file may be created and maintained with EDIT or by a separate program.

VARIABLE — A function that you define that can have more than one value. The machine stores the current value of the variable in memory, and you can retrieve and use that value as often as you wish. Although the current value is stored as a string of characters, you may use it in various ways. (See Lessons Six, Seven and Eight and the Notes on PRINT)

VOID SPACE — A blank area on the screen that does not correspond to the text file, i.e., the text file does not contain blanks at that location in the file. Void space can occur in three circumstances: to the right of a carriage return, to the right of the cursor on an incomplete line, and to the right of the last word on a line following wordsplitting. You can move into and out of void space as if it were part of the text itself. The action of some commands are slightly affected if the cursor is in void space, but the differences are barely noticeable.

INDEX*

- BDOS errors 79f
- Block markers 10,90
- Block copy 67, 90
- Block move 11, 90
- Block delete 11, 91
- Bold face (BF) 38, 104
- Bottom margin (BM) 69, 99
- Buffers, adjusting 120
- Centered format (CENTER) 100
- Centered line (CTR) 30, 100
- Clear Screen (CLS) 48, 122
- Conditional commands (See IF)
- Command marker (CMD) 102
- Copies, multiple (COPY) 25, 106
- Copying a file, CP/M (PIP) 81
- Current drive, changing, CP/M 80
- Cursor movement 4, 83
- DATA 52, 110
- Delete, character 5, 85
- Delete, line 9, 85
- Delete, block 11, 91
- Directory of a disk, EDIT (See files)
- Directory of disk, CP/M (DIR) 80
- DISK ON 75, 119
- Disk status (See STAT)
- Display buffers (DB) 122
- Display file variables (DF) 122
- Display status (DS) 24, 122
- Display variables (DV) 122
- Disk full 94
- Displaying a file, EDIT 18, 91
- Displaying a file, CP/M (TYPE) 82
- Dollar (decimal) comparisons 62, 117
- DRAFT 105
- END, EDIT (See Exiting from EDIT)
- END, PRINT 57, 60, 107
- Erasing a file, EDIT 94
- Erasing a file, CP/M 80
- External data files (FILE) 51, 109
- Exiting from EDIT 6, 19, 94
- Files command 18, 93
- Footings (FOOT) 73-74, 119
- Form feed (FF) 106
- Form feed, mechanical (FORMFEED OFF) 68, 106
- Form type (single or continuous) 105
- Formatting features, EDIT 94, 123
- FSIZE 120
- GET 43, 108
- Ghost hyphen (HY) 35, 103
- Headings (HEAD) 72, 74, 119
- Horizontal movement by increments (H) 101
- HSIZE 120
- IF statement 59, 115
- Ignore character (IGNORE) 64, 104
- Include command 19, 91
- Indentation (IN) 27, 98
- Insert, character 5, 85
- Insert, full 10, 85
- Justified text 29, 99f
- Kerning (K) 101
- Left flush (LEFT) 27, 99
- Left margin (LM) 26, 98
- LINE command 105
- Line feed 87
- Line length, EDIT 4, 89
- Line length, PRINT (See Right Margin)
- Lines per inch (LPI) 101
- Literal format (LIT) 30, 100
- Microsoft Basic 123f
- Memory overflow 88
- Multiple conditions 118
- Naming a file 83
- New line (NL) 47, 59, 106
- New page (NP) 46, 72, 105
- NOTE 34, 121
- Numeric comparisons 116
- OUT 104
- Page feed 87, 123 (Also see NP)
- Page length 68, 99
- Page number, changing (PG) 70, 106
- Paragraph indentation (PI) 28, 98
- Pitch (CPI) 41, 100
- PRINT OFF 75, 119
- Printing in EDIT 4, 93
- Printing to disk (See DISK ON)
- Proportional printing 102
- QUIT, EDIT (See Exiting from EDIT)
- QUIT, PRINT 107
- Read command 88
- Recognition characters 40, 102
- Renaming a file, CP/M (REN) 81
- Right flush (RIGHT) 27, 99
- Right margin (RM) 26, 98
- Scrolling 7, 84
- SET 48, 108
- SETUP 55, 106
- Search and replace 12-14, 86-87
- SHOW 48, 121
- SKIP 63, 117
- SKIP TO 64, 118
- Spacing (SP) 30, 101
- START/STOP 57, 77, 107
- STAT, CP/M 81f
- String comparisons 116
- Sub/superscripting (SSB,SSA) 39, 104
- Tab, PRINT 105
- Tab setting, EDIT 90

TEXT 45, 55, 106

Top margin (TM) 69, 99

Turning off printer (See PRINT OFF)

Underscoring (UN) 37, 103

Variables, general 42, 107

Variables, colon 44, 111

Variables, dollar 49, 112

Variables, equals 111

Variables, length 60, 114

Variables, number 61, 113

Variables, numeric 61, 112

Variables, string 111

Variables, system 56, 115

%COL 115

%EOF 65, 115

%LINE 114

%LINES 71, 114

%PAGE 70, 114

%PASS 114

%REC 56, 114

Vertical movement by increments (V) 102

VSIZE 120

WAIT 47, 121

Write command 89