Simply Better

You can PAY more for a CoCo3 word processor, but will it...

- Quickly create a **second word processor** in a **second window on the same screen**, and allow you to **work on two files at the same time**?
- **Automatically save to disk** as you work, so you will lose little work if computer loses power?
- Create a **table of contents** or **index** by simply marking items you wish to be included, and letting your computer do the rest?
- Pop up a handy **4-function calculator**?
- Auto-**spool printing** and return **use of your computer quickly** while printing?
- Fill in blanks on **pre-printed forms easily**?
- **Show onscreen fonts and underlining** with your various printer fonts displayed as different colored text, instead of cryptic codes?
- Allow fast typists to **type ahead without losing characters**, with a keyboard buffer?
- **Find & replace single or multiple strings** of text, using wildcards, and even being able to specify **fonts** and **underlining** to replace?
- Perform user-definable **serial-numbering**?
- Allow you to choose to **print all, odd or even pages** for easy double-sided printing?
- Create & print **French-language characters**?

All these special features, plus dozens more, at half the price of competing wordprocessors! Simply Better is a package that was designed to be easy for a novice to learn, yet powerful enough to satisfy the needs of even the most advanced user!

Distributed exclusively by:

COCOPRO!

1334 Byron Avenue
Ypsilanti, MI 48198
(313) 481-DAVE (3283)

Simply Better

Contents

Section 1- Learning Simply Better "hands-on"

USING THIS MANUAL 1
BEFORE YOU BEGIN 1
CONVENTIONS USED IN THIS MANUAL 2
INFORMATION ON CONFIGURATION 3
CONFIGURING YOUR SYSTEM 3
GETTING STARTED 10
BASICS OF TEXT ENTRY 11
INTRODUCING NEW KEYS 16
MOVING ABOUT IN YOUR TEXT 18
THE TAB KEY AND TAB STOPS 19
INSERTING TEXT 20
DELETING TEXT 20
UNDERLINING AND PRINT FONTS 22
WORKING WITH BLOCKS OF TEXT 27
PRINT PARAMETERS 30
IMBEDDED MARKERS 37
THE VIEW MODE 40
PRINTING TEXT ON PAPER 42
FINDING AND REPLACING TEXT 43
LOADING AND SAVING TEXT FILES 47
CREATING, USING, AND SAVING TASKS 52
NEXT NUMBER OUT 55
SORTING SECTIONS OF TEXT 57
THE "WINDOW" - A SECOND WORD PROCESSOR 59
AUTOMATIC PRINT SPOOLING 62
PRINT FILLING A FORM 63
CREATING A TABLE OF CONTENTS 65
CREATING AN INDEX 67
PERFORMING MAIL-MERGES 68
CRAY-O-LATOR "CALCULATOR" 70
GETTING HELP INSTANTLY 73
EXIT TO BASIC 73

Section 2- Expanded Command Reference

COMMAND MODE COMMANDS 74
TEXT MODE <CTRL> COMMANDS 83
TEXT MODE <F1> COMMANDS 94
PRINT PARAMETERS 99

Section 3- Condensed Command Reference

COMMAND MODE COMMANDS CONDENSED 103
<CTRL> KEY COMMANDS CONDENSED 104
<F1> KEY COMMANDS CONDENSED 106

APPENDIX A: Standard and Alternate Keys 107
APPENDIX B: System Errors Explained 109
INDEX 114-117

Section 1

Learning Simply Better "Hands-on"

USING THIS MANUAL

SECTION ONE:
This section was designed to teach you, in a "hands-on" way, about the program. It will aquaint you with every feature of the program and will cover every command available. It also covers how to initially set up (configure) the program for your own system. If you skip over this section, you may never really be able to use SIMPLY BETTER to its full potential.

SECTION TWO:
This section was designed to be used as a reference for each of the various commands. It was not designed to teach you about the program. After you have completed "SECTION ONE", then you can use this section to look up a specific command.

SECTION THREE:
This section was designed as a quick reference for each of the commands. The information in this section is also on your disk in a file named "POCKET.CRD". This file can be loaded and set up however you want. Print the file out on paper and use it as a quick reference "pocket card".

BEFORE YOU BEGIN

DO NOT use your original disk except to make backup copies. Store the original disk where it will be safe, and use the backup copy as a working disk. To backup the original, start with a clean formatted disk. (Use the DSKINI command to format a disk.)

SINGLE DRIVE USERS:
If you only have one disk drive, place your original program disk in Drive #0 and key in the command...

BACKUP <ENTER>

Then follow the instructions as they are shown on the screen.

MULTI-DRIVE USERS:
If you have more than one disk drive, first, place the original program disk in Drive #0. Next, place the formatted disk in Drive #1 and key in the command...

BACKUP 0 TO 1 <ENTER>

When the backup is complete, remove the new copy for use and place the original disk away in a safe place.

CONVENTIONS USED IN THIS MANUAL

DISK FILE NAMES:
In this manual, the word "filename" refers to the name of a disk file, its extension, and the drive number for the file.

REPRESENTING KEYBOARD KEYS:
Throughout this manual are lines of text for you to key in. Some of the keys can not be represented by single characters. These keys will be shown using capital letters and will be surrounded by "less than" and "greater than" symbols. Each key is listed and described below...

<ENTER>	the ENTER key	<CLEAR>	the CLEAR key
<ALT>	the ALT key	<CTRL>	the CTRL key
<F1>	the F1 key	<F2>	the F2 key
<BREAK>	the BREAK key	<SHIFT>	the SHIFT key
<R-ARW>	the ➡ key	<L-ARW>	the ⬅ key
<U-ARW>	the ⬆ key	<D-ARW>	the ⬇ key

Consider this example...

This is a single line. <ENTER>

If you were to key in this line, (with the program up and running) you would type each character except the <ENTER>. Treat the <ENTER> as a single character by pressing the <ENTER> key.

Any time that you are to hold down the <SHIFT> key while pressing a second key, the <SHIFT> key and the second key will be separated by a dash as shown here...

<SHIFT>-=

Here, the <SHIFT> key is to be held down while pressing the "=" (equals) key. Here's another example using the <SHIFT> key...

Going to the top <CTRL> <SHIFT>-<U-ARW>

To key this line, type the characters "Going to the top" in. Then press the <CTRL> key once. Then hold down the <SHIFT> key and press the <U-ARW> key.

IMPORTANT: Notice that there is a "space" between <CTRL> and <SHIFT> in the sample line above. When a key like <CTRL> or <ENTER> is shown, for clarity purposes I will include a "space" both BEFORE and AFTER it! The "space" is only for clarity and is NOT to be keyed as part of the line.

INFORMATION ON CONFIGURATION

Realizing that most of us have different printers, different monitors, and different tastes, SIMPLY BETTER includes a program that will enable you to set up your word processor to your own likes and dislikes.

SIMPLY BETTER was designed so that many of its parameters are grouped into one area at the end of the program. This area can be thought of as a data module. This data module can be replaced by other data modules that may be set up for different printers, monitors, etc..

On your disk is a program named CONFIG.BAS. This program is used to create new data modules. The new data module can then be saved to disk, or it can be used to replace the data module at the end of SIMPLY BETTER. Each time you use CONFIG.BAS, it will ask for the name of a module to load and use as initial data. The first time you use CONFIG.BAS, select a module name according to the information listed below...

Module Name	Comments
GRN128.BIN	Use with green screen monitors
AMB128.BIN	Use with amber screen monitors
RGB128.BIN	Use with color monitors

Later, you can use the name of a different module that you have saved to disk. Set up each module to your own likes. After the module has been modified, you'll have the option of saving it as a module to disk, or using it to replace the module at the end of SIMPLY BETTER.

Each module occupies only 1 granule on a disk. Thus, you can create and save numerous modules (as versions) on a single disk.

The section entitled "GETTING STARTED" will cover booting the program using the default data module that is part of the program, and it will cover booting the program using one of the modified data modules that has been saved to disk.

CONFIGURING YOUR SYSTEM

To configure your system, place the disk that has CONFIG.BAS and C.BIN in Drive #0 and key in...

RUN "CONFIG" <ENTER>

You will be prompted for the drive that contains SIMPLY.BIN. Key the number for the drive.

Next, you will be asked for the name of a data module to use for its initial set of data. If you have saved a module prior to this, you can give it the name of the one saved. If you have not, then you must give it the name of one of the default data

modules supplied on your disk. Use the module RGB128.BIN if you are using a color monitor; use the module AMB128.BIN if you have an amber screen monitor; use the module GRN128.BIN if you have a green screen monitor. Type in the name of the module chosen and press <ENTER>.

Next, there will be three screens containing parameters that you can modify. You can print (dump) any screen to the printer by keying...

<SHIFT>-*

THE FIRST SCREEN:

```
SIMPLY BETTER Configuration Program
128k        (C) 1988 By D.M.Rickert

Pri-buffer: 49152        Alt-buffer: 32768
Screen size: 40          Color: ON
Tracks: 35               Auto: OFF
Text width: 39           Wrap: ON
Mask: OFF                Verify: ON
Beep: ON                 Eyes: OFF
Number: 1                Increment: 1
Insert/Overstrike: O

Up/Down arrows go to a new item
Shift-Up/Down arrows go to new screen
Left/Right arrows change/toggle values
Shift-Right/Left arrows inc/dec more
Use <BREAK> to exit
```

Notice on your screen that the number after "Pri-buffer:" is both blinking and in reverse color. Parameters that can be modified are shown in reverse color. The item that is to be modified will be the one blinking. If the item blinking is a left arrow "<-", then the item to be modified will be a string of characters or numbers to the left.

Use the <D-ARW> to advance to the next item, and use the <U-ARW> to move to a previous item. If you press the <D-ARW> key while the "Pri-buffer:" value is blinking, you will advance to "Screen size:" and the "40" (or "80") will be blinking.

You can change the value of a parameter by keying the <R-ARW> or the <L-ARW>. If the item toggles between two or more values, such as "Y" for "yes" or "N" for "no", then keying either the <R-ARW> or the <L-ARW> will change the parameter to the next value. If the parameter increases and decreases in value, key the <L-ARW> to decrease it, or key the <R-ARW> to increase it.

While the "40" is blinking, key either the <R-ARW> or the <L-ARW>. Immediately the screen changes from a 40 column screen to an 80 column screen and the "40" becomes an "80". Pressing the <R-ARW> or <L-ARW> again will return you to the 40 column screen.

If you key <SHIFT>-<D-ARW>, you will advance to the next screen of parameters. Keying <SHIFT>-<U-ARW> will return you to the previous screen. You can now move from screen to screen easily.

If the item blinking is a left arrow "<-", pressing the <L-ARW> will give you a "<" cursor at the left. You can then key in new letters or values for that line. Items preceded with "On:" or "Off:" are for the numbers that must be sent to your printer to turn "On" or "Off" that particular print font. Numbers must be separated by commas and hexidecimal numbers may be used if they are preceded by a "$" dollar sign. Look at the example below...

```
On: 27,34      <-
Off: $1b,$23   <-
```

Here, the numbers 27 and 34 are the values to be sent to turn on the font, and the hexidecimal numbers $1B and $23 (decimal 27, 35) are to be sent to turn off the font.

While keying in a line, you can key <BREAK> to abort the entry. The <L-ARW> will back up one character. Pressing <ENTER> will terminate entry and those characters to the left of the "<" cursor will remain. Any characters at or to the right of the "<" cursor will be omitted. Some items have large number ranges. For example, the page number (PN) on the third screen can be any number from 0 to 65,535. To help you move faster through such numbers, holding down the <SHIFT> key while pressing the <L-ARW> or <R-ARW> will decrement or increment the item by larger amounts.

ON THE FIRST SCREEN:

PRI-BUFFER: This number is initially 49152. It is the "primary buffer" size for 128k systems. If you have 512k of memory, while this number is blinking, press the <ALT> key and you will toggle from 128k to 512k and the "128k" in the title will read "512K". DO NOT USE 512k IF YOU ONLY HAVE 128k! There are two areas in memory (buffers) for text storage, a "primary buffer" and an "alternate buffer". These are covered in detail later. Here I'll only mention that you can select the size of memory for each buffer. The amount of memory taken from one buffer will automatically be given to the other buffer.

SCREEN SIZE: You can select between either the 40 column text screen or the 80 column text screen.

COLOR: If you select "OFF", the color burst on the computer's composite output is turned off. If you are using a monochrome monitor, you may want to experiment with this.

TRACKS: Set this to 35 if your disks are formatted for 35 tracks. Set it to 40 if your disks are formatted to 40 tracks.

AUTO: If you select "ON", your text will automatically be saved to disk if no key is hit for a period of 5 minutes. Read the section under "THE AUTO SAVE FEATURE" before using the "ON" setting.

TEXT WIDTH: This is the "screen" text width. It is not the printed text width. At first, use a value of one less than the "Screen size" that's selected (39 or 79).

WRAP: If word wrap is "ON", words that extend beyond the "width" setting will not be cut off. Instead, the entire word will drop down to the start of the next line.

MASK: When the mask is "ON", searches for characters using the "locate" feature will mask the letters from A to Z so that it will find both upper- and lower case letters.

VERIFY: When "ON" is selected, many commands will ask "Are you sure?" before they execute. Some always do.

BEEP: This turns the sound "ON" or "OFF" for key beeps, alert beeps, and error beeps. EYES: If you have a hearing impairment, you can select "ON" here and the screen border will flash at times when the error beep or alert beep would sound.

NUMBER: This is the value of the first number to be used as the "next-number-out" with the <CTRL> N command. You may want to wait until you have read section "NEXT NUMBER OUT" before changing this parameter.

INCREMENT: This number will be added to or subtracted from the number at each "next-number-out" with the <CTRL> N command. See "NUMBER" above.

INSERT/OVERSTRIKE: Choose either the insert mode or the overstrike mode.

THE SECOND SCREEN:

Now comes some of the real fun! This screen is where you choose the display colors for your normal text and background, the 5 print fonts, and the colors for blocks of text. To get to this screen, use either the <D-ARW> until you pass the last item on the first screen, or key <SHIFT>-<D-ARW>.The second screen is shown below (the values for the parameters may be different than what is on your screen)...

```
SIMPLY BETTER Configuration Program
 128k      (C) 1988 By D.M.Rickert

Normal text      63 Normal background    9
Block text        0 Block background     18
Condensed-1      <- Text63      Background33
On:27,20                                <-
Off:27,19                               <-
Elongated-2      <- Text8       Background59
On:27,14                                <-
Off:27,15                               <-
Bold-3           <- Text36      Background7
On:27,31                                <-
Off:27,32                               <-
Compressed-4     <- Text25      Background1
On:27,23                                <-
Off:27,19                               <-
Italics-5        <- Text23      Background2
On:27,66,1                              <-
Off:27,66,0                             <-
Underline on :15                        <-
Underline off:14                        <-
```


NORMAL TEXT and NORMAL BACKGROUND: Select colors for normal text and normal text background.

BLOCK TEXT and BLOCK BACKGROUND: Select colors for block text and block text background.

After these items are 5 groups of items. Each group is for an individual print font. Since each group is modified in the same way, we will only describe how to modify one of them.

The first item of each group is the descriptor for that particular print font. Move down until the "<-" to the right of the descriptor is blinking and then press the <L-ARW>. Now key in up to 15 characters that describe the print font for this group. You can abort the entry by keying <BREAK>. Press <ENTER> when you have the descriptor as you want it. Notice that our descriptors are followed by a dash and then a number from 1 to 5. Later, this number will indicate which key must follow the <F1> key to begin printing in that particular font.

You can use the <L-ARW> to backspace. Text at and to the right of the "<" cursor will be omitted when you press <ENTER>. Use <BREAK> to abort.

The next item in the group is the text color for this font. This is followed by the text background color for the font. I have found that it is easier at times to select the text background first, and then select the text color. You can make the background the same as the normal text background and only change the text color if desired. It's all up to you.

After you select the text color and its background color, move down one to the next "<-" (below and to the right). The initial values here are the numbers for a Radio Shack DMP 132 printer. Press the <L-ARW> and enter the numbers that must be sent to your printer to turn on this print font. You can enter numbers in decimal or precede hexidecimal numbers with the dollar sign. You must separate numbers with commas. After entering the "On" codes, move down one and do the same for the codes that will turn off this print font.

The last two items on the screen are the codes necessary to turn underlining on and off. Set up these values as required by your printer. THE THIRD SCREEN:

This screen contains print fonts and print modes. The screen is shown below...

```
SIMPLY BETTER Configuration Program
 128k      (C) 1988 By D.M.Rickert

AL 0    AS A    BD 0    BL 0    BM 60
BS A    FL 0    FS A    HL 0    HS A
IC 0    IP N    JN 2    JS N    JU L
LF N    LM 8    LS 1    NP N    NU 0
PA N    PC N    PF N    PH Y    PL 66
PN 1    PP A    RM 72   TC 0    TM 6
Tab: 1 C
```

Each parameter and its range in "( )" is shown below...

AL: Auxiliary line A line number (0 to "PL")
AS: Auxiliary line A status (AEO)
BD: Baud rate (0 to 6)
BL: Auxiliary line B line number (0 to "PL")
BM: Bottom margin ("TM"+1 to "PL")
BS: Auxiliary line B status (AEO)
FL: Footer line number (0 to "PL")
FS: Footer status (AEO)
HL: Header line number (0 to "PL")
HS: Header status (AEO)
IC: Index column number (0 to 230)
IP: Invisible print (YN)
JN: Justify space number (0 to 10)
JS: Justify spaces (YN)
JU: Justification (LRCB)
LF: Line feeds (YN)
LM: Left margin (0 to "RM"-1)
LS: Line spacing (1 to 10)
NP: No-prints printed (YN)
NU: Nulls (0 to 230)
PA: Pagination (YN)
PC: Print comments (YN)
PF: Print fill form (YN)
PH: Printer handshake (YN)
PL: Page length ("BM" to 255)
PN: Page number (0 to 65,535)
PP: Print pages (AEO)
RM: Right margin ("LM"+1 to 250)
TC: Table of contents column number (0 to 230)
TM: Top margin (0 to "BM"-1)

An item in between quotes in the range is one of the parameters listed to the left.
(AEO) is All, Even or Odd
(YN) is Yes or No
(LRCB) is Left, Right, Center, or Both

EXITING AND SAVING YOUR MODULE:

After you have all the parameters as you want, press the <BREAK> key and exit. You will then be asked...

Save, Append, Quit (S/A/Q)?

If you DO NOT want to save your changes, key...

Q <ENTER>

If you DO want to append this module to SIMPLY.BIN, key...

A <ENTER>

If you want to save it as a data module, key...

S <ENTER>

If you key "s", you will be asked for a name for the module. You may then save the module as a new module on disk by entering a new module name, or you can save it back as the first module that you loaded by just keying <ENTER>.

GETTING STARTED

To boot your program, first place your copy of Simply Better in Drive #0. You must run the "BOOT.BAS" program to boot Simply Better. With the disk in Drive #0, key in the command...

RUN "BOOT" <ENTER>

You will then be given four options...

1. Pressing <ENTER> will boot SIMPLY.BIN using the data module that is part of SIMPLY.BIN.

2. Key in the filename of a data module on the disk that you want to use in place of the one that is part of SIMPLY.BIN.

3. Press <SHIFT>-? followed by <ENTER> to view the disk's directory.

4. Press <BREAK> to abort.

After making your selection, the program will boot in the way specified unless you keyed <BREAK> to abort.

If you have not yet modified a module for your system, select a default module according to the information below...

Module Name	Comments
GRN128.BIN	Use with green screen monitors
AMB128.BIN	Use with amber screen monitors
RGB128.BIN	Use with color monitors.

Each of these modules are set up for 128k systems. If you have 512k of memory, use the CONFIG program (covered in above) to set up and save a module configured for 512k.

BASICS OF TEXT ENTRY

When you first boot your copy of SIMPLY BETTER, the top of the screen will look as shown here (the rest will be blank)...

You always have two different size screens available. You can use the 40 column screen with large text characters, or you can use the 80 column screen with smaller characters. To change screens, you must be in the COMMAND mode. If you have just booted (executed) the program, then you're in the COMMAND mode. On the command line, key...

40 <ENTER>

to change to the 40 column screen, or key...

80 <ENTER>

to change to the 80 column screen.

THE COMMAND MODE:
This mode is for entering commands that are words, numbers, or the combination of both.

THE TEXT MODE:
This mode is where you will type in text . Simply use your keyboard as a typewriter and type in your text. There are numerous commands that can be keyed while in the TEXT mode and each will be covered in detail.

THE VIEW MODE:
Use this mode to view your text on the screen as close as possible to how it will look when it is printed.

WHILE IN THE COMMAND MODE:
The blinking square after the word "PriCmd:" is the cursor. The cursor indicates where the next character keyed will be placed. The cursor is non-destructive and is not part of your text. So that we are all looking at similar screens, key in the line below...

WIDTH79 <ENTER>

When you do, the screen will display the new width, and will not print the "PriCmd:" prompt until the next key that is part of a command is pressed. Now key in...

80 <ENTER>

This changes the screen to 80 column format. If you want to have "key-beep", you can key in the command...

BEEP <ENTER>

If the screen reads "BEEP ON" then "key-beep" will now be on. If it reads "BEEP OFF", key in the command again.

If you have a hearing impairment, and would like the border to flash when the program issues an error or an alert, use the command...

EYES <ENTER>

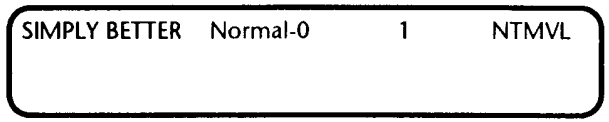
If the screen now reads "EYES ON", then the border is set to flash. If the screen reads "EYES OFF", enter the command again.

You exit the COMMAND mode to the TEXT mode by keying <BREAK>, <R-ARW>, or <D-ARW>.

Key the <BREAK> key now and go to the TEXT mode.

THE TEXT MODE SCREEN:

When in the TEXT mode, the screen looks similar to what is shown below...



"SIMPLY BETTER" is the name of the program. When you key many of the <CTRL> and <F1> commands, the top line will display the name of the command keyed where the words "SIMPLY BETTER" are. When you press another key, the words "SIMPLY BETTER" will return.

The "Normal-0" indicates what the current print font is. This will be the font of the character just before the cursor. In this way, if the previous character was font #2, then the next character you type will also be font #2. To the far right of this is a "1". The "1" indicates the screen column number for the cursor position. This number will change as the cursor position changes.

At the far right will be from 2 to 5 letters. If a letter has been replaced by a "space", then that particular item is "off".

The first letter is the key status. If the letter is an "N", the next key you press will output a normal text character. If you key <CTRL>, the letter will change to a blinking "C". This indicates that the next key you press will act as a <CTRL> command, and not a normal character. If you key <ALT>, the letter becomes a blinking "A". This indicates that the next key hit will output an "alternate" character. If you press the <F1> key, the letter becomes a blinking "1" to signify that the next key hit will be a <F1> command or character.

Next to this is the letter "O". This indicates that you are in the "overstrike" mode. In the "overstrike" mode, text typed in will replace the text at the cursor position. If the letter here is an "I", you are in the "insert" mode and text keyed in will be inserted into the text to at the LEFT of the cursor. You can change between modes keying...

<CTRL> I

The next letter will be an "M" if the mask is on. If the mask is on, the "locate" will

treat upper and lower case letters as if they were the same letter. If the mask is off, upper and lower case letters must match exactly. Turn the "mask" on or off by keying...

<CTRL> <SHIFT>-=

Next to this will be a "V" if the verify is on. If on, many of the commands will ask "Are you sure?" before they are executed. You can turn the verify on or off by keying...

<ALT> <CTRL> V

The last letter is either an "L" or a "U" and indicates the status of the shift-lock key. If an "L" is present, then lower case letters are output (unless you hold down the <SHIFT> while pressing a letter). If the letter is a "U", then the <SHIFT> key is "locked down" and all letters output will be in upper case. You "lock" or "unlock" the <SHIFT> key by keying...

<SHIFT>-0

After keying <BREAK> from the COMMAND mode, the cursor will initially be at the first text line on the screen. While in the overstrike mode, key in the following...

My first text line.

If you make a mistake, use the <L-ARW> or <R-ARW> to position the cursor over the letter you wish to change and key the correct letter. When you have the line keyed in correctly, place the cursor just after the period and key <ENTER>.

When you keyed the <ENTER> key, two obvious things occurred. First, the cursor dropped down to the beginning of the next line, and second, a small underline character was left on the screen after the period. This is the "carriage return" character. Treat it as a REAL character. You can delete it, insert it, and even send it to the printer.

Move the cursor to the beginning of the line that you just entered by pressing the <U-ARW> twice. (The reason that it must be pressed twice will be explained later.) With the cursor on the "M", key in...

An accident

Your screen should now read...

An accidentxt line._

Now press the <BREAK> key. What happened to "An accident"? Keying <BREAK> performs one of the "oops" commands.

To better understand how the "oops" works, you need to better understand how text is stored by the program. Text is stored in what are called buffers. A buffer is simply an area in the computer's memory designated for text storage. SIMPLY BETTER uses a "primary buffer" to store all of your text, and it uses a "current line buffer" to store the text of the line that the cursor is in. As you key text into a line, the text is stored in the "current line buffer". If the cursor moves to a new line, text from the "current line buffer" will then replace what was previously stored in the "primary buffer" for the line. If you key <BREAK>, the program will replace the text

in the "current line buffer" with what was previously stored in the "primary buffer" for this line. In the example above, the line "My first text line." had been previously stored in the "primary buffer". When you went back to the line with the cursor, the program moved the line from the "primary buffer" back into the "current line buffer". You then typed in "An accident". This was not moved to the "primary buffer", so when you keyed <BREAK>, the old line from the "primary buffer" was again moved to the "current line buffer" and the screen was redrawn to show the old line.

There are several things that can cause the text in the "current line buffer" to be moved to the "primary buffer". Pressing <ENTER> is just one of the ways. While inserting text in a line, if the text at the end of the line gets pushed off the right side to the next line down, the program will move the "current line buffer" to the "primary buffer". If you press the <U-ARW> or the <D-ARW> after making any changes in the line, the "current line buffer" will be moved to the "primary buffer" and the cursor will not go up or down a line until the next <U-ARW> or <D-ARW> is keyed.

Place the cursor at the beginning of a new (empty) line and key in the following line...

The quick fox jumped. <ENTER>

If the line should have read "brown fox", change to the insert mode and insert the word "brown" plus a <SPACE> ahead of the word "fox". To do this, key...

<CTRL> I

This will put you in the insert mode. Check the top right on your screen for the "I". If there's an "O", repeat the sequence again .

Move back up to the line just entered. Use the <R-ARW> or the <L-ARW> to move the cursor to where the word is to be inserted. Remember, text is inserted to the left of the cursor, so place the cursor over the "f" in "fox". Now key in...

brown <SPACE> <U-ARW>

This inserts the word "brown", and the <U-ARW> moves the new line to the "primary buffer".

Get back into the overstrike mode again. (Use <CTRL> I)

If your program was configured with word wrap "on", then you are set for what we are about to do. If not, you'll need to go back to the COMMAND mode (key <CTRL> <SPACE> to do so) and turn on the word wrap by keying...

WRAP <ENTER>

If the top line now reads "WRAP OFF", enter the word "WRAP" again. If not, key <BREAK> and return to the TEXT mode.

Clear out all text entered so far by keying in...

<ALT> <CTRL> C

Answer the "Are you sure?" prompt with "y" for "yes".

Try typing in some lines of text on your own. DO NOT press <ENTER> when you reach the end of a screen line unless you want to force the line to end there. When you reach the end of a screen line, the program will automatically take the word that went beyond the end and "wrap" it down to the next line. The cursor will follow and remain in the correct place. Later, when you want to print your text out on paper, the program will automatically end each line where it should end according to your left and right margin settings. After you reach the bottom of the screen, each new line you type in will cause the screen to scroll up (each line moves up) to accommodate the new line. The top line will scroll off the top and will no longer be visible. If you stop and use the up <U-ARW> to move up, when you reach the top of the screen with the cursor, pressing the <U-ARW> again will cause the screen to scroll down, revealing the lines that were scrolled off the top.

When most of the keys are held down, the "key repeat" feature will cause the key to be output continuously until you release the key. Keys such as <BREAK>, <ENTER>, <CTRL>, <ALT>, <F1>, and <F2> will not repeat if held down.

You have now seen that the screen scrolls up and down. It is as if the screen were a window through which you are looking at long sheet of paper with text on it. When the screen scrolls up, the window seems to go down the sheet of paper. Not only can this window scroll up and down on this long sheet of paper, it can also scroll to the left and right if the paper is wider than the screen. To see this happen, goto the COMMAND mode by pressing <CTRL> <SPACE> and key in the following...

WIDTH100 <ENTER>

This command will cause the cursor to go to the top of your text when you return to the TEXT mode. Return to the TEXT mode by keying <BREAK>, and then key in the following line starting on a new line at the bottom of any previously entered text...

This time we are going to key in so much text into our
line that it will not

Don't hit <ENTER> yet but watch closely as you finish the line by typing...

all fit on the screen.

By changing the screen WIDTH setting to 100, the lines can now be longer than our screen. This seems to make our paper wider. Now our window can scroll to the right. After you key in the above, press <ENTER>. When you do, window returns to the left side and the cursor moves to the beginning of the next line. Move the cursor up into the long line. Using the <R-ARW> and the <L-ARW> to move from one end of the line to the other, will cause the window to scroll back and forth.

The screen WIDTH setting has nothing to do with how wide the text lines will be printed on paper. Try setting the WIDTH to small values like 10, 20, etc.

INTRODUCING NEW KEYS

COMPUTERS, PRINTERS, AND TEXT CHARACTERS:

When a computer sends text to a printer, it can't actually send the letter "A". Instead, it sends a number for the letter "A". If you want your printer to print the letter "A", it must be sent a number 65. Printers don't really understand numbers. They only know if the voltage level on a line is ON or OFF.

So that we humans can better work with computers, we look at an ON as the either the number 1 or 0, and an OFF as either the number 0 or 1. Because we are using only 0's and 1's, we are forced to count using the "binary" number system. Instead of counting 1, 2, 3, 4..., etc., binary numbers count 1, 10, ,11, 100, 101, 110, 111. Thus a decimal 2 would be represented by 10 in binary and 7 would be 111 and 255 would be 11111111 (255 is the highest we can count using up to eight 0's or 1's). If the computer looks at eight different voltage lines, then we can represent what is seen for ON's and OFF's as a number from 1 to 255. The computer sends the number 65 for the letter "A" to the printer. This number is also 01000001 in binary. What the computer actually did was to show the printer voltages it intrepreted as "OFF-ON-OFF-OFF-OFF-OFF-OFF-ON. Appendix A is a table showing the numbers from 1 to 127 and the key or sequence of keys to type to send that number to the printer.

The keys on your keyboard are each assigned numbers. Just as an "A" is assigned the number 65, the letter "a" is assigned the number 97. Other keys assigned values are like the backspace (8), horizontal tab (9), and the carriage return or <ENTER> (13). Some numbers are not assigned to any key on your keyboard. There isn't a key for the number 27, yet many printers must be sent the number 27 to produce certain results. This is where the <ALT> key comes into play.

THE <ALT> KEY:

When you key the <ALT> key, the "N" at the top right of your screen becomes a blinking "A" to alert you that the next key pressed will output an alternate value assigned to that key. For example if you key...

<ALT> A

you will not output the 65 for the letter "A" as discussed earlier. Instead, the number 1 will be output. Look at the table in Appendix A. It contains a list of the numbers generated by each <ALT> key sequence.

THE <CTRL> KEY:

The <CTRL> key is used to execute TEXT mode commands. You first press and release the <CTRL> key, and then the key assigned to the command. Each <CTRL> key command will be covered later. Here, I only want you to learn how to execute a <CTRL> key command. When you press and release the <CTRL> key, the "N" in the upper right of the screen changes to a blinking "C" to alert you that the Control key is active. The next key pressed executes the command for that particular key. The letter for the key you press after the <CTRL> key will be printed in upper case in this manual, however either upper or lower case may be used in actual practice.

THE <F1> KEY:

The <F1> key is used to generate print markers. It is also used to turn on or off one of the 5 pre-defined print fonts, as well as to turn underlining on or off. Get into the TEXT mode. (Press <BREAK> to exit from the COMMAND mode to the TEXT mode.) Be sure you are in the overstrike mode (Check for the "O" in the upper right corner of the screen and use <CTRL> I to toggle between modes.) Now key in...

I can turn <F1> Uunderlining on <F1> U or off.

No, "underlining" is not misspelled. The combination <F1> U turns on underlining and then later turns it off. Key a <SPACE> after the word "turn". There is no <SPACE> between the word "on" and the "<F1>". Without turning the underline on or off, the line would have read...

I can turn underlining on or off.

THE <F2> KEY:

The <F2> key serves only one purpose. It's the TAB key. If a tab stop is set to the right of the cursor, then pressing the <F2> key will move the cursor to the tab stop. Setting tab stops is covered later.

THE <CLEAR> KEY:

This key serves only one purpose. Later, when you learn about creating a second word processor on the same screen, you will use the <CLEAR> key to move back and forth between the "windows" that each word processor occupies. If you key the <CLEAR> key, you will get an error message that reads "Error: window not available"

THE <BREAK> KEY:

As you previously learned, the <BREAK> key will allow you to exit the COMMAND mode. It also acts as an "oops" key for the current line. Pressing the <BREAK> key also halts the execution of Tasks (covered later).

SIMPLY BETTER has what is called a "type ahead buffer". This buffer contains the keys you hit until the program is ready to process them. This way, if your disk drive is being accessed, you can still type in characters, but they'll be stored in the "type ahead buffer" until the program finishes its disk operation. If you are inserting text near the beginning of a long file, each time you insert enough text to cause a word to "wrap" to the next line down, the program must open a hole in the "primary buffer" for your new text. This "opening" can take enough time to cause the text typed in to be displayed slower than you are typing. This is why key beep is important. With out key beep, you would have no indication that the program actually got the key you pressed while it was trying to catch up to your typing. This is especially true when your disk drive is running. Text entry will be slowed way down because the disk controller actually stops the computer briefly as it performs each read and write. If, while typing in your text, you actually fill the type ahead buffer, (very difficult to do) you will hear a different error beep tone to let you know that new keys are not being saved and for you to pause for a second while the program catches up.

The <BREAK> key nulls all characters in the "type ahead buffer" except for the

<BREAK> key itself. Consider this example. While your disk system is loading in a file, you decide to type ahead. The keys you want to type are simply the word "WIDTH" followed by an <ENTER>. This command line will display the current screen width value. However, while keying this in (remember it is not being displayed as you type ahead) you think you may have accidentally keyed the wrong key. You can erase all of the characters not yet processed by keying <BREAK>. Remember that the <BREAK> key will be left in the type ahead buffer, so the first key the program uses when it finishes loading the file will be this <BREAK> key, and this in turn will place the program in the TEXT mode. In a case like this, after pressing the <BREAK> key, counteract it by keying <CTRL> <SPACE> so that the program will again be back in the COMMAND mode. Then key in "WIDTH" as it should have been.

MOVING ABOUT IN YOUR TEXT

So far you have been using the four arrow keys to move the cursor around inside the text. In this section you will learn more efficient ways of moving from place to place.

Key in lines of text until you have more than a single screen of text. Once you have done this, use the <U-ARW> to move the cursor to your top line of text.

NOTE: The letter after the <CTRL> key will be shown in upper case, but it can be keyed in as upper or lower case.

THE <CTRL> L AND <CTRL> R:

Using the <R-ARW> or <L-ARW>, move until the cursor is somewhere near the center of a line of text.

You can move the cursor to the left one word at a time by keying...
<CTRL> L

or you can move the cursor to the right one word at a time by keying...
<CTRL> R

NOTE: After you key the <CTRL> key and follow it with a second key, this sequence of keys can be repeated by holding down the <SHIFT> key and then keying the <ALT> key. The <SHIFT>-<ALT> repeats the last <CTRL> sequence keyed. If you hold both the <SHIFT> key and the <ALT> key down, the key repeat will execute the <CTRL> sequence repeatedly.

If you move the cursor to the beginning of a line and then use either the <L-ARW> or the <CTRL> L, the cursor will go to the end of the line above. You can continue backing up as far as desired.

THE <CTRL> <R-ARW> AND THE <CTRL> <L-ARW>:

Use the <CTRL> <R-ARW> to move to the end of a line and use the <CTRL> <L-ARW> to move to the beginning of a line. Realize that you move to the end of the screen line that the cursor is in. If the end of the screen line is off the right side of the screen, the window will scroll far enough to display all of the text on that screen line.

THE <CTRL> <U-ARW> AND THE <CTRL> <D-ARW>:

Use the <CTRL> <U-ARW> to scroll one screen up, and use the <CTRL> <D-ARW> to scroll one screen down. The screen actually moves one line less than a full screen so that as you move up, the line that was at the top will become the line at the bottom. As you move down, the line at the bottom will become the line at the top.

If you hold down the <SHIFT>-<ALT> to continuously repeat either the <CTRL> <U-ARW> or the <CTRL> <D-ARW>, the screen will not be redrawn for each repeated command. Listen carefully to the key beeps, as each beep is a screen page up or down. This provides super fast screen paging.

THE <CTRL> <SHIFT>-<R-ARW> AND THE <CTRL> <SHIFT>-<L-ARW>:

Use the <CTRL> <SHIFT>-<R-ARW> to move to the top of the screen. Use the <CTRL> <SHIFT>-<L-ARW> to move to the bottom of the screen.

THE <CTRL> <SHIFT>-<U-ARW> AND THE <CTRL> <SHIFT>-<D-ARW>:

Use the <CTRL> <SHIFT>-<U-ARW> to go to the very beginning of your text. Use the <CTRL> <SHIFT>-<D-ARW> to go to the very end of your text.

THE TAB KEY AND TAB STOPS

SIMPLY BETTER was designed so that it was "simple" to use, and yet had "powerful" features. One thing that has always bothered me with other word processors was setting tab stops. With SIMPLY BETTER, just put the cursor where you would like a tab stop. Watch the column number on the top line if you know the column number that you want the tab stop at. Then key...
<CTRL> T

If no tab stop exists where you do this, one will be set. If a tab stop does exist, then it will be erased.

The Tab key is the <F2> key. You can have a tab stop at every single position if desired. If your screen width is set to 250, then you can have 250 tab stops!

To erase a tab stop, simply tab over to the one that you want erased and key...
<CTRL> T

To erase all tab stops, go to the COMMAND mode and key in...
TABOUT <ENTER>

Answer the "Are you sure?" with "y" for yes or "n" for no.

INSERTING TEXT

One way if inserting text inside the text that already exists is by typing in the text while in the insert mode. (Use <CTRL> I to toggle between the insert and overstrike mode. Note which you are in by the presence of an "O" or "I" in the upper right corner of the screen.) Simply move the cursor where you want to insert text, and type it in. The text characters typed will be inserted to the left side of the cursor. The cursor will shift to the right so that the next character will be inserted where it belongs. This may be ok for a few characters but because the program has to continually open new holes in the "primary buffer" for your new text, it tends to slow the display down quite a bit.

USING <CTRL> O:

The <CTRL> O is the "open for text" command. If you want to insert text in the middle of a line, place the cursor where you want to insert the text and then key...

<CTRL> O

All text at and beyond the cursor will drop down to the next line, leaving the line open and empty after the cursor. Now type in your new text. If you are in the overstrike mode, and you type past the end of a line, the text will once again open up for another line of text. If you type past the end while in the insert mode, the text will not open and it will leave you inserting text as you were before using the <CTRL> O. Thus, if you are going to insert large amounts of text, it is wiser to be in the overstrike mode and open the text for the new text using the <CTRL> O command.

Using the <CTRL> O is not the only time that your text will open up for a line of text. This also happens any time that you are in the overstrike mode and you type past the end of a screen line. When you do, the text will open up for the next line of text typed.

If you want to "close up" the lines of text after it has opened, simply key either the <U-ARW> or the <D-ARW>. If, while the text is open, you move the cursor by using a <CTRL> command, the text will close automatically.

DELETING TEXT

THE <CTRL> D:

This is the "delete character" command. If you want to delete one or more characters, place the cursor over the character you want deleted and key...

<CTRL> D

This will delete the character from the line and shift the rest of the line to to the left, closing the hole left by the deletion. This only changes the current line buffer and the

only deletes characters from the current line buffer, the <CTRL> D command can not delete more than the current line that the cursor is in. When you leave the line or use the <U-ARW> or <D-ARW>, the current line will be moved to the "primary buffer".

THE <ALT> <CTRL> L AND THE <ALT> <CTRL> R:

Use the <ALT> <CTRL> L to delete the word to the left of the cursor, and use the <ALT> <CTRL> R to delete the word to the right of the cursor. If the cursor is inside a word, then only the portion of the word to the right or left of the cursor will be deleted the first time the sequence is keyed. The deleted words can be restored using the <BREAK> key as with <CTRL> D. It is important to remember that a text word includes the space at the end of the word. Periods, commas, etc. are considered part of the word to their left. Multiple spaces together are each considered a word.

Don't forget that a lot of the power of these commands is in your ability to execute them consecutively by using the <SHIFT>-<ALT>. (The <SHIFT>-<ALT> will repeat the last control key hit.)

THE <CTRL> <SHIFT>-< AND <CTRL> <SHIFT>->:

The <CTRL> <SHIFT>-< will delete all text from the cursor to the beginning of the line. The <CTRL> <SHIFT>-> will delete all text from and including the character at the cursor to the end of the line. Just remember which way each symbol is pointing.

THE <CTRL> X:

This command (Xouts) erases the entire line that the cursor is in. The line can also be restored by using <BREAK> as an "oops" key. If you want to delete more than one line, you can NOT delete lines consecutively by using the <SHIFT>-<ALT> and holding down the <ALT> key. This is because you must key either the <D-ARW> or the <U-ARW> before the text will close up the hole made by the deletion. If you need to delete numerous lines, it is easier to mark off the lines to be deleted as a block, and delete all of the lines at one time. Blocks will be covered later.

You can erase all text before the cursor by keying...

<ALT> <CTRL> A

You can delete all text after the cursor by keying...

<ALT> <CTRL> Z

You can delete all text from the buffer by keying...

<ALT> <CTRL> C

Answer the "Are you sure?" prompt with "y" for yes.

After executing both the <ALT> <CTRL> C or the <ALT> <CTRL> Z, you can recover the deleted text by going to the command mode and keying the command...

OOPS <ENTER>

You can also delete all text while in COMMAND mode by keying in the command...

NEW <ENTER>

Answer the "Are you sure?" prompt with "y" for yes.

UNDERLINING AND PRINT FONTS

Now you are at one of the more exciting parts!

DO NOT RUSH THROUGH THIS SECTION!

There are a few important rules that you should be aware of if you want to eliminate troubles using underlining and print fonts. Once you grasp how underlining and print fonts are added (or removed), you will appreciate the simplicity.

> ***RULE:*** Your current print font is always the print font of the text character prior to the cursor. This print font will be displayed at the top of the screen to the right of the column number., Underlining is not displayed at the top of the screen; only the print font.

First, make sure you are in the overstrike mode. (Check for the presence of an "O" in the upper right corner of the screen.)

Underling is toggled "on" or "off" while typing in text by keying...
<F1> U

Underlining and print fonts can be easily added or changed after your text has been keyed in. This will be covered later.

Try keying in the following line...

> It's great to <F1> Uunderline <F1> U text
> on the screen. <ENTER>

You should end up with this on your screen...

> It's great to underline text on the screen.

If not, key it in again carefully. Again, you must be in the overstrike mode. Note that in the line, "<F1> Uunderline <F1> U " is keyed by pressing the <F1> key, then the "u" key, then the characters "underline", then the <F1> key, then the "u" key, and then a single <SPACE>.

Your screen can display up to 5 print fonts in colors selected while using the "CONFIG" program. A print font is turned on by keying the <F1> and then a number from 1 to 5. The sequence <F1> 0 will turn on normal print font. Changing from one print font to the next, automatically turns off the previous font and turns on the new font, including going to normal print font. If you are in print font #4, you can go immediately to another print font by keying <F1> and then the number for the new font. Think of normal font as an actual print font.

Key in the following line which will change to print font #1 and then back to normal print font.

> This is normal, <F1> 1this is font
> #1, <F1> 0 this is normal. <ENTER>

Try keying in a few characters, then change to a new font, key in a few more characters, change fonts again.

Key in this line carefully...

> This is <F1> 3GREAT <F1> 0 fun. <ENTER>

You should have...

> This is GREAT fun._

on your screen with only the word "GREAT" being of a different color. Remember that our notation includes a <SPACE> after the word "is" (before the first <F1>) and there is no <SPACE> keyed between the word "GREAT" and the second <F1> key.

Move the cursor so that it is on the "G" of the word "GREAT". Now look at the top line of the screen and notice that the font is "Normal-0". This is because the current font is the font of the character to the left of the cursor. In this case, the space to the left of the cursor is of normal font.

Now move the cursor to the right one character. The top line now shows that the current font is print font #3. Move the cursor to the space after the "T". Even though the space at the cursor is normal print font, the line at the top of the screen still shows print font #3. Again, it is the print font of the character prior to the cursor. You will see how this has a great advantage.

If you are typing in text over the top of some text and you come to some text of a different font, your text will continue to be the font of the characters you have been typing. If you begin typing after the last character of a specific font, the new text will be of that font.

Now get into the insert mode using <CTRL> I if there is an "O" in the upper right corner of the screen. Move the cursor over the very first character of a new print font. If you key in characters, they will be of the print font of the previous characters. If you place the cursor inside or just after the print font, new characters are inserted in the same print font as that print font.

Clear out all text by keying...
<ALT> <CTRL> C

Go to the COMMAND mode by keying <CTRL> <SPACE> and set your screen width to 30 by entering...
WIDTH30 <ENTER>

Now, return to the TEXT mode by keying <BREAK>, and key in...

> This paragraph will be used to
> show how you can edit your
> text after you have entered it
> into the word processor. <ENTER>

Be SURE you are now in the insert mode!

Move the cursor so that it is on the first "p" in the word "paragraph". Hang on to your hat as you key...

<F1> U

Wow! You should now have underlining all of the way to the end of the paragraph! Move the cursor to the "a" in the word "after" and again key...

<F1> U

Your text should now look as shown here...

> This paragraph will be used to
> show how you can edit your
> text after you have entered it
> into the word processor._

Now move the cursor to the "w" in "will" and key...

<F1> 4

You should now have text of print font #4 going all the way to the end of the paragraph from the cursor. Move the cursor to the "e" in "edit" and now key...

<F1> 0

Now, only the words "will be used to show how you can " should be of print font #4.

> ***RULE:*** While in the insert mode, keying the <F1> followed by a font number from 0 to 5 will turn on that font for characters starting with the character under the cursor. This font will continue until it comes to a font different than that of the one being changed.

If you turn on a different font in the paragraph above, starting at the beginning of the paragraph, the font will stop when it reaches the "w" in "will" where font #4 starts.

This rule also enables you to return sections of print back to normal font. Simply place the cursor where you want normal font and key...

<F1> 0

Font #0, which is normal text, will continue until it reaches a font other than the one that is being changed to font #0.

This same principal works with underlining. If you are in the insert mode and you turn on underlining, it will continue until it comes to text that is already underlined.

AN IMPORTANT EXCEPTION: If you turn off underlining or change a print font for a section of text, (while in the insert mode) the hidden code used by the program to mark the beginning will be erased, however, if the place where the font or underlining ended is not in the same screen line as where it was turned off, the hidden code that was used to mark the end of the font or underlining will remain in the "primary buffer". This has both an advantage and a disadvantage. The advantage is that you can easily move where a section of print font begins without having to go down numerous lines to reset where it ends. The disadvantage is that if you leave any of these hidden codes in the the text and then use the text as source code for an

assembler or other program that does not recognize the code, this can sometimes cause problems. If you have a situation where you don't want the codes that SIMPLY BETTER uses to display print fonts and underlining, go to the COMMAND mode and use...

CLEAN <ENTER>

This will remove ALL underlining and ALL print fonts from ALL of your text!

> ***RULE:*** If, while in the overstrike mode, you go past a carriage return or wrap to the next line down, your current print font will continue past the cursor until it reaches a print font different than the print font that was at the end of the line before you went past it.

You will probably be startled when the happens to you. To demonstrate how this happens, try this example. First, go to the COMMAND mode by keying <CTRL> <SPACE>. Then key in...

80 <ENTER>

to create an eighty column screen. Then key in...

WIDTH79 <ENTER>

to set the screen width to 79. Return to the text mode (press the <BREAK> key) and then, while in the overstrike mode (check for the "O" in the upper right corner of the screen and toggle in "on" with <CTRL> I if needed) and key in these two lines...

> This will <F1> 3show <F1> 0 what happens. <ENTER>
> This is the line that will change. <ENTER>

If you keyed this in properly, the word "show" will be in print font #3. Move the cursor so that it is over the "w" in show. Now key the following...

> w what happens to your text

Stop here and do not press <ENTER>. Your text should now read...

> This will show what happens to your text
> This is the line that will change._

The text you just keyed over ran the end of the line. The print font at the end of the line was normal print font. Your current print font is #3. Now allow the text to close by keying...

<D-ARW>

As the rule stated, the new rint font continues until it reaches a font different from that of the print font that was at the end of the line.

When this does happen, get into the insert mode and use the proper <F1> sequence to reset the print font back to what it was before. Do this by first keying <F1> followed by the number of the print font that was changed.

CHANGING DEFAULT PRINT FONTS:

You can change any of the 5 default print fonts. These are the print fonts created by using the <F1> key plus a number from 1 to 5. To change one of these fonts, first key...

<ALT> <CTRL>

This is the "redo Font" command. When you key this, the screen will ask...

Redo (1-5):

Key the number of the print font that you want to change. When you do, the screen will then read...

Desc:

Key in up to 15 characters that describe your new font. It is recommended that you make the last two characters a dash "-", and the number of the font. For example, if you were changing print font #2 to "italics", you might use...

Italics-2

After you enter the description, the screen will then ask for the "On" and "Off" codes for the new print font. Key in the numbers and if there is more than one number, separate them with commas. Your line might look like...

On: 27,14

After having done this, the new font will be active until the program ends. Any font of the same font number that was created previously will also change to this new font!

WORKING WITH BLOCKS OF TEXT

MARKING BLOCKS OF TEXT:

You mark blocks of text using the same method as marking print fonts. While in the overstrike mode, key...

<F1> B

This marks the beginning of a block. Continue keying in text. All of your text will now be the color chosen for blocks. You can toggle it off by keying...

<F1> B

Although blocks are represented in their own color, they are NOT print fonts. While you are keying in text that is part of a block, you can still turn on or off print fonts and underlining. The block will be displayed in the block color and the color of different print fonts will be hidden.

Although you can enter new text as part of a block, (as you did above) you will likely be editing sections off text that already exist. You mark a block of text using the same method as when you marked a section of text to be a specific print font. To show this, get in the insert mode and place the cursor at the beginning of a section of text that you want to mark. Now key...

<F1> B

The block just created will continue until it reaches the next block, or until it reaches the end of the text. Move your cursor to the first character past where you want the block to end and again key...

<F1> B

Now you have marked a block of text that can be erased, copied, moved, printed, and even saved to disk.

It is important to remember that if you mark a block that continues down one or more screen lines to another marked block of text, the hidden "on" code will still exist where the second block began. Try marking a small block of text down a few lines in your text. Then go up a few lines above the block and, while in the insert mode, turn on a block here. The new block will continue until it reaches the block below it. Now turn the block off again. Notice that the second block remains because of the hidden on code left there. REMOVING BLOCK MARKERS:

If you want to remove ALL of the "block" markers at or below your cursor, you can use...

<CTRL> Z

This is the "zap block markers" command. It only "zaps" those markers at or beyond the cursor. If the cursor is inside a marked block, then the beginning of the block will be before the cursor, and it will not be erased. Since all markers after the cursor will be erased, the block will now continue from its beginning above the cursor to the end of text!

While in the overstrike mode, key in the following at the beginning of a line...

W <F1> Bo <F1> Br <F1> Bd <F1> B <ENTER>

As seen here, you can actually mark every other character as a block. If your computer has 512k of memory, you can mark over 79,000 separate blocks!

ERASING BLOCKS OF TEXT:

You can erase a marked block of text regardless of where the cursor is in your text. To erase a block of text, (be sure one is marked somewhere) simply key...

<CTRL> E

If there is one or more blocks of text marked, a "bar" will be drawn across the screen one line from the bottom and the first line of the first block marked in your text will be displayed below the bar. The top of the screen will ask "Correct block?". If this is not the correct block, key "n" and the program will move on to the next block. If the next block is the correct one, answer the prompt with "y" for "yes". If the verify is on, (look for a "V" in the upper right corner of the screen) the program will ask "Are you sure?". If you are, answer with "y" for "yes". The program will then erase the block from your text.

When you erase a block of text with the cursor above the block of text, your position in the text will stay the same. If your cursor is in the block of text to be erased, the first line after the erased block will now be at the top of your screen and the cursor will be at that line. If your cursor is below the block of text erased, the line with the cursor, and the cursor, will now be at the top of the screen.

COPYING BLOCKS OF TEXT:

You can copy any block of text to anywhere OUTSIDE that block of text. Place the cursor where you want to copy the block to. Then key...

<CTRL> K

This is the "kopy" block command. Select the block in the same way as when you erased a block. The block will be copied to the cursor and the original block will remain as it was. One thing you are going to appreciate is that all underlining and print fonts are copied right along with the block. You can copy a block of text that is an underlined print font into the middle of a section of text that has a different font, and the underlined print font will be maintained properly. Try copying just the middle of a section of text that is one of the print fonts to some place else. Even though your block did not get the beginning or end of the text in that font, it still copies properly.

MOVING BLOCKS OF TEXT:

You can move blocks of text in the same way that you copied them above. The only difference is that when you move a block of text, the original block will be erased. It is as if you picked up the block of text, closed the hole where it was originally, and then inserted it where the cursor was at. To copy a block of text, first mark the block to move, and then key in...

<CTRL> M

Select the block to move just as you did in "block copy" above.

PRINTING BLOCKS OF TEXT:

If you have configured your program properly for your printer, (such as baud rate, line feeds, nulls) you can now print a block of text. If you have not yet configured your program for your printer, don't worry, there's a whole section on printing later. (See section "PRINTING TEXT ON PAPER".)

To print a block of text, first mark the section to be printed as a block. Then key...

<CTRL> B

This is the "block print" command. Select the block as you did previously. After selecting the proper block, your printer will print out only the text in the block. If the program does not begin printing and seems to have "hung up", check to see that your printer is both "on", and "on line". If this fails to correct the problem, you can abort the attempt to print by keying...

<SHIFT>-<BREAK>

PRINTING A BLOCK TO MEMORY:

If you want to save a block of text to disk with it formatted identical to how it would be printed, first print the text into memory. When you do this, a copy of the text in the block will be appended to the end of your existing text, just as if it had been printed. Margins, headers, etc., will be added to the text. Since the text may later be printed using a different type printer, printer codes for underlining and fonts will all be filtered out. To print a block to memory, key in the following command...

<ALT> <CTRL> B

Select the correct block as you would with the <CTRL> B command above. After you initiate the printing, an alert will sound when printing is completed. You will find the printed block of text at the end of your current text. You can now alter it if needed, and then save it to disk using either the "CS" (cursor save) command or by marking it as a block and using the "BS" (block save) command.

SAVING A BLOCK OF TEXT TO DISK:

To save a block of text, first go to COMMAND mode, and then key in a command using the following format...

BSfilename <ENTER>

where "filename" is the name, extension, and drive number to be used in the block save. If you omit any of these three items, the program will use the current default value. The filename used becomes the default filename!

If you omit the name, precede the the extension with a "." (period). If you omit the extension, precede the drive number with a ":" (colon). If you use only "BS", the program will use the entire default filename. Look at the samples at the beginning of the section "LOADING AND SAVING TEXT FILES".

PRINT PARAMETERS

Print parameters are those values listed on the third screen in the CONFIG program. You can view the print parameters at any time by going to the COMMAND mode and keying...

/ <ENTER>

The screen will then clear and display the print parameters and their current values. Each parameter is represented by a two letter abbreviation, followed by its current value or setting. Each parameter and its range in "( )" is shown below...

AL: Auxiliary line A line number (0 to "PL")
AS: Auxiliary line A status (AEO)
BD: Baud rate (0 to 6)
BL: Auxiliary line B line number (0 to "PL")
BM: Bottom margin ("TM"+1 to "PL")
BS: Auxiliary line B status (AEO)
FL: Footer line number (0 to "PL")
FS: Footer status (AEO)
HL: Header line number (0 to "PL")
HS: Header status (AEO)
IC: Index column number (0 to 230)
IP: Invisible print (YN)
JN: Justify space number (0 to 10)
JS: Justify spaces (YN)
JU: Justification (LRCB)
LF: Line feeds (YN)
LM: Left margin (0 to "RM"-1)
LS: Line spacing (1 to 10)
NP: No-prints printed (YN)
NU: Nulls (0 to 230)
PA: Pagination (YN)
PC: Print comments (YN)
PF: Print fill form (YN)
PH: Printer handshake (YN)
PL: Page length ("BM" to 255)
PN: Page number (0 to 65,535)
PP: Print pages (AEO)
RM: Right margin ("LM"+1 to 250)
TC: Table of contents column number (0 to 230)
TM: Top margin (0 to "BM"-1)

An item in between quotes in the range is one of the parameters listed to the left.

(AEO) is All, Even or Odd (YN) is Yes or No
(LRCB) is Left, Right, Center, or Both

CHANGING PARAMETERS ON THE COMMAND LINE:

To change the value of a parameter on the COMMAND line, first key a slash "/". Follow this with the parameters and values that you want to change. Each parameter must be separated from the next by a comma. A sample line is shown below...

/tm0,bm54,lm10

Here, the top margin is changed to 0, the bottom margin is changed to 54, and the left margin is changed to 10.

CHANGING PARAMETERS INSIDE YOUR TEXT:

To change the value of a parameter in your text, first key...

<F1> /

This will create a "begin parameters" marker. Then continue with the parameters that you wish to change just as you did on the COMMAND line above. The parameters MUST be terminated by a carriage return character.

If you are changing the value or status for either the header, the footer, or an auxiliary line, then it is best to place the "begin parameters" marker at the beginning of a printed line. Consider these two lines...

My line <F1> /HL3 <ENTER>
has a parameter line inside it. <ENTER>

The carriage return (created by the <ENTER> key) at the end of the top line terminates the parameters and is NOT part of the text. The line would be printed as...

My line has a parameter line inside it.

However, if this were the top line for a printed page, the top margin would be printed BEFORE the parameters were processed and the program would not generate a header for this page.

This program was designed to allow you to place parameter changes inside lines. This allows you to change parameters inside a paragraph without knowing where each printed line ends. Consider this example where the default left margin is set to 8...

The first line <F1> /lm13 <ENTER>
of the paragraph has the default left margin
while the rest have a left margin of 13. <ENTER>

This could be done to obtain a printed paragraph as shown below...

The first line of the paragraph has the
default left margin while the rest
has a left margin of 13.

SETTING DEFAULT PARAMETERS:

You can reset all parameters to the values that they were when you first booted the program by using the asterisk "*" as a parameter in your line as the first or only

character. In the line below...

/*

all values are set to their initial value for printing. In the line below...

/*,jub

all values are set to their initial value, but then the justification is changed to "b" for both left and right justification.

HEADERS, FOOTERS, AUXILIARY LINES:

Each of these lines are identical as far as SIMPLY BETTER is concerned. That is, you can set up the Footer as the Header, or the Header as an Auxiliary line. The name given to each line is for your benefit. If you use the "Header" for headers and the "Footer" for footers, you will find it easier to keep track of which line is what.

Each of these lines has a line number that can be assigned to it using the parameters...

HL = header line number FL = the footer line number
AL = aux line A line number BL for aux line B line number

A line assigned the number 0 will not be printed because a page never has a line number of 0. After defining what the text for the line will be, you can turn off the output of the line for certain pages by setting its line number to 0.

Each of these lines also has a status (the parameters HS, FS, AS, and BS) that is assigned to it. The status can be All, Even, or Odd. If you only want the line printed on even pages, then set the status to "e" for "even". If you want it printed on all pages, then set the status to "a" for "all" pages. If you only want it printed on odd pages, then set the status to "o" for "odd". If two of these lines are assigned the same line number, then the line that will actually be printed will be by its priority in the following order...

1. Header 2. Footer 3. Auxiliary A 4. Auxiliary B

To define one of these lines, generate a marker by keying the <F1> key followed by an "h" for header, an "f" for footer, a comma "," for auxiliary A, or a period "." for auxiliary B. The marker must be the first character of the line! After the marker, key in the text that you want to have printed for that line followed by <ENTER> to terminate it.

IMPORTANT: Headers, footers, and auxiliary lines are all "left" justified and are not assigned left or right margins. This is because these lines must be printed the same, regardless of how your text is formatted at the time the line is printed. Therefore you must include spaces at the beginning of the line if you wish to have it printed other than at the extreme left side of the page. Because each has no right margin, be careful not to use more characters than will fit on the printed line.

You can use underlining and print fonts in any of these lines. Each line will always be printed in its own font, regardless of what is happening

turns off all print fonts and underlining before printing the header. It then sets up the print fonts for the header, footer, or auxiliary line. After printing the line, the program then turns off any fonts or underlining left on, and once again turns on any print font or underlining needed for the regular text.

You can use one of these lines to include your page number. One of the features of SIMPLY BETTER is that you can have the current page number inserted anywhere inside your text by using an "insert page number" marker. To do this, simply key...

<F1> <SHIFT>-#

This will put a reversed "#" on your screen. Later, during the actual printing (or viewing in the VIEW mode), the page number for the page containing the "#" will be inserted for the "#" character.

Look at this example...

<F1> , Page number <F1> <SHIFT>-# of 10 <ENTER>

When printed, Auxiliary line A would look like this...

Page number 4 of 10

Here's another example...

<F1> . - - - - - - - - - - - - - - - <ENTER>

This line would place a line of dashes across the page that could serve as a fold or tear line.

You may NOT use "header", "footer" or "auxiliary" markers inside these lines. Do not use "no-print" markers, "comment" markers, "index item" markers, or "table of contents item" markers inside these lines.

If you want to begin printing your text BELOW the lines that defines and turns on the header, footer, or auxiliary line, then you must first "window" through the lines that define it and turn it on while in the VIEW mode. (See section "THE VIEW MODE".)

MARGINS:

The margins include the top margin (TM), bottom margin (BM), left margin (LM), and the right margin (RM). The acceptable range for these are shown in the parameter list above.

INDEX COLUMN NUMBER:

See section entitled "CREATING AN INDEX".

TABLE OF CONTENTS COLUMN NUMBER:

See section entitled "CREATING A TABLE OF CONTENTS".

INVISIBLE PRINTING:
You can begin invisible printing by changing the "invisible print" parameter (IP) from "n" to "y" for "yes". If you turn on invisible printing for a section of your text, the program will still move through the text exactly as if it were printing the text except that no print will be sent to the printer. If you do not want a section of text printed, and yet you want your page numbers, margins, etc., to be updated as if they were printed, then use invisible printing for that section.

If you "invisible print" a large section of text, have some patience because it will still take the program time to print it invisibly.

TEXT JUSTIFICATION:
You can change the justification (JU) for your text to Left, Right, Center, or Both.

If the justification is set to Left (JUL), then all lines will begin evenly at the left margin and will end where the last word printed will not go beyond the right margin setting. This leaves a ragged right side.

If the justification is set to Right (JUR), all lines will end evenly at the right margin, and will begin where the maximum number of words that will fit in each line backs up to. This leaves a ragged left side.

You can set the justification to Center (JUC). Each line will then be centered between the left and right margins.

When the justification is set to Both (JUB), the program will insert enough spaces between words in each line that is not terminated by a carriage return, until the line both begins and ends evenly at the left and right margins.

SINGLE LINE JUSTIFICATION:
The text justification can be changed for individual lines. The new justification can be forced to Left, Right, or Center for this line only. Somewhere in the line to be changed, (preferably at the beginning) insert markers as follows...

<F1> <SHIFT>->
will generate a "force right" marker (a reverse colored ">") and forces the line to be printed with Right justification. This is handy for such things as "forcing" the date in a letter to end at the right margin.

<F1> <SHIFT>-<
will generate a "force left" marker (a reverse colored "<") and forces the line to be printed with Left justification.

<F1> C
will generate a "force center" marker (a reverse colored "c") and forces the line to be printed with Center justification.

JUSTIFICATION SPACES:
This feature allows you to set the maximum number of spaces that can be inserted between two words when text is printed with the justification set to Both (JUB).

This can alert you to lines that should perhaps be hyphenated to prevent unsightly over spacing. To set this up, set the "JN" parameter to the value for the maximum number of spaces to be inserted. Then turn on the feature by setting the "JS" parameter to "y" for "yes". Once you have determined the maximum number of spaces you will most often allow, you may wish to use the CONFIG program to change the default number to the one used most often. Then you won't have to remember a number and you can simply use the "JS" parameter to turn on this feature when you want it.

LINE FEEDS:
A line feed is simply a number sent to a printer to advance the paper a single line. If your printer requires this along with the carriage return, turn the "LF" parameter to "y" for "yes". If you are not sure about your printer, leave it "off" and try printing a couple of lines. If each line is printed on top of the previous, then you do need line feeds on.

LINE SPACING:
Set the "line spacing" parameter (LS) to the desired value. A value of 1 outputs your text with single line spacing. A value of 2 is double line spacing, etc.. If a header, footer, or auxiliary line is printed within the line pacing, the line spacing will resume again after the header, footer, or auxiliary line by printing the next text line and then the additional line spaces.

NO-PRINT AREAS:
You can skip printing sections of text by inserting "no-print" markers (a reverse colored "n"). Generate the marker by keying...
<F1> N

All text between "no-print" marker will not be printed and the text will be completely ignored. It is as if everything between the markers does not exist. You can print the text between these markers by changing the "no-print prints" parameter (NP) from "n" to "y" for "yes".

NULLS:
Nulls are short periods of time between when a carriage return is sent to the printer, and when the next character is sent. If your printer requires additional time for carriage returns, adjust this time by increasing the "NU" (null) parameter.

PAGINATION:
If you want printing to stop after each page is printed, change the "PA" parameter from "n" to "y" for "yes". You will need to do this if you are feeding your printer single sheets of paper and need time to insert each new sheet. When you want printing to resume, simply press a key.

PRINT COMMENTS:
You can add comments to your text that will not be printed when you print out your text. Begin your comment with a "comment marker" (reverse colored "*") by keying...
<F1> <SHIFT>-*

Now, key in your comments. Terminate the comment with <ENTER>. The carriage return character created by the <ENTER> will not be part of your text. For example...

```
This is what  <F1> * The comment line <ENTER>
is actually printed. <ENTER>
```

results in the following sent to the printer...

```
This is what is actually printed.
```

The comment "The comment line" is not printed. You can make an entire line a comment, such as...

```
<F1> * This line is a comment. <ENTER>
```

Now the entire line, including the <ENTER> is a comment.

If you have one or more comments in your text and you want to print the text, including the comments, you can do so by setting the "print comments" parameter (PC) to "y" for "yes".

PRINT FILL FORM:

This feature of SIMPLY BETTER enables you place a pre-printed form in your printer and easily print text on the form where you want it. If you've ever tried to print text exactly where you want it on a form using a word processor, you probably realize it's nearly impossible. Now you can do it easily!

If you have a form that you fill out repeatedly, then read section "PRINT FILLING A FORM" that uses the "PF" parameter.

PRINTER HANDSHAKE:

Most printers have an additional "busy" line that it uses to tell the computer when it is ready for more text. If this is the case with your printer, then set the "printer handshake" parameter (PH) to "y" for "yes". If the parameter is set to "n" for "no", then text is sent to the printer without waiting for any signal from the printer.

PAGE LENGTH:

Set the length of your printed page using the "page length" parameter (PL). This way, the program will automatically advance you paper to the next page when it is finished printing the page it's on. If you want to be able to print on every line of your page, then set the top margin (TM) to 0 and set the bottom margin (BM) to the page length (PL).

PAGE NUMBER:

You can set the page number for the current page using the "page number" parameter (PN). If you want to begin printing text so that the first page will begin with a certain number, set the "PN" parameter to that value and begin printing. Remember to reset it before you print again.

PRINT PAGES:

The "print pages" parameter (PP) controls which pages of your text are to be printed. If you want all pages printed, set the "print pages" parameter to "a" for "all". If you want only the even numbered pages printed, set it to "e" for "even". If you want only the odd numbered pages printed, set it to "o" for "odd". This is handy if you want to print your text on both sides of your paper while using continuous feed paper. Simply set the parameter to even or odd and print out your text. When the printing ends, remove the paper, insert it in the printer so that the backside will be printed correctly, and then set the parameter value opposite to what it was (even or odd) when in was just printed, and then print the text again.

If you want the page numbers to alternate from left to right, simply change the the line (header, footer, or auxiliary line) that generates the page number before printing the back sides.

BAUD RATE:

Use the BD parameter to set the baud rate according to the following table...

0	Parallel output		
1	300 baud	4	2400 baud
2	600 baud	5	4800 baud
3	1200 baud	6	9600 baud

IMBEDDED MARKERS

In this section we will list and explain each of the markers that are available to you.

FORCE JUSTIFICATION CENTER MARKER: <F1> C

This marker forces the justification of the line containing the marker to be printed with "center justification".

END SORT MARKER: <F1> E

This marker, (explained later in section 1.23 "SORTING SECTIONS OF TEXT") marks the end of the last section of text that is to be sorted.

FOOTER LINE MARKER: <F1> F

This marker is used to define your footer line. It MUST be the first character of a printed line. After the marker, key in the text that will make up the footer line and terminate the line with a carriage return character.

HEADER LINE MARKER: <F1> H

This marker is used to define the header. It MUST be the first character of a printed line. After the markey, key in the text that will make up the header line and terminate the line with a carriage return character.

INDEX ITEM MARKER: <F1> I
This marker is used to mark the beginning of an item that is to be placed in the index if indexing is turned on. (See section "CREATING AN INDEX" for more information.)

MAIL-MERGE LIST MARKER: <F1> L
This marker is used to mark the beginning of a list to be used when doing a "mail-merge". It is followed immediately with the number of items (lines) that each group in the list has. The number is terminated by a carriage return character and is then followed by the first line of the first group of lines. (See section "PERFORMING MAIL-MERGES" for more information.)

MAIL-MERGE ITEM INSERT MARKER: <F1> M
This marker is placed before and after the number of the item in a mail-merge list that is to be inserted at that point in your text. (See section "PERFORMING MAIL-MERGES" for more information.)

NO-PRINT MARKER: <F1> N
All text between two "no-print" markers is ignored when text is sent to a printer (or viewed in the VIEW mode).

PRINT PAUSE MARKER: <F1> P
When the print routine comes to a "print pause marker" in your text, it halts printing until you press a key on the keyboard. You may wish to halt printing for a moment while you check the output at the printer, or for some other reason.

SORT TEXT MARKER: <F1> S
Use this marker at the beginning of each section of text that is to be sorted. (See section "SORTING SECTIONS OF TEXT" for more information.)

TABLE OF CONTENTS ITEM MARKER: <F1> T
Place this marker in your text and then follow it with the text that you want to have placed in the Table of Contents. Terminate the text with a carriage return character. Both the text and the carriage return are NOT part of the actual text. (See section "CREATING A TABLE OF CONTENTS" for more information.)

AUXILIARY LINE A MARKER: <F1> ,
Use this marker to define Auxiliary line A. It MUST be the first character of the printed line! Follow it with the characters that will make up the auxiliary line and terminate it with a carriage return character.

AUXILIARY LINE B MARKER: <F1> .
Use this marker to define Auxiliary line B. It MUST be the first character of the printed line! Follow it with the characters that will make up the auxiliary line and terminate it with a carriage return character.

PARAMETER LINE MARKER: <F1> /
This marks the beginning of a group of one or more parameters that are to be changed. The parameters are separated by commas and terminated by a carriage return. The parameters and the carriage return are not part of the actual text.

FORCE JUSTIFICATION RIGHT MARKER: <F1> <SHIFT>->
This marker overrides the "JU" parameter and forces the line with the marker to be printed with "right justification".

FORCE JUSTIFICATION LEFT MARKER: <F1> <SHIFT>-<
This marker overrides the "JU" parameter and forces the line with the marker to be printed with "left justification.

COMMENT MARKER: <F1> <SHIFT>-*
This marks the beginning of a comment. The comment continues until it is terminated by a carriage return character. The comment is not part of the text and will not be printed unless the "PC" parameter is set to "y" for "yes".

HARD SPACE MARKER: <F1> <SPACE>
This is not a "marker" in the way the others are. Instead, this character is substituted for a "space" between two words that are not be separated at the end of a line because of word wrap. For example, if you had "Mr. Smith" in your text, you would not want a line to end in "Mr." with the name "Smith" at the beginning of the next line down. If you make the space between "Mr." and "Smith" one of these "hard spaces", the words will not be broken apart.

INSERT PAGE NUMBER MARKER: <F1> <SHIFT>-#
You can use this marker anywhere in your text. Place it in a header, footer, or auxiliary line when you want to print page numbers. Any time this marker is encountered in your text during printing, the current page number (PN) will be printed instead of the "#" marker.

PRINTER CODE MARKERS: <F1> <SHIFT>-(and <F1> <SHIFT>-)
The "(" marker sets the start of one or more print codes. They are then terminated by the ")" marker. Refer to Appendix A for the proper key or keys to use to generate the desired numbers.

For example, you would use...

<ALT> E

to generate a number 5. These markers are used primarily for code that is not used very often. If you use the code often, it may be better to redefine one of the print fonts <F1> 1 through <F1> 5 to output the code.

If you need to generate a number 32, do not use the <SPACE>. Instead use a "hard space" character generated by keying...

<F1> <SPACE>

In so doing, the 32 will still be sent to the printer but the "hard space" will not allow additional spaces to be inserted here if justification is set to "both".

GRAPHICS CHARACTERS MARKERS: <F1> 8 and <F1> 9
The <F1> 8 key sequence will place a "begin graphics" character on the screen. It is displayed as a reverse colored "[". All characters entered after this will have an offset of 128 added to the character. For example, the "A" character generates the number 65, and it will have 128 added to that. The result is the number 193 will be sent to the printer.

The <F1> 9 key sequence will place an "end graphics" character on the screen. It is displayed as a reverse colored "]", and it is used to return to normal text characters, after having used the <F1> 8 sequence to print graphics characters.

NEW PAGE MARKER: <F1> <D-ARW>
This marker will generate a reverse colored "F" and marks (forces) the end of one page and the beginning of the next.

Note: Even if you end a page above the bottom margin, footers, headers, and auxiliary lines below the "F" marker will still be printed.

THE VIEW MODE

The VIEW mode is used to view your text on the screen formatted nearly the same as it will be when it is printed on paper. I say "nearly" because there are certain limitations. The Color Computer 3, for example, is not capable of directly printing text in italics or as elongated text on the screen. The only way this is possible is to use a graphics screen and actually draw each character, dot by dot. This is very time consuming as compared to using the computer's 40 or 80 column screen. Other than this, the VIEW mode allows you to view your text on the screen rather than on paper. Not only will this confirm that the printed copy will be what you want, it will also alert you to possible errors that would cause the printing to abort. For example, if you accidentally set the left margin to 75 instead of the right margin.

To enter the VIEW mode, first place the cursor where you want to begin viewing text, and then key...
<CTRL> V

Text will now be displayed on the screen, formatted in the way it will be printed. Margins will be added; headers, footers, and auxiliary lines will be seen; page numbers will be seen if hey are used; and underlining and print fonts will be seen.

The carriage return character (short underline) will appear at the end of each line. If you do not want this character to be shown, you can toggle it "off" (or later "on") by pressing the <ALT> key.

The top line of the screen will display information about the text on your screen. All data at the top of the screen pertains to the status of the print at the time when the last line on the screen was printed.

The page number for the bottom line will be displayed, followed by its line number. Then the left and right margins at the bottom will be displayed.

The following keys are active in the VIEW mode...

<D-ARW>	Move one screen down in the text.
<R-ARW>	Move screen window to the right.
<L-ARW>	Move screen window the the left.
<SHIFT>-<D-ARW>	Move to the next printed page.
<ALT>	Toggle displaying carriage returns.
>	Go to page command.
<BREAK>	Exit view to where you entered view.
<ENTER>	Exit to text at place where viewing

Notice that you can go all directions except up. Once you pass a place in your text, you can not move back up to it.

The "go to page" command is the ">" key. When you key it, (if you are not on the final page) the top line of the screen will wait for you to key in the page number that you want to "go to". After you key in the number, press the <ENTER> key. If a page with that number exists below the current view point, the screen will move to it. If you give it a number larger than exists, the screen will move to the final page.

If, while viewing your text, you want to exit the VIEW mode and be at the same place in your text as where you are viewing, press <ENTER>. If, while in the TEXT mode, you want to move to a certain page number, enter the VIEW mode (Set the "page number" parameter (PN) to the page where you are going to begin viewing.) and use the ">" (go to) command to move to the desired page. Then press <ENTER> to exit at that page.

There are many reasons for using the VIEW mode besides the enormous amount of paper that it may save. For example, you can set the page number at the beginning of your text to 1 and then using the VIEW mode and the "go to" command, find how many pages of text you have by giving it a number that must be larger than the possible number of pages.

PRINTING TEXT ON PAPER

IMPORTANT: Before you send text to the printer, use the VIEW mode to make sure that the text looks exactly as you want it to look.

Once you are sure that your text is as you want it, you can print it out by keying...
<CTRL> P
and answering the "Are you sure?" prompt with "y" for "yes".

PRINTING BEGINS AT THE CURSOR POSITION!

If you key "y" for "yes" and then nothing happens, don't panic! First check and see if your printer is on. If it is, check to see that it is "on line". If this doesn't solve the problem, check the connections between the computer and the printer.

Still doesn't work? If not, abort the printing by keying...
<SHIFT>-<BREAK>

If you are using serial output through an interface that has an adjustable baud rate switch, make sure the switch is set properly.

View your print parameters and make sure the BD (baud rate) value is set properly for your printer. (Use the table in section entitled "PRINT PARAMETERS".)

Check the print parameters and make sure each parameter is correct. For example, if the "invisible print" parameter is set to "y" for "yes", then text will not be output. If this still does not solve the problem, first save your text file to disk, and then return to BASIC where you can check printing using BASIC.

Once printing begins, it will continue until it reaches the end of your text.
If you wish to print out only a section of text, first mark the section that you want printed as a "block", and then use the "block print" command...
<CTRL> B

If something goes wrong and you want to halt the printing, you must use...
<SHIFT>-<BREAK>
Keying the <BREAK> key alone will not stop the printing.

Later you will learn how to both print your text to a printer and at the same time continue editing another piece of text.

PRINTING TEXT TO MEMORY:

If you want to save all or part of your text to disk with it formatted identical to how it would be printed, first print the text into memory. When you do this, a copy of the text will be appended to the end of your existing text, just as if it had been printed. Margins, headers, etc., will be added to the text. Since the text may later be printed using a different type printer, print codes for underlining and fonts will all be filtered out. To print text to memory, key in the following command...
<ALT> <CTRL> P
When printing ends, an alert will sound. You will find the printed text appended to the end of your current text. You can change it if needed, and then save it to disk.

FINDING AND REPLACING TEXT

The "find" and "replace" features add an enormous amount of power to a word processor. With SIMPLY BETTER you can find text characters, carriage returns, and end of page markers. You can even find a particular print font or underlining.

To find a text character, or group of characters, you MUST place the cursor prior to the character(s) that you are trying to find.

DOING A SIMPLE FIND:

To do a find, begin by keying the "find" command...
<CTRL> F

When you do, the top of the screen will read "Find: ". Now, key in the characters you wish to find. After you have them keyed in, press the <D-ARW> key to initiate the search.

If the "mask" is turned off, (no "M" in the upper right corner of the screen) then upper and lower case letters must match exactly. For example, if the characters you key in are...

Color Computer

then every character must match exactly. It will not find...

color Computer

and it will not find...

Color computer

However, if the "mask" is on, (toggle it with <CTRL> <SHIFT>-=) then it will find...

Color Computer

and it will find...

Color computer

and it will even find...

cOloR ComPuTer

USING WILD CARD CHARACTERS:

If you are not sure of how something was spelled, you can use a "wild card character" for characters that you are unsure of. A "wild card character" will match ANY character that it is compared to! For example, say you have used the word "meet" and the word "meat" several times in your text. Now, you are not sure if you spelled it properly each time. You can replace the third letter of the word with a "wild card character" and then the find will locate all instances of "me?t", where the

"?" is the "wild card character". The "wild card character" is generated by keying the <U-ARW> while keying in the "Find: " line. In this case you would key in...

me <U-ARW> t <D-ARW>

Remember that, in this manual, "spaces" are placed before the "<" and after the "<" for clarity. The <U-ARW> creates the wild card character (a reverse colored "?") and the <D-ARW> starts the search.

You can also find characters that are underlined. After keying the <CTRL> F to begin a find, and you are about to key in the characters to find, you can set the print font, or underlining, or both, for the first character of the characters that you are looking for. You do this by first keying...

<F1> U

if the character is to be underlined, and/or key...

<F1> <SHIFT>-#

where "#" is the number of the print font that the first character must be.

For example, if you were looking for "some words", and the "s" of "some" was both underlined and in print font #2, you could do so by first keying...

<CTRL> F

to initiate the find, and then entering the following for "Find:"...

<F1> U <F1> 2some words <D-ARW>

The strings found would be those where each character matches, and where the "s" is both underlined and in print font #2. If you want to find the next word that is underlined, simply key...

<CTRL> F

to begin the find, and for the "Find: ", key in...

<F1> U <U-ARW> <D-ARW>

The "<F1> U" specifies a search for an underlined character. Then the <U-ARW> specifies that the underlined character is a "wild card" character. This will be any character underlined. Finally, the <D-ARW> begins the search.

If you want to find the next text that is of a specific print font, first key in...

<CTRL> F

to begin the find, and then key in...

<F1>

followed by the number of the print font. Then key...

<U-ARW>

to generate a "wild card" and then press <D-ARW> to start the search.

FINDING THE STRING:

If the program finds a set of characters that matches, the line containing the string will be displayed on the text line of the screen and the cursor will be at the first position after the string. You can then continue on with your search for the next occurrance of the string without finding the same string again.

CONTINUING THE SEARCH:

To continue looking for another occurrance, key...

<CTRL> C

This is the "continue" command. If another string is found, the line and the cursor will again be at the top of the screen. If no match is found, a loud beep will sound (if beep is on) and the screen top line will read "None found"

FINDING AND REPLACING TEXT:

To find a set of characters and replace them with something else, first key...

<CTRL> F

to begin the find, and then key in the set of characters that you are looking for. This time, do NOT press the <D-ARW> to initiate the search. This time let the program know that you want to change the characters to something else by keying the <R-ARW>. The top of the screen will then read...

Repl:

Now, key in the characters that are to replace the string of characters. You can include carriage returns, and most any other character that can be part of your text. You can NOT use "wild card" characters here. You can, however, specify what print font that each new character is to be. To specify the print font, (or underlining) use the <F1> and then the particular font number desired. When you finish keying in the text that you want change a string to, key...

<D-ARW>

This begins the find and replace. When a string is found, and if the the verify is on, the line with the string will again be displayed as the first text line on the screen, an alert beep will sound, and you will be prompted with "Change! Are you sure?". If you want the change to be made, key "y" for "yes". If the verify is off, the change will be made without asking if you are sure.

If there is another string that you want to find and replace, just key in...

<CTRL> C

for continue, and the search will again take place along with any changes that are to be made.

To see how some of this works, key in the line...

THIS IS A <F1> 3SIMPLE <F1> 0 SENTENCE. <ENTER>

You should have...

THIS IS A SIMPLE SENTENCE.

with the word "SIMPLE" in print font #3. Now find the word "SIMPLE" and change it a bit. Place the cursor at the beginning of the line and key in the following...

<CTRL> FSIMPLE <R-ARW>

to find "SIMPLE" and to make a change. The top of the screen will now read...

Repl:

Carefully key in this line...

<F1> U <F1> 2SI <F1> 3MP <F1> U <F1> 4L <F1> 0E <D-ARW>

If verify is on, answer the "Are you sure?" with "y" for "yes". What should happen if all was keyed in properly is that the word "SIMPLE" was replaced with the word "SIMPLE" except that now, the "SI" is underlined and in print font #2. The "MP" is also underlined but is now print font #3. The "L" is not underlined and is now print font #4. Finally, the "E" is as it was in the beginning.

FIND AND REPLACE WITH NULL STRING:

You can use the "find" and "replace" to null (erase) a string of characters. You do this by changing the string to a "null string". When you are asked "Repl: ", do not key in any text. Simply use the <D-ARW> for one occurrance, or use the <R-ARW> if you want to do more than one change.

MULTIPLE FIND AND REPLACES:

You can replace a specific number of strings, or all occurrences of a string, or only some occurrances of a string, by using "find" and "replace". First, to change a specific number of occurrences, first enter the string that you want to find followed by the <R-ARW>. Then key in the characters that the string is to be replaced by. Do not key the <D-ARW> as you did above. This time, because you want to make a specific number of changes, key...

<R-ARW>

The top of the screen will now read "Number: ". Simply key in the number of changes that you want to make and press <ENTER>.

IMPORTANT: Unless you are absolutely sure of what is going to be replaced, do the find and replace with the verify mode on. Otherwise, the program may find occurrences that you never realized would match, and change them as well. For example, if you were looking for multiple occurrences of "hat" and replacing them with "cap", the program might find "what" and replace it with "wcap". Had you turned on the verify mode, the program would have showed you each occurrence of a match and would have asked "Are you sure?" before making the change.

If you want all occurrences to be replaced, simply use a number greater than the possible number that can be found. After every "find" and "replace", where a specific number of replacements are requested, upon completion, the program will report (show on the top line) the number of replacements that it was able to make.

While the program is performing a multiple "find" and "replace", you can abort by keying <BREAK>.

Remember, if you do NOT specify a print font in the "find", any set of characters that match, regardless of their print font, will be found. If you DO specify a particular font for the "find", only the text that is of that print font will be found. This includes underlining. In other words, if you want to find all occurrances of "and" that are underlined, and if underlining is specified, an "and" that is both underlined and of a different print font will NOT be found.

LOADING AND SAVING TEXT FILES

FILENAMES AND DEFAULT VALUES:

Filenames follow the same rules as in Disk Basic. Each name can be from 1 to 8 characters. Spaces, commas, slashes (/), and colons are not allowed in the name or the extension. The extension is from 1 to 3 characters. Separate the name from the extension with either a period "." or a slash "/". Separate the extension from the drive number with a colon ":". When you first boot SIMPLY BETTER, the default filename is...

NONAME.TXT:0

You can examine or change the default filename from the COMMAND mode. Use the command...

FN <ENTER>

to view the current default filename. If you want to change all or part of the default filename, follow the "FN" command with that portion that you want to change. If you only wish to change the extension, precede it with a "." (period) or "/" (slash). If you want to change the drive number, precede it with a ":" (colon). Look at this example...

FNmyfile <ENTER>

Here, only the name is changed. The default extension and drive number remain, however the new default name is "MYFILE". Consider this example...

FN/bas

Here, the default name and disk drive number remain the same, however the default extension now becomes "BAS". Look at this example...

FN:2

Here, only the default drive number is changed.

VIEWING THE DISK DIRECTORY:

To view the disk directory, you must be in the COMMAND mode. Then use the "DIR" command followed by the number of the drive with the disk. If you do not specify a drive number, the program will use the default drive number. DO NOT precede the drive number with a colon ":". Look at this example...

DIR2 <ENTER>

This command will display the directory on the disk in Drive #2. Look at this example...

DIR <ENTER>

This command will display the directory on the disk that is in the disk drive with the default drive number.

After entering a "DIR" command, the screen will clear, and the top line of the screen will display the drive number followed by the number of free granules on the disk. On a 35 track disk there are a total of 68 granules available for text storage, and on 40 tracks there are 78 granules for storage. A granule can hold 2304 characters.

Each file on the specified disk will be displayed in double column format. The name and the extensions will be separated by periods simply for clarity. Following each extension will be the number of granules that that particular file occupies on the disk. If there is not enough room on the screen to display all of the files, the screen listing will pause and wait for you to press a key before it continues on to the next screen listing. If you then press <BREAK>, you will exit to the TEXT mode. If you press <ENTER>, you will exit the listing to the COMMAND line and the current screen listing will remain. Any other key will continue on to the next listing.

Once all files have been displayed, the program will wait for your next command. Keying <BREAK> will take you directly to the TEXT mode.

RETRIEVING THE DIRECTORY AS TEXT:

If you want a copy of the directory placed as text at the end of your current file, from the COMMAND mode, key in...

GDIR <ENTER>

The appended listing will be identical to that viewed using the "DIR" command, except the listing will be single column format.
You can specify a drive number, such as...

GDIR2 <ENTER>

The drive number specified becomes the default drive number.

CHECKING MEMORY AND STORAGE SIZE:

If you need to know how many words there are in your text file, regardless of where the cursor is at, just key...

<CTRL> W

When you do, the total number of words in your text will be displayed at the top of the screen.

If you want to find the size of your text file, and also how many granules of disk space are required to hold your text, from the TEXT mode key...

<ALT> <CTRL> S

The top of the screen will then show how much of your buffer is "Used", how much is "Left", and the number of granules that the file requires on a disk.

If the "alternate buffer" is not in use, you can combine it with the "primary buffer" using the command "BOTH" from the COMMAND mode. This will give you the combined buffer area for a text file. The "alternate buffer" can be restored only by using the <ALT> <CTRL> C command which also erases all text.

SAVING TEXT FILES TO DISK:

There are three ways to save a file, or part of a file to disk. Each of these ways are done while in the COMMAND mode. If you want to save ALL of your text to disk, then use the command "FS" followed by the filename. Any part of the filename not given will assume the default values.

Look at our first example...

FSmyfile.txt:1 <ENTER>

Here, the file name is "MYFILE". The extension is "TXT", and the drive number is "1". All of the default parameters are changed by this command.

Before saving the file, the program checks to see if the filename already exists on the disk. If it does, the program asks "Are you sure?" before it saves your text on top of the file that exists on the disk.

Here's another example...

FSmyfile <ENTER>

In this example, text is saved as "MYFILE" and the save is made using the default extension and default drive number.

Consider this example...

FS <ENTER>

Here, the text will be saved using the complete default filename.

SAVING FROM THE CURSOR TO THE END:

The second way of saving will save part of a text file. The command "CS" will save your text from the cursor position to the end of the file. From the COMMAND mode, use the command "CS" followed by the filename. This command is identical to the "FS" command except that it only saves from the cursor rather than from the beginning of your file. Consider this example...

CS <ENTER>

In this case, all text from the cursor position to the end of the file will be saved using the default filename.

SAVING A BLOCK OF TEXT:

The third way to save text is with the block save command. First you must mark the block of text that you want to save. (See section "WORKING WITH BLOCKS OF TEXT") Then, from the COMMAND mode, use the command "BS" followed by the filename to use. The program will show you the first line of the first block at the bottom of the screen. Select the proper block just as you did when using other block commands. When the selection is made, the program will save only the text inside the block.

Look at this example...

BSmyfile

Here, after selecting the proper block, the program will save only the text within the block to disk as "MYFILE" using the default extension and drive number.

THE AUTO SAVE FEATURE:

The "auto save" feature is toggled on or off while in the COMMAND mode by using the command...

AUTO <ENTER>

The line will then show whether it is now "on" or "off".

Its initial state can be set using the CONFIG program. Because of the way it works, *use EXTREME CAUTION* when it is "on"! When "auto save" is "on", a timer begins running. Every time a key is pressed, the timer is reset to zero. If no key is pressed for a period of 5 minutes, the entire text file in memory will be "automatically" saved to disk using the default filename.

The program will NOT ask "Are you sure?" when it finds a file with the same name on the disk. It assumes that you are not at the computer and saves your text right over the top of the file that exists on the disk. Clearly, if you turn it "on", be sure that the default name is correct, and do NOT have any text in memory that you do not want to have saved as that particular file on disk.

To demonstrate one situation that could happen, consider that you have just loaded a file from your disk. You have auto save "on" and the default filename is that of the file just loaded. While working on the file, you delete a section of the text accidentally. You get out the manual to try and find out if there is a way to restore the lost text. While reading the manual, the clock that is running reaches the 5 minute mark and the program automatically saves the existing file over the top of the file on the disk. If the original file contained the lines you accidentally erased, then the auto save just performed erased them permanently.

LOADING FILES FROM DISK:

To load a text file, you must be in the COMMAND mode. Use the command "FL" followed by the filename that you want to load.

Any time you load a file from disk, any text in memory is NOT erased. The file that is loaded is appended to (added to the end) of the existing text. In this way you can load several files, merging each onto the end of the previous, creating one large file in memory.

Again, any part of the filename that is not specified will assume the default value for that part. For example...

FL:0

Here the program will load in the file with the default name and extension from the disk in drive number 0.

If the text file being loaded fills the "primary buffer", and if the "alternate buffer" is not in use, the program will stop loading and ask...

Full! Use alt buffer Y/N?

If you answer "y" for "yes", the program merge the two buffers into one and loading will continue.

If the "alternate buffer" is not available, or if you answer "n" for "no" above, you will then be asked...

Full! Reset & continue Y/N?

If you answer "n" for "no", all loading will be suspended and only that portion of the file loaded will remain in memory.

If you answer "y" for "yes", the program transfers about 250 characters from the end to the beginning of the buffer. Then loading will continue starting at the end of the moved block.

This "primary buffer" is the same as the one that was shown on the first screen in the CONFIG program. If your files continuously fill the "primary buffer" as happened here, you may want to configure a module that you can use that has a larger "primary buffer" area for text.

The "alternate buffer" is also used when you use the "WINDOW" command.

RENAMING DISK FILES:

To rename a file that exists on a disk, from the COMMAND mode, use the "FR" command followed by the filename that you want to to change. As before, any part not specified will assume the default value. Look at the following...

FRmyfile:1

Here the program will look for a file named "MYFILE" that has the default extension and that is on the disk in Drive #1. If the program finds the file, it will respond with...

New name:

Now key in the name that you want the file changed to. Enter only the parts of the filename that you want to change. You can not change the drive number.

IMPORTANT: This command changes the default filename to the new name. If you simply wanted to change the name of a file on the disk but NOT the default filename, you will have the use the "FN" command to change the default name back as it was.

KILLING TEXT FILES ON DISK:

If you want to kill a file on a disk, you must be in the COMMAND mode. Use the command "FK" followed by the filename to kill. Any part of the filename left out will assume the default value. For example...

FKmyfile.dat

This will kill the file named "MYFILE" with the extension "DAT" on the disk in the drive that has the default drive number.

If the program finds the file on the disk, it will respond with "Are you sure?". If you are sure, key "y" for "yes" and the file will be permanently killed. Any other key will cause the program to abort the kill.

The filename killed becomes the default filename.

CREATING, USING, AND SAVING TASKS

What is a "task"? Some word processors refer to these as "macros" or "functions". The word "macro" can be confused with the "macros" used by an assembler, while the word "function" reminds me of an algebraic term. A "task" is simply a set of key strokes that are assigned to a single key. It also includes the number of times that the task is to be performed.

A task can be as simple as this...

<D-ARW>

If the task is only the <D-ARW> key, why assign it to another key? Because a task can be given the number of times it is to be performed. If our task above is to be performed 10 times, then pressing the key assigned the task will result in a <D-ARW> performed 10 times.

Tasks can be any possible combination of keys that you would normally key on the keyboard. Tasks can load files, print files, type words, or do most anything.

Tasks can be assigned to the keys 0 through 9 (ten tasks in all).

CREATING TASKS:

To create a task, from the TEXT mode, key in...

<CTRL> A

This is the "assign a key a task" command. When keyed, the top of the screen will read...

Select task:

Assign the task to a key by keying a number from 0 to 9. The screen will then read...

Keys:

Here is where you will key in the keys that will make up the task. (Maximum of 60 keys in 80 column mode and 28 in 40 column mode.)

At times you are going to want to use one of the arrow keys as part of your task. To use an arrow key as part of your task, first press the <ALT> key. For example, to use the <U-ARW> in you task, key...

<ALT> <U-ARW>

When you do, an "up arrow" will be displayed on your line. If you key...

<ALT> <L-ARW>

a "left arrow" will be displayed on the line. However, if you use the <ALT> <D-ARW>, a reverse colored "up arrow" will be displayed. If you use the <ALT> <R-ARW>, a reverse colored "left arrow" will be displayed. Remember that if the arrow displayed is in reverse color, it stands for an arrow that points in the opposite direction.

At times, you are going to want to use the <BREAK> key in a task so that you can get from the COMMAND mode to the TEXT mode. If you key the <BREAK> key while you are entering the keys for the task, entry will abort and you will be back where you were in your text file. This is why you can use the <D-ARW> in place of the <BREAK> key to exit the COMMAND mode. For example, if your task is supposed to load the default filename and then go to the TEXT mode, you would key...

FL <ENTER> <ALT> <D-ARW>

You can "nestle" one task inside another. That is, you can have one task call (use) another task that was previously created. You can not however have a task call a task that is already running.

When you have all your keys for the task keyed in, press the <D-ARW> key. When you do, the screen will respond with...

Number:

Key in the number of times you want the task to be performed and then press <ENTER>. The task is now created.

EXECUTING A TASK:

To execute a task, you must be in the TEXT mode. First, key the <CTRL> key, and then key the number of the task that you want to execute. If the verify mode is on, you will be asked "Are you sure?". If you key "y" for "yes", the task will be executed. Had the verify mode been "off", the task would simply have executed.

Once a task begins running, you can cause it to abort by keying the <BREAK> key. If you have created a task, (or loaded a task from disk) you can change how many times that the task is to be performed. To do this, key the "<CTRL> A" just as if you were going to create a task. Then key the number of the task that you are changing. After you do this, DO NOT key in any keys as the task. Simply key the <D-ARW> and answer the "Number:" prompt with the new number of times it is to be performed.

SAVING A TASK TO DISK:

To save a task to disk, you must be in the COMMAND mode. First key "TS" as the command and then follow this with the number of the task that you want to save. After the number, key a comma and then the filename that you want to use. If you want to use the default name, leave off the comma and the filename. If you want to use part of the default name, leave that part out just as you did reviously in loading and saving files. For example...

TS1,mytask

Will save the task assigned to "1" using the name "MYTASK". It will use the default extension and the default drive number. If you do not specify a task number, an error will result. If there is no task assigned to the number specified, an error will also result.

Task files are not the same as text files! The task file also includes the number of times that the task is to be performed.

It is advisable to use a file name extension such as "TSK" for task files to easily identify them.

LOADING A TASK FROM DISK:

To load a task, you must be in the COMMAND mode. Use the command "TL" followed by the key (0 through 9) that you want to assign the task to, then a comma, and finally the filename of the task. DO NOT LOAD A STANDARD TEXT FILE! Again, you must assign it a number from 0 through 9 or an error will result. The filename follows the same rules as when you saved the task.

EDITING TASKS

To edit a task, follow the same procedure as when you create a task. First key...

<CTRL>A

as if to create a task. At the "Select Task:" prompt, key in the number corresponding to the task you wish to edit.

The task assigned to that key will now be displayed after the "Keys:" prompt, and the cursor will be at the first character. All keys entered will be inserted at the cursor position. You can use the <R-ARW> and <L-ARW> to move the cursor to the right or left. Use the <SHIFT>-<F2> to delete a character at the cursor. When you are done editing the task, move the cursor past the last character that is to be part of the task and key the <D-ARW>. As before, key in the number of times the task is to execute at the "Number:" prompt, and press <ENTER>.

NEXT NUMBER OUT

To show you how this feature works, first go to the COMMAND mode and key in the line...

NUMBER1,1 <ENTER>

This will be explained in a moment, but for now go back to the TEXT mode and key the command...

<CTRL> N

This is the "next number out" command. When you keyed this, the number "1" appeared at the cursor just as if you had keyed it in yourself. Try keying...

<CTRL> N

This time the number "2" appears. Each time you key the command, the next number in sequence will appear.

Now that you understand this, you can better understand the line where you used the "NUMBER" command. The format for the command is "NUMBER", followed by the next number that you want to have displayed. After this number, key a comma, and finally the number for the increment or decrement. The second number will be an increment if it is positive, and a decrement if you precede it with a "-" minus sign. Look at the example below...

NUMBER100,10

This command directs the next number out to be 100. The increment is 10 and so the second number will be 110, and the third will be 120, etc.. Look at this example...

NUMBER1000,-10

This command directs the next number out to be 1000. The decrement is 10 since the second number was preceded by a minus sign. The second number out will be 990 followed by 980, 970, etc..

At first, this feature may seem to have little use except perhaps for writing BASIC programs. The real power of this feature lies in placing the command inside a task. A very simple application is shown below. First create the task by keying...

<CTRL> A1

Now enter the following as the actual task...

<CTRL> N <F2> <F1> U <SPACE> <F2> <ENTER>

The "<SPACE>" was needed to start the underlining. Key <D-ARW> to terminate entry of the task and then key in...

5 <ENTER>

This is the number of times that the task is to be performed. Now, place the cursor at column number 5 (check the column number at the top of the screen) and key...

<CTRL> T

This places a tab stop here. Move to column 25 and set another tab stop. To check the tab stops, move the cursor to the beginning of line and press the <F2> key twice. The first time the cursor should move to column 5, and the second time it should move to column 25.

The task contains a "<F1> U" for turning on underline, so you must be in the overstrike mode for it to work properly when you execute it.

Go to the COMMAND mode and key in the following to set the "next number out" back to "1"...

NUMBER1,1 <ENTER>

Now place the cursor after any text you have, and at the start of a screen line. Execute the task by keying...

<CTRL> 1

If the verify is on, respond with "y" for "yes".

What should happen is that the screen should now have the following...

1 _____
2 _____
3 _____
4 _____
5 _____

SORTING SECTIONS OF TEXT

You can sort characters, groups of characters, groups of lines, paragraphs, or most anything.

Sorting is done in alphabetical order and characters such as !"#$%&'()*, 'space', etc., come ahead of "A,a,B,b,...Z,z". In this way, an item such as "Louis Bybee" will come ahead of "Louise Bybee".

Before you can do a sort, you must mark the beginning of each section that is to be sorted with a "sort" marker. Other than the first "sort" marker, each "sort" marker also marks the end of the previous section. To generate a "sort" marker, key...

<F1> S

Continue marking each section to be sorted. When you have done this, place a "sort end" marker at the end of the last section that is to be sorted by keying...

<F1> E

Now you are ready to perform the sort.

> ***RULE:*** Sorts begin with the first "sort" marker found at or after the cursor position and continues until it reaches the "end sort" marker.

As the rule implies, move the cursor at or above the first sort marker and then key...

<CTRL> S

This is the "sort text" command. Your text will be sorted in alphabetic order and the "sort" markers will remain in the text. If you want to remove the "sort" markers, place the cursor prior to the first "sort" marker and key...

<CTRL> U

This is the "unmark sort markers" command.

The sort is set up so that if a section begins with say "A", while another section begins with "a", the "A" will come first, followed by "a". Then "B", "b", "C", "c", etc..

If you have an alphabetized list of items that you keep on disk that you continually add items to, you may want to save the list with the "sort" markers left in. In this way, when you add a new item to the list, place a "sort" marker followed by the new item at the top of the list. Then sort the list and the new item will be placed where it belongs. Save the list back to disk. When you print the list, remove the sort markers using...

<CTRL> U

This is the "unmark sort markers" command and it removes all "sort" and "sort end" markers at or after the cursor.

Although the sort routine will seem slow, an enormous amount is done other than

just sorting the text. Each line has to be set up so that when you sort the text, all of the print fonts and underlining will be maintained properly. To see this better, try the following example by first keying in...

acegikmoqsuwybdfhjlnprtvxz <ENTER>

Now get in the insert mode. Place the cursor on the "a" and key...

<F1> U <F1> 3

This will change the line to underlined print font #3. Now move the cursor to the "b" and key...

<F1> U <F1> 1

The last half is now print font #1 and not underlined.

Now sort each character in the line. To save a lot of time, create a task that will place the "sort" markers in the line for you by keying...

<CTRL> A1 <F1> S

to begin creating the task, and to insert a sort marker. Then key...

<ALT> <R-ARW>

to move to the next place for a sort marker, and then key...

<D-ARW> 25 <ENTER>

to exit and set the number of executions to 25. Now, while in the insert mode, place the cursor at the "c" and key...

<CTRL> 1

If the verify mode is on, answer the prompt with "y" for "yes". You should now have a "sort" marker after each letter except the "z". Now, insert the "sort end" marker by keying...

<F1> E

You can now sort the line by keying...

<CTRL> S

When it has finished sorting, remove all of the markers by keying...

<CTRL> U

Now look at the line! Every other character is underlined, and every other character is a different print font. If these print fonts and underlining are configured for you printer, (as well as your baud rate, etc.) try printing the line.

Now, perhaps you can understand why it takes a little time. Realize that even though the unsorted and sorted lines are of the same length on the screen, the text file for the second sorted line is actually 4 times as long due to the number of hidden codes that are required to turn on and off each font and underlining.

THE "WINDOW" - A SECOND WORD PROCESSOR

WORKING ON TWO SEPARATE FILES:

If you want to work on two separate files at the same time, create a second window by keying (from the command mode)...

WINDOW <ENTER>

The only time you can not have this "window" is when "alternate buffer" has been allocated for some other purpose.

When you enter the "WINDOW" command, a reverse colored "bar" will be drawn across the screen just below center. The "bar" has numbers on it so that it can also serve as a "column" ruler.

You can specifiy a number from 4 through 20 immediately after the word "WINDOW" for the initial line number to place the horizontal bar. The command...

WINDOW6 <ENTER>

creates a window with the horizontal bar across the screen at line number 6.

Now, the command line will change to...

AltCmd: N L

The command line reads "AltCmd" if you are in the "alternate" window, and it reads "PriCmd" if you are in the upper "primary" window.

Exit the COMMAND mode by pressing <BREAK> and then move to the upper "window" by keying <CLEAR>. (The <CLEAR> key is used to move back and forth between windows.)

You actually have two separate word processors up and running!

To show this, key in the following line...

This line is the text that that is in
my original word processor. <ENTER>

Goto the COMMAND mode and key in...

WIDTH25 <ENTER>

Key <BREAK> to return to the TEXT mode.

Move to the lower word processor by keying...

<CLEAR>

You are immediately in the other word processor. Now type in...

This line is not part of the line
in the other word processor. <ENTER>

Notice that the screen width here is independent of the screen width that was set in the first word processor. You can be in the insert mode in one window and in the overstrike mode in the other window.

Check the memory here in the second word processor by keying...
<ALT> <CTRL> S

Go back up to the first word processor by keying...
<CLEAR>

Check the memory here. Notice that each has its own text storage area. Other things that you will notice is that each has its own separate filename. In this way, if you load a file into each word processor, you will not have to remember what each one is named.

Now that you realize that each is a separate word processor, I am going to change how I refer to each. From now on I will refer to the first as the "primary window" and the second as the "alternate window".

You can clear all text from the "primary buffer" and use only the "alternate buffer" if you'd like.

RULE: The PRIMARY window must exist at all times.

In other words, you can not create the alternate window and then kill the primary window. You can, however, copy all text from the alternate window to the primary window, and then kill the alternate window. (This assumes that you have enough memory in the primary window.)

To eliminate (kill) the alternate window, simply repeat the "WINDOW" command from the COMMAND mode while you are in either window. If you have changed the default filename for the "alternate window", this name will still be the default name for the "alternate window" even after you eliminate the window. The next time you create the window, it will still have the name when the window was eliminated.

If you are working with both windows active, you can change the size of the window by moving the "bar" up or down. To move the "bar" down one line, key...
<CTRL> <SHIFT>-+

Remember that after you do this once, using the <SHIFT>-<ALT> will speed things up if you want to move the "bar" faster.

To move the "bar" up, key...
<CTRL> -

As an example of how useful the alternate window can be, say you were in the middle of typing in a letter to someone and then found that you needed some information that you sent them in a prior letter. The letter that you sent them earlier is in another file. The solution should be evident. Simply create the alternate window, load in the second letter, move to the information you need, go to the primary window and key in the information as you see it in the alternate window, and then kill the alternate window using the "WINDOW" command again. (Later you will learn how to copy blocks of text from one window to the other.)

Although each window is a separate word processor, they are "linked" together in several ways. They both share the same "locate" and "change" buffers. They

always are in the same verify mode (changed with <ALT> <CTRL> V). The tasks are common to both windows so that one task can move back and forth between windows.

If you are working on two separate files and you are looking for text that exists in both files, after doing a "find" in one window, simply go to the other window and key...
<CTRL> C

This will continue on with the same "find" (or "find" and "replace").

Another way they are linked together is by the "get block" command. Now you can "get" a block of text from one window and transfer it to the cursor position in the other window. You do this by first marking the block that you wish to copy. (The "get" is actually a "block copy" as compared to a "block move".) After you have marked the block, go to the other window and place the cursor where you want to copy the block to. Then key...
<CTRL> G

This is the "block get" command. As with "block copy", the first line of a block will be displayed (if one exists) at the bottom of the screen. Key "y" for "yes" if this is the proper block, else press "n" for "no". When you find the proper block and press "y", the block will be copied to the cursor position in the opposite window.

USING THE SAME FILE IN BOTH WINDOWS:

You can work on two separate files as mentioned above, or you can use the "get block" command to copy you entire file to the second window. This is sometimes helpful for viewing one section of your text while working on a different section.

AUTOMATIC PRINT SPOOLING

Print spooling is easy to accomplish. Since you can create a separate word processor at any time, use one to print your file while you work on other text in the other window. Before you begin printing, create the alternate window using the "WINDOW" command.

Initiate the printing as you have done before by keying...
<CTRL> P

When printing begins, use the <CLEAR> key to change to the other window. You can now load, work on, and save other files while your other text is being printed. The "spooling" is designed so that it's nearly impossible for the program to miss a key pressed as you are keying text while your text is being printed in the other window.

You can not modify text in the same window where you are printing. You can type keys, but they will be placed in the "type-ahead" buffer and will not be processed until after printing ends. The one key that will be processed when it is keyed, is the <SHIFT>-<BREAK> key that aborts printing. This can be keyed to halt printing regardless of which window you are in, or regardless of which window printing is being done. The <CLEAR> key which will move you from one window to the other will not work properly if you have keyed in text while in the window where printing is happening. To erase this text so that the <CLEAR> key can be processed, press <BREAK> and then press the <CLEAR> key.

A short note about print speed. When you are in the same window that is printing, the program realizes that you can not work on the text while it is being printed. For this reason, the program places a high priority on the job of printing and less priority on checking the keyboard. You may have to type much slower than normal. When you move to the window opposite to where the text is being printed, the program will give a higher priority to the keyboard, and you will detect a slow down in printer speed. If you are in the window opposite the printing and have no work to do, move to the window where text is being printed so that printing will speed up.

HANDLING PRINTER ERRORS:

If a printing error occurs while you are in the same window as the printing, an error message will be displayed at the top of the screen, the screen will move to where the error occurred, and the cursor will show where the error is in the line. If an error occurs in the printing while you are in the other window, you will not be interrupted by the error. You may notice that the printer has stopped, however the other window will wait until you return to it to let you know about the error.

COMBINING BOTH BUFFERS:

If you are not using the "alternate buffer" as a window, you can combine both the "primary buffer" and the "alternate buffer" into one buffer. This will give you the combined buffer space for a single file. To combine buffers, you must be in the COMMAND mode, and then key in...
BOTH <ENTER>
Once you have combined these buffers, you can separate them back as they were by using the <ALT> <CTRL> C command while in the TEXT mode, or by using the NEW command while in the COMMAND mode. If you do this, all text will be erased.

PRINT FILLING A FORM

If you have a pre-printed form that you fill in with text periodically, then you may want to use this feature.

If you were to place a pre-printed form in you printer, you would probably find it very difficult, if not impossible, to use a word processor to fill in only the blank place on the form. This is the reason that "print filling" was added as an additional feature.

SETTING UP PRINT FILLING:

The first thing to do is to get a few extra copies of the form to experiment with. Check the distance between lines with a ruler and make sure that your printer is capable of advancing from line to line using carriage returns. If your printer can not, don't despair, there may still be help, so read on.

Assuming that your printer can advance properly, look at one of the forms and try to recreate it on the screen. The lines need not read exactly the same as on the form since you will not be printing out the form, you will be printing the information into the "blank" areas.

As you recreate the form, each time you come to a "blank" area that will later be filled in, use underlining for this area.

IMPORTANT: DO NOT USE UNDERLINING EXCEPT WHERE BLANKS ARE TO BE FILLED IN!

What will happen when you print out your text is that if a character in your text is not underlined, a "space" will be sent to the printer instead of the character. If the character is underlined, then only the character will be sent to the printer. Carriage returns will always be sent so that your form will advance to each line properly.

After you have recreated your form, (or part of it) change the parameter "PF" from "n" to "y" for "yes". Do this on the command line using...
/pfy <ENTER>

or imbed it in your text by keying...
<F1> /pfy <ENTER>

Now fill in the underlined areas with x's or something. Be sure the areas remain underlined.

Estimate and set your top, left, and right margins. Place the cursor at the beginning of the form and try printing it out. After printing it, remove the form from the printer and examine how well you filled the blanks. Adjust your text until each blank space fills in properly. You can also print the text on a blank sheet of paper, and then overlay the paper on the form and hold it up to a light to see how well you matched the lines on the form.

Once it is working correctly, VERY CAREFULLY (in the overstrike mode) key "spaces" into each underline area. Now save this out as a disk file. You may wish to give these files a unique extension for easy recognition.

Now, each time you want to fill in one of the pre-printed forms, just place the form in your printer as you did here. Then load the proper file, fill in the blanks, set the "PF" parameter to "y" and "print fill" the form.

CAUTION: Be sure and reset the "PF" parameter to "n" when you are finished, or you may get some startling results when you send the next text to your printer.

If your printer can not advance properly to each line, see if you can manually advance it line by line. If you can, place "print pause" markers at the beginning of each line that contains text that will be printed on the form. This way, printing will halt and allow you to move the paper to the proper line.

CREATING A TABLE OF CONTENTS

In this section you will learn how to create a table of contents for a text file that you are currently keying in, or for one that you have previously saved.

> ***RULE:*** You can not create a table of contents if you are creating an index (the IC parameter must be 0), or if you are performing a mail-merge.
>
> ***RULE:*** You can not turn on indexing inside your text if you are creating a table of contents.
>
> ***RULE:*** You can not use the VIEW mode if the "TC" parameter is other than 0.

It is highly recommended that you first create your text file and save it to disk BEFORE you set it up for creating the table of contents.

INSERTING TABLE OF CONTENTS ITEMS:

Each item that is to appear in the table of contents must be inserted into your text, preceded by a "table of contents item" marker and terminated with a carriage return. The marker, item, and carriage return are not part of the normal text. You generate a "table of contents item" marker by keying...

<F1> T

Look at the following example...

To print <F1> TPrinting text <ENTER>
your text on a printer...

Because the text and the carriage return are not part of the normal text, these lines will print out as...

To print your text on a printer...

The words "Printing text" will appear in the table of contents, followed by the page number where it exists.

> ***RULE:*** Always place the "table of contents item" marker after a section of text that is sure to be on the same page number as you want the item to have.

For example, it you wanted to have "Chapter Two" in your table of contents, and if chapter two began with...

Prevent incorrect numbering by...

then inserting the table of contents item as shown below can cause problems...

<F1>TChapter Two <ENTER>
Prevent incorrect numbering by...

This is because the page number at the marker may not yet have been incremented

to the page number that the "Prevent..." is on. When this happens, your table of contents will not be numbered properly.

As seen here, inserting items into your text can cause it to look rather "broken up". It is not easy to tell at a glance which is the normal text and which is the table of contents item. This is why I recommend that you first create your text, and then later add the items.

To insert a table of contents item after you have created your text, first get into the insert mode. Then place the cursor where you want the item inserted and key...

<F1> T

followed by the item and terminate it with a carriage return.

GENERATING THE TABLE OF CONTENTS:

You can generate a table of contents while you are printing out the text on paper, or you can use the "invisible print" feature to generate the table of contents without printing the text on paper. (Set "IP" to "y" for "yes".)

After you print the text, the newly created table of contents will not be printed on paper. This is because you may want to go back and change its appearance or change the margins, etc..

To generate the table of contents, first decide upon the column number for the page number. The page number will be placed after each item in the table of contents. Make sure the column number is larger than the longest table of contents item. If the column number you select is larger than the screen width setting, the line will wrap on your screen until you set the screen width larger.

The column number for the page number is actually the column number for the beginning of a 5 digit number plus a carriage return. It is not the column number of the first digit in the page number.
In this way, the page numbers will be right justified as shown here...

```
Chapter One     5
Chapter Two     36
Chapter Three   845
```

Next, set the "TC" parameter to the column number selected. This both sets the column number and turns on the feature. You must later set the "TC" parameter to 0 to turn it off.

All that is left now is to print your text. If you don't want to waste paper by printing out your text, set the "invisible print" parameter (IP) to "y" for "yes". Place the cursor where printing is to begin, and key...

<CTRL> P

Answer "Are you sure?" with "y" for yes and printing will begin.

After printing stops, at the end of your text you will find the newly generated table of contents. If you want it to be at the beginning of your text, mark it all as a block and use the "block move" (<CTRL> M) command to move it. You can now make any additional changes to set it up the way you want.

CREATING AN INDEX

Here, you will learn to create an index for a text file that you are working on, or for one that you have previously typed and saved.

There is only one essential difference between a "table of contents" and an "index". The "table of contents" is in the same order that each item was when it was in the text, while the "index" is in alphabetical order. Thus, the only difference in creating an index as compared to creating a table of contents is that the program must sort the index.

Mark an "index item" in the same way you marked a "table of contents item" above, except use an "index item" marker. To generate an "index item" marker, key...

<F1> I

The index item and the carriage return that terminates it, are not part of the actual text. If you simply marked off words in your text as words to be included in your table of contents, you couldn't control whether the words in the index were to be all lower case letters, etc., without changing your actual text.

To create the index, set the "IC" parameter to the column number for the page number the same way you did with the "TC" parameter for the table of contents. Then print your text on paper, or set the "IP" parameter to "y" to turn on "invisible printing". After you print your text, the index will be found at the end of your text. Modify it to look the way you want, as you did with the table of contents earlier.

The very last thing you do before printing or saving the index is to always check for duplicate items. They will be easy to find since the alphabetizing groups then together. If this occurs, eliminate one of the items and put its page number along with the page number of the other item.

You can alter the index to how ever you want it to be printed. Then save it to disk or print it on paper.

PERFORMING MAIL-MERGES

A "mail-merge" does not necessarily have a thing to do with "mail", however many times it is used to generate numerous copies of a letter that is to be sent to different people at different addresses.

A "mail-merge" is where items from a list containing groups of items are "merged" into a specified file, and then the combined text is then sent to a printer. The procedure is repeated for each group of items in the list, until the items from the last group in the list have been merged into the specified file and printed. If this sounds complicated, don't worry, you'll find it simpler than it sounds.

THE MAIL-MERGE LIST:

Let's say that you want to send a copy of a letter to twenty members of a club, informing them of a meeting that they are to attend.

The first thing you will need is a list of all twenty members in the club. Information in the list should contain not only their names, but also their addresses. The list can contain numerous other items if you want, even if the items are not going to be used in this particular letter. Each item in the list is terminated by a carriage return (the carriage return is not part of the text or the item).

Look at the sample list below...

Smith <ENTER>
John <ENTER>
1253 N. Summer St. <ENTER>
Sumtown <ENTER>
OR <ENTER>
97000 <ENTER>
Color Computer 2 <ENTER>
Doe <ENTER
John <ENTER>
925 South St. <ENTER>
Sumtown <ENTER>
OR <ENTER>
97001 <ENTER>
Color Computer 3 <ENTER>

Here, each "group" in the list contains 7 lines.

RULE: Each group in a list must always contain the same number of lines.

The persons last name and first name are each individual lines (items) in the list. In this way, you can use their first name only, as in...

Dear John

or combine the first and last names as in an address...

TO: John Smith

At the top of the list, place a "list" marker, followed by the "matrix size" (number of lines per group), and terminate this with a carriage return. This must be at the very beginning of your list. To generate the "list" marker, key...

<F1> L

For the above list, the top line would be keyed as...

<F1> L7 <ENTER>

The "<F1> L" places the marker on the line, the "7" is the matrix size, and the "<ENTER>" terminates the line.

When you have created your list with the "list" marker and matrix at the top, save your list as a file to disk.

THE MAIL-MERGE TEXT:

The "mail-merge text" is the text that is to be printed with the merged lines from the list. In the above example, this will be the letter that is to be sent to each member of the club.

Create your text as you normally would. If you want to merge (insert) a line from the list into your text, first determine the line number that the item is in each group of the list. In other words, if the 4th item in each group is the item that you want, then the number you want is "4". Next, mark where the item is to be inserted with a "mail-merge item" marker. You generate it by keying...

<F1> M

Type the number of the item after the marker, and then place a second "mail-merge item" marker after the number. If the item were number 4 , you would type...

<F1> M4 <F1> M

NOTE: You may want to place a "new page" marker (generated by keying <F1> <D-ARW>) at the end of your text so that each page printed will advance to the proper place. After you have created your text, save it as a file to disk. You may want to use a unique extension for the file, such as "SPF" for "standard print form" for easy recognition.

PERFORMING A MAIL-MERGE:

To perform the mail merge, first load in the form that is to be printed. Next, load in the list that is to be used. The list must be the final text in the buffer. Place the cursor at the beginning of the form and key...

<ALT> <CTRL> M

When you do, a copy of your text file will be printed using text from the first group of lines in the list. After the text has been printed, the program checks to see if another group of lines is present in the list. If so, another copy of your text is printed with the text from this group. This continues until it has printed the text using the last group of lines in the list.

NOTE: If you want to have a copy of the text file WITH the merged text in it, use the <SHIFT>-<BREAK> to halt the printing. If this is done, the text as it is being printed will be found at the end of your text in memory.

NOTE: If you abort printing during a mail-merge, you will need to erase the copy of the text that was being printed. You'll find it appended to the end of your list

CRAY-O-LATOR "CALCULATOR"

The "Cray-O-Lator" is a four function (+ - * /) calculator that can be used to perform floating point arithmetic. It is capable of displaying numbers in either decimal (base 10), binary (base 2), or in hexidecimal (base 16). You can also store numbers for use later.

To "pop-up" the Cray-O-Lator, use the following TEXT mode command...
<ALT> <CTRL> O

When you do, the top line of the screen will display the name "Cray-O-Lator", followed by a reverse colored area for displaying numbers. This is then followed by "Dec" which lets you know that you are currently in "decimal" (base 10).

You can change bases at any time. If you key the "%" key, you will change to binary (base 2) and the line will display "Bin". If you press "$", you will change to hexidecimal (base 16) and the line will display "Hex". If you press "T", you will return to decimal (base 10) and the line will again display "Dec". As mentioned, you can change bases at any time, even in the middle of keying in a number! When you change bases, the number in the display is converted and displayed using the new base.

If you press the "+", "-", "*", or "/" key, the same key will be displayed to the right of the base ("Dec" if you are in base 10) to let you know what operation you keyed.

If you press the "S" key, the number currently being displayed will be "saved" for use later. If you press the "G" key, you will "get" the last number saved. If a number has been saved, an "M" will be displayed to let you know a number is in "memory".

The <ALT> key can be used to toggle the number from negative to positive, or positive to negative.

The <CLEAR> key performs a "CE" or clear entry the first time it is pressed. This can be done if you want to begin the entry of a number over again without loosing the calculation being performed. If you press <CLEAR> twice, the entry and all calculations are reset to 0.

The <BREAK> key is used to exit the Cray-O-Lator.

The following is an example of a simple calculation...

Key(s) Pressed	Displayed
(none)	0 Dec
8	8 Dec
<ALT>	-8 Dec
+	-8 Dec +
6	6 Dec
=	-2 Dec

Here's another example that stores a number and then uses it again later...

Key(s) Pressed	Displayed
(none)	0 Dec
3.14	3.14 Dec
S	3.14 Dec M
*	3.14 Dec M*
2	2 Dec M
=	6.28 Dec M
G	3.14 Dec M
+	3.14 Dec M+
1	1 Dec M
=	4.14 Dec M

Constant Number: When you key in a number, that number is moved to the constant number area. When you press "+" or "-", all pending operations are completed and the result will be displayed as well as moved to a constant number area. When you press "*" or "/", the constant is not updated until a "+", "-", or "=" is pressed, at which time, the result of the pending "*" and/or "/" operation(s) are completed. The result is treated as if it were a number keyed in and moved to the constant number area. When the "=" is pressed, the constant number is used as the second number of the operation to be performed using the previous operation. The result of this will then be used as the first number, and the constant as the second number if the "=" is pressed again.

To demonstrate this, look at this example...

Key(s) Pressed	Displayed
(none)	0 Dec
4	4 Dec
+	4 Dec
=	8 Dec
=	12 Dec

Here, when the "+" was keyed, the result of pending operations is still 4. The 4 is moved to the constant area. Next, when the "=" is pressed, the constant (4) is added to the first number (4+4), and the result is displayed as 8. The next time the "=" is pressed, the constant (4) is added to the number 8 and results in the number 12. Here's another example...

Key(s) Pressed	Displayed
(none)	0 Dec
3	3 Dec
+	3 Dec +
10	10 Dec
/	10 Dec /
5	5 Dec
=	5 Dec
=	7 Dec

In this example, when the first "=" is pressed, the result of the pending "/" operation (10/5=2) is performed and the 2 result is moved to the constant area. This is then used to perform the pending operation with the 3. The 3+2 results in 5. The next

"=" uses the result (5) and the constant (2) along with the previous operation (+), or 5+2, resulting in the final number 7.

CHANGING BASES:

As mentioned above, you can change bases at any time. You must, however, be aware of the ranges for each base. If you change to a base that can not display a number as large as the one needed, a "Overflow" error will occur, and the number will be lost. Refer to the table below for number ranges.

You can not use decimal points while in binary (base 2) or while in hexidecimal (base 16).

IMPORTANT: If you have a number with a decimal point displayed while in decimal (base 10), changing to binary (base 2) or hexidecimal (base 16) will not loose the digits to the right of the decimal point UNLESS you continue to key in digits as part of that number.

Here's an example that will demonstrate changing bases...

Key(s) Pressed	Displayed
(none)	0 Dec
3.14	3.14 Dec
$	3 Hex
+	3 Hex +
F	F Hex
=	12 Hex
T	18.14 Dec

Note that because we did NOT key in any further digits into the number 3.14 after converting to hexidecimal, the ".14" was not lost.

RANGES FOR EACH BASE (Displayed in decimal):

Binary (base 2)..... -4095 to 4095
Decimal (base 10)..... -999999999 to 999999999
Hexidecimal (base 16).. -268435455 to 268435455

Remember that the number -1 will be displayed as $FFFFFFF in hexidecimal, and as 111111111111 in binary. Also, -268435455 will be displayed as the number 1 in hexidecimal, and -4095 will be displayed as 1 in binary.

RETRIEVING NUMBERS WHILE IN THE TEXT MODE:

If you have saved a number while using the Cray-O-Lator, after you return to the text mode, you can "get" the saved number to the cursor position (as if you had keyed it in) by using the "get number" command ...

<ALT> <CTRL> G

This saves you from having to remember or write down a number that you may later want, as well as saving you from having to key it in. Remember to be in the insert mode if there's a chance of having the number erase text at the cursor position.

GETTING HELP INSTANTLY

Help can be obtained instantly for each of 4 general areas...

1 TEXT mode help
2 COMMAND mode help
3 <CTRL> command help
4 <F1> command help

The first time you obtain help, the disk in the default drive MUST contain the help file named HELPFILE.TXT. The program will load in this file, and it will not need to load it again until the next time you boot the program.

Help pertaining to the TEXT mode is available while you are in the TEXT mode by keying...

<CTRL> /

Help with COMMAND mode commands is available while you are in the COMMAND mode by keying...

<SHIFT>-? <ENTER>

Help with <CTRL> key commands is available while you are in the TEXT mode by keying...

<CTRL> <SHIFT>-?

Help with <F1> key commands is available while you are in the TEXT mode by keying...

<F1> <SHIFT>-?

Help is instantly displayed in a reverse colored overlay on your screen. The "help" listing will not fit in one overlay, so after each listing, the program will wait for you to hit a key before continuing to the next. If the key hit is <BREAK>, the help window disappears, and you will be back where you were when you sought the help. Any other key will bring up the next help listing, or when all has been displayed, the window will disappear and you can continue on.

The help in these overlays is condensed and is only to refresh your memory about what you should already know. It isn't intended to teach you about what is covered in the manual.

EXIT TO BASIC

You can exit to BASIC from the COMMAND mode by keying...

BYE <ENTER>

When you do, the program will ask "Are you sure?". If you are, key "y" for "yes" and the program will return you to BASIC.

Section 2

Expanded Command Reference from "A" to "Z"

COMMAND MODE COMMANDS

The commands listed in this section must be keyed in while in the COMMAND mode. To get to the COMMAND mode from the TEXT mode, key...
<CTRL> <CTRL>

When you first enter the COMMAND mode, the screen will look as shown here...

PriCmd: N L

The cursor will be blinking after the word "COMMAND:", and the program will be waiting for your next command to be entered.

To exit the COMMAND mode to the TEXT mode, key <BREAK> or <D-ARW>.

Text keyed in on the COMMAND line can be edited as follows...

(1) Text characters are inserted to the left of the cursor position.
(2) Use the <SHIFT>-<F2> to delete the character at the cursor position.
(3) Use the <R-ARW> and the <L-ARW> to move within the COMMAND line
(4) Text at and after the cursor is not used when the line is processed.

Each COMMAND mode command is described below.

AUTO:

This command will toggle the "auto save" feature on or off each time it is used. After you enter the command, the top line of the screen will display either "AUTO ON" or "AUTO OFF" to let you know its current status.

When the auto save is on, the program times the length of time since the last key was pressed. Each time you press a key, the timer resets and starts again. If the timer reaches 5 minutes, the program will automatically save your text file to disk using the default filename. The program will assume that you are no longer at the keyboard, and will not ask "Are you sure?" if it finds a file of the same name on the disk. The program will overwrite a file on a disk if the file on the disk has the default filename.

The auto save feature must be used with caution! Because the auto save is performed without verification, it can destroy a file on a disk that shouldn't have been overwritten. Any time you turn on "AUTO", remember to check the current filename and change it to the name that should be used when the program automatically saves your text.

When the program boots, the default filename is "NONAME.TXT". If you only use

this name for the filename to be used with "auto save", it could prevent you from destroying a file accidentally. By using this name, when the program is first booted, it will use "NONAME.TXT" if it auto saves a file. If you later change the default filename, be sure that the new name is a file that can be written over, or be sure to turn off the "auto save" feature.

BEEP:
If your system has audio output (sound) and if your program was configured with "BEEP ON", then the program will sound a short "key beep" when you press a key. It will also sound a longer "error beep" if an error occurs. When the program needs to alert you for some reason, it will sound two short "alert beeps". To turn the beeps on or off, use the "BEEP" command. After you enter the command, the top line of the screen will show the word "BEEP" followed by either "ON" or "OFF" to let you know what the current status is.

If you have a hearing impairment, or if your system has no audio output, you may want to use the "EYES" command described below.

BOTH:
Use this command to combine the "primary buffer" with the "alternate buffer". If the "alternate buffer" is not being used already, when you enter this command the program will display "BOTH BUFFERS MERGED" on the top line. The purpose of the command is to create the maximum size buffer available for a text file. (See the "WINDOW" command for more information on "primary" and "alternate" buffers.) After you have "merged" both buffers together, the only way to change them back into separate "primary" and "alternate" buffers is to use the TEXT MODE command <ALT> <CTRL> C. This will also erase all text.

BS: This is the "block save" command. The format for this command is...
BSfilename <ENTER>

where "filename" is the name, extension, and drive number to be used in the block save. If you omit any of these three items, the program will use the current default value.

The filename used becomes the default filename!

If you omit the name, precede the extension with a "." (period). If you omit the extension, precede the drive number with a ":" (colon). If you use only "BS", the program will use the entire default filename. There are filename examples at the beginning of section 1.20 "LOADING AND SAVING TEXT FILES". After you enter this command, the program will search your text for the first block it can find. If no block is found, an error will occur, otherwise the program will display the first line of the block found on the bottom line of the screen. If this is the correct block, answer the prompt with "y" for "yes", otherwise use "n" for "no" and continue until the correct block has been selected. When selected, the text in the block is saved to disk.

BYE:
This command is used to exit SIMPLY BETTER and return to BASIC. Be sure that you have saved your text to disk before you exit.

CLEAN:
This command will "clean" the text buffer of all markers that turn on or off underlining, and all markers that turn on print fonts. (Print fonts are only turned on, never off.) If you turn off underlining or change a print font for a section of text, the program may not remove the "hidden" codes that the program uses to mark the end of the underlining and/or print font. These hidden codes may later cause a problem if the text file is used by a program that does not recognize these markers. The CLEAN command will erase all such markers from your text.

CS:
This is the "cursor save" command. The format for the command is...
CSfilename <ENTER>
where "filename" is the name, extension, and drive number to be used in the save. If you omit any of these three items, the program will use the current default value.

The filename used becomes the default filename! If you omit the name, precede the extension with a "." (period). If you omit the extension, precede the drive number with a ":" (colon). If you use only "CS", the program will use the entire default filename. There are filename examples at the beginning of section 1.20 "LOADING AND SAVING TEXT FILES".

This command is identical to the "FS" command listed below, except that this command saves all text from the cursor to the end of text. It does not save any of the text prior to the cursor.

DIR:
This is the same as BASIC's "DIR" command. The format for the command is...
DIRn <ENTER>
where "n" is the number of the drive to be used. If the number is omitted, the default disk drive number will be used. The drive number used will become the default drive number!

When you enter this command, the screen will clear and display a listing of the files on the disk in the specified disk drive. If there are more files than can be displayed on a single screen, the program will wait for a key before it displays the next listing. Pressing <BREAK> will move you to the TEXT mode, pressing <ENTER> will exit back to the COMMAND line without changing the display, and any other key will continue on to the next screen listing.

EYES:
If you have a hearing impairment and you can not hear the "error beeps" or the "alert beeps", (or if you have no audio output) you can use this command to turn on a "border flash" feature. When the program normally does an "alert beep" or an "error beep", the screen's border will flash to get your attention. When you enter the command, the top line of the screen will display either "EYES ON" or "EYES OFF" to let you know what the current status is. To see this feature work, first turn it on and then enter some characters on the command line that can not possibly be a valid command. When you press <ENTER>, the border will flash as an error warning.

FK:
This is the "file kill" command and it is used to kill a file on a disk. The format for the command is...

FKfilename <ENTER>

where "filename" is the name, extension, and drive number of the file to be killed. If you omit any of these three items, the program will use the current default value.

The filename used becomes the default filename! If you omit the name, precede the extension with a "." (period). If you omit the extension, precede the drive number with a ":" (colon). If you use only "FK", the program will use the entire default filename. There are filename examples at the beginning of section 1.20 "LOADING AND SAVING TEXT FILES".

If the program finds the file on the disk, it will prompt you with "Are you sure?" before killing the file.

FL:
This command is used to load a text file from disk. The format for the command is...

FLfilename <ENTER>

where "filename" is the name, extension, and drive number of the file to be loaded. If you omit any of these three items, the program will use the current default value.

The filename used becomes the default filename!

If you omit the name, precede the extension with a period. If you omit the extension, precede the drive number with a colon. If you use only "FL", the program will use the entire default filename. There are filename examples at the beginning of section 1.20 "LOADING AND SAVING TEXT FILES".

After entering the command, the text file will be loaded into memory starting at the end of your current text. By starting at the end of any text in memory, you are able to load one file after another, with each new file appended to the end of the previous file.

If the text buffer becomes full while text is being loaded, the program will temporarily stop loading the file. If the "alternate buffer" is not in use, you will be prompted with...

Full! Use alt buffer Y/N?

If you answer the prompt with "y" for "yes", the program will merge the "alternate buffer" with the "primary buffer" to gain space and loading will then continue where it left off. If you answer with "n" for "no", or if the "alternate buffer" was already in use, the program will prompt you with...

Full! Reset & continue Y/N?

If you answer this prompt with "y" for "yes", the program will first move the last 512 characters loaded to the beginning of the buffer, and then it will "reset" its load pointer so that loading will begin at the end of the text moved. If you answer the prompt above with "n" for "no", loading will be halted and all text loaded will remain in the buffer.

FN:
This is the "file name" command. Its format is...

FNfilename <ENTER>

where "filename" is the name, extension, and drive number that is to be substituted into the default filename. If you omit any of these three items, the program will not change that part of the default filename. If you omit all of the filename and only use "FN", the program will display the current default filename.

If you omit the name, precede the extension with a "." (period). If you omit the extension, precede the drive number with a ":" (colon). There are filename examples at the beginning of section entitled "LOADING AND SAVING TEXT FILES".

FR:
This is the "file rename" command. The format for this command is...

FRfilename <ENTER>

where "filename" is the name, extension, and drive number of the file on disk that is to be renamed. If you omit any of these three items, the program will use the current default value.

If you omit the name, precede the extension with a "." (period). If you omit the extension, precede the drive number with a ":" (colon).

After you enter this command, the program will search the disk for a program with the same name. If it finds one, it will then ask...

New name:

Key in the filename that you want to rename the file to. Any part of the filename omitted will not be changed. The drive number can not be changed.
The filename used will become the new default filename!

FS:
This is the "file save" command. The format for the command is...

FSfilename <ENTER>

where "filename" is the name, extension, and drive number to be used in the save. If you omit any of these three items, the program will use the current default value. The filename used becomes the default filename!

If you omit the name, precede the extension with a "." (period). If you omit the extension, precede the drive number with a ":" (colon). If you use only "FS", the program will use the entire default filename.

This command is identical to the "CS" command listed above, except that this command will save all of the text in the buffer to disk.

GDIR:
This is the "get directory" command. The format is...

GDIRn <ENTER>

where "n" is the number of the drive to be used. If the number is omitted, the default drive number will be used. This command appends a directory listing to the end of your current text. The listing is in single column format.

NEW:
This command clears all text from the current text buffer. If you have combined buffers using the "BOTH" command, this command will again separate the buffers. Text can be recovered by using the "OOPS" command explained below.

NUMBER:
This command is used to setup the "next number out" feature. The format for the command is...

NUMBERn,i

where "n" will be the "next number out" when the TEXT mode command "<CTRL> N" is used, and the "i" is the increment that will be added to or subtracted from "n" each time the "<CTRL> N" command is used. If you key in...

NUMBER100,-10 <ENTER>

then the next number out, using the "<CTRL> N" command, would be 100. The next time the "<CTRL> N" command is used, the number output will be 90, then 80, etc...

The range for the number "n" is from 0 through 65,535. The range for the increment "i" is from -32,767 through 32,767

OOPS:
This command can be used to recover text erased by either the <ALT> <CTRL> C or the <ALT> <CTRL> Z commands.

TABOUT:
This command resets all of the current tab stops.

TL:
This is the "task load" command. Its format is...

TLn,filename <ENTER>

where "n" is a key from 0 through 9 that will be used later with the <CTRL> key to execute the task. Tasks must be executed while in the TEXT mode. If a filename is specified, then a "," (comma) must separate the number "n" from the "filename". The "filename" is the name, extension, and drive number of the task to be loaded. The file must have been saved using the "TS" command described below. Do not attempt to load a file that was not saved as a task! If you omit any of these three items, the program will use the current default value.

The filename used becomes the default filename! If you omit the name, precede the extension with a "." (period). If you omit the extension, precede the drive number with a ":" (colon). If you use only "TLn", the program will use the entire default filename. Look at this example...

TL3,atask.tsk:1 <ENTER>

Here, the task will be assigned to the "3" key. Later, you can execute the task by keying...

<CTRL> 3

The name of the text file containing the task is "atask". It was saved with the extension "tsk" to distinguish it as a task. The task is to be loaded from Drive #1.

In the following example...

TL0,:1 <ENTER>

the command will attempt to load a task using the default name and extension from Drive #1. It will assign the task to the "0" key.

TS:
This is the "task save" command and its format is...

TSn,filename <ENTER>

where "n" is a key from 0 through 9 that has the task assigned to it. The "filename" is the name, extension, and drive number to be used for the saved file. If any of these three parts are omitted, the current default value will be used.

The filename used will become the current default filename!

After entering the command, an error will result if no task exists for the number given. If a task does exist, it will be saved to disk as a task file.

WIDTH:
This is the "screen text width" command and its format is...

WIDTHn <ENTER>

where "n" is a number from 5 to 250 to use as a screen text width. If you do not include the number, the current screen width will be displayed.

The "screen text width" is used exclusively for displaying text on the screen and it's not used when text is printed on paper.

WINDOW:
This command will create a second word processor in a second screen "window" if the "window" does not already exist, (and if the "alternate buffer" is not in use) or it will eliminate the "window" if one does exist. You can immediately follow the word "WINDOW" with a number from 4 through 20 as the line number to initially place the horizontal bar.

This is not simply a "window" where you can view a different section of your text as with some word processors. This "window" is a complete new word processor that only shares certain things with the original word processor that will make the two of them more powerful than ever. For example, they both share the "locate" and "change" feature. In this way, if you are locating a specific string of text in one window, you can go to the second window and continue the search without having to key in the locate and change strings again.

In the CONFIG program (See section 1.5 "CONFIGURING YOUR SYSTEM"), the "Pri-buffer" number is the size of the original word processor's text buffer, and the "Alt-buffer" number is the size of the second word processor's text buffer.

One major way that the two word processors are related is through the TEXT mode command "<CTRL> G" which is used to copy text from one word processor to the other. (See <CTRL> G.)

Now that you realize that this second "window" is a complete word processor, I will also refer to it as the "alternate window" and to the first "window" as the "primary window". When you enter the "WINDOW" command, (provided the "alternate buffer" is available) the lower section of the screen will have a horizontal "bar" across it. The "bar" will be marked by "-" (dashes) and numbers placed 10 spaces apart. The numbers are for measuring only, and do not change if the screen scrolls to the left or to the right.

The top line of the screen will still be the "command line" for both windows, and block selection will still be done at the bottom of the screen for both windows.

You can switch back and forth between windows instantly by keying <CLEAR>. When you key <CLEAR>, you will switch windows and the cursor position will be where it was when you last left this window. The command line will read "PriCmd:" if you are in the "primary" window, and "AltCmd:" if you are in the "alternate" window.

The alternate window can be used to work on a separate file, or you can copy your file from the primary window to the alternate window (using the TEXT mode "<CTRL> G" command) so that you can view one section of your text while working on a different section.

Automatic print spooling (printing one file while working on a second file) is accomplished by working on text in one window, while the second window (word processor) is printing its file. To do this, initiate printing in one window using one of the print commands. When printing begins, use the <CLEAR> key to move to the opposite window. Printing will continue while you work on the text in this window.

Only one window can be involved in a print operation at a time. If you are in the window that is printing, only two keys will function before printing ends. First is the <CLEAR> key that will move you from window to window, and the second is the <SHIFT>-<BREAK> that will abort any printing regardless of which window you are in.

While you are in the window where the printing is taking place, the printer will be given a higher priority than the keyboard. Once you move to the opposite window, the keyboard will be given a higher priority (you will probably note a slow down in the printer) and you will be able to key in text as if there were no printing happening.
You can change the screen size of each window by moving the horizontal "bar" up or down. You can move the "bar" down by keying...

<CTRL> <SHIFT>-+

and you can move it up by keying...

<CTRL> -

Note: The "bar" can not be moved if one of the windows is sending text to the printer.

Each window has its own current filename. If you load text into one window, and then load a second file into the opposite window, each window will maintain its own filename.

When you no longer want the "alternate window", you can use the "WINDOW" command from either window to eliminate it. If you eliminate the "alternate window", all of the text in that window will be erased! Therefore, save any text that you want to keep before eliminating the window.

WRAP:
This command will toggle the "word wrap" on or off. When you enter this command, the top line will display either "WRAP ON" or "WRAP OFF" to let you know what its status is. If word wrap is off, then all words that go beyond the screen width setting will be broken at that point and the broken off section of the word will be carried down to the next line. If word wrap is on, words that go beyond the screen width setting will not be broken. Instead, the entire word will be moved (wrapped) to the beginning of the next line down. The same will apply for text output to the printer. Each line will end where the maximum number of characters will fit within the right margin.

40:
This command changes the screen to the 40 column screen.

80:
This command changes the screen to the 80 column screen.

/:
This is the "print parameter" command and it is used to change or view the print parameters. If you only use the "/" followed by <ENTER>, the screen will clear and display all of the print parameters. To change one or more of these print parameters, follow the "/" with the two letter abreviation for the parameter (See the table in section 1.15) and its new value. Place commas between parameters when more than one are being changed. Look at the example shown here...

/*,tm0,bm54 <ENTER>

In this example, the "*" (asterisk) is used to reset all parameters to their initial value. Then the "tm0" changes the top margin to 0. This is followed by a "," (comma) and then the "bm54" changes the bottom margin to 54. Each parameter and its range is covered in detail in section "PRINT PARAMETERS". Remember that if the "index column number" parameter (IC) is not 0, then the "table of contents column number" parameter (TC) must be 0, and vice versa.

<SHIFT>-?:
This is the "COMMAND mode" help command. The first time that a help command is used after the program is booted, the program will go to the current default drive and look for a file named "HELPFILE.TXT". If the file exists, it will be loaded and kept in memory. The "COMMAND mode" help will be displayed in an overlay window in reverse color. More than one window of help will be available. To move to each successive window listing, press any key. After the last window of help has been displayed and the next key pressed, the window will be erased and any text overlayed by the window will again be present. These "instant" help windows contain only a limited amount of information about the commands. Their purpose is simply to remind you of what you should already be familiar with from having read this manual.

TEXT MODE <CTRL> COMMANDS

The commands in this section must by keyed while in the TEXT mode. After you execute most of these commands, the name of the command will be displayed on the screen's top line until the next key is pressed.

<CTRL> A:
This is the "assign a key a task" command. When keyed, the top of the screen will read...

Select task:

Select a key from 0 through 9 to be assigned the task. (You can have 10 tasks in all.) When you select a task, the screen will then read...

Keys:

If a task already exists for the key selected, the task will be displayed after the "Keys:" prompt. This serves to let you know that an active task exists there, and also enables you to edit the task. (See task editing below.)

Now key in the keys that are to become the task. Key them in exactly as if you were performing the task yourself. If the <ALT> key is to be part of the task, generate it by keying the <ALT> twice. If one of the arrow keys is to be part of the task, precede the arrow key with the <ALT> key. If part of the task is to leave the COMMAND mode and go to the TEXT mode, do NOT use the <BREAK> key. If you do, you will terminate the entry of the task. Instead, use the sequence...

<ALT> <D-ARW>.

While keying in a task, the <CTRL> key will appear as a reverse colored "C", the <F1> key appears as a reverse colored "1", the <CLEAR> key appears as a reverse colored "F", the <F2> key (tab) appears as a reverse colored "T", the <SHIFT>-<ALT> will appear as a reverse colored "Æ ", and the <SHIFT>-@ appears as a reverse colored "@".

As with any entry on the COMMAND line, you can use the right and left arrow to move within the line, use the <SHIFT>-<F2> to delete a character at the cursor, and new keys will be inserted at the cursor position. All characters up to the cursor position will be part of the task.

You can "nestle" one task inside another. That is, one task can call and execute a second task, that can call and execute another, etc.. A task can not call a task that is already running. For example, task #1 can not call task #2 if task #2 in turn calls task #1, since task #1 would already be running.

Tasks can perform any set of keys that you would normally key in directly from the keyboard. They can load files, output text to the printer, or simply type the same thing over and over.

After you have your task keyed in, press the <D-ARW>. The screen will then ask...

Number:

Key in the number of times that the task is to be performed, followed by <ENTER>. The task has now been created and can be executed by keying the <CTRL> key followed by the key assigned to the task (0 through 9).

Tasks can only be executed while in the TEXT mode.

You can edit a task by following the same steps as when you create a task, except when the existing task (the one to be edited) is displayed after the "Keys:" prompt, you can use the right and left arrows to move about in the task, any new keys will be inserted to the left of the cursor position, and the <SHIFT>-<F2> can be used to delete the character at the cursor. When you exit, using the <D-ARW>, all characters up to the cursor position will be part of the task.

<ALT> <CTRL> A:
This is the "erase to beginning" command. It erases all text in the buffer from the cursor position to the text beginning.

<CTRL> B:
This is the "block print" command. When you key this, the program will search from the beginning of your text for a marked block. When it finds a block, it will display the first line of the block at the bottom of the screen and then it will check to see if it is the "Correct block?". If you respond with "n" for "no", it will continue on to the next block. When you respond with "y" for "yes", the program will output the block of text to the printer.

<ALT> <CTRL> B:
This is the "print block to memory" command. Its executes identical to the <CTRL> B command above, however, this command appends a copy of the printed block to your existing text in memory and does not print to the printer.

<CTRL> C:
This is the "continue" command. After you have performed a "find", (see <CTRL> F below) you can "continue" to search for the next occurrence of the text found by using this command. If you are doing both a "find" and a "replace", (see <CTRL> F below) then the "continue" will continue with both the "find" and the "replace".

<ALT> <CTRL> C:
This is the "clear all text" command. It is used to erase all text in the text buffer.

<CTRL> D:
This is the "delete character" command, and it will delete the character at the cursor. After the character has been deleted, the line will close up to fill the hole left by the deletion. Characters deleted can be restored (provided you have not used the <U-ARW> or <D-ARW> or have not left the line) by keying the <BREAK> key as an "oops" key.

<CTRL> E :
This is the "block erase" command. When keyed, block selection is performed as with "<CTRL> B" above. After you have selected the block, if the "verify" mode is on, you will be asked "Are you sure?" before it erases the block of text. If the "verify" mode is off, text will be erased immediately. If the block was below the cursor, the text at and before the cursor will remain the same. If the block was above the cursor, the screen will be redrawn and the line that the cursor is in will become the top line of the screen. If the cursor was inside the block erased, the screen will be redrawn and the first line below the block will become the top line on the screen and the cursor will be at the beginning of this line.

<CTRL> F:
This is the "find" or "find and replace" command. When you key this command, the top line on the screen will read...

Find:

Key in the characters that you want to find. You can find any characters that exist in your text, including the "carriage return" character generated by the <ENTER> key. Key in the characters as you did when you originally keyed them into your text. You can use a "wild card" character, generated by keying the <ALT> <SHIFT>-?, as part of your locate string. All "wild cards" will match any character that they are compaired to.

If the "mask" is on, (An "M" will be present in the upper right corner of the screen.) then all characters in the locate string will be "masked" so that they will match either upper or lower case letters in the text being searched. Thus, "String" would match "sTriNg" if the "mask" was on. If it is off, then each character must match exactly. You toggle the "mask" on or off by using the "<CTRL> <SHIFT>-=" while in the TEXT mode. If you do not specify a print font and/or underlining for the first character of the "find" string, then the string will match characters in any print font and/or underlining. You can specify a print font and/or underlining for the first character of the "find" string by keying the <F1> key followed by the key for the print font and/or underlining. You can combine these to search for a string that begins in both a print font and underlining. This is done by first keying <F1> and then the number for the print font, followed by <F1> and "u" for underlining. If you specify a print font or underlining, then the print fonts and underlining must match exactly.

After you have keyed in the "find" string, you will then have the option of performing a "find" by pressing the <D-ARW> key, or you can perform a "find and replace" by pressing the <U-ARW> key.

If you press the <D-ARW> key, the program will search for the next occurrence of the "find" string beginning at the cursor position. If it finds a matching string, the program will move to that place in the text. The screen will be redrawn with the line containing the string as the top line of the text on the screen. The cursor will be immediately after the string found.

If you press the <U-ARW>, the top line will then read...
Repl:

Key in what you want to replace the "find" string to. You can use print fonts and underlining in the same way that you would while entering text in the TEXT mode. After you have keyed in this string of characters, you can press either the <D-ARW> key, in which case only one replacement will be made, or you can press the <U-ARW> and then key in the number of occurrences to replace and then press <ENTER>. If the "verify" mode is on, and if a string is found that matches the "find" string, the screen will be then be redrawn so that the string found will be at the top of the text screen and the program will ask "Are you sure?". If you do not want this occurrence to be replaced, press "n" for "no". If you do want it to be replaced, press "y" for "yes".

If you terminated the "repl" string by keying the <D-ARW>, (for a single replacement) the command will now be finished. You can, however, use the <CTRL> C command to continue on to the next occurrence. If you terminated the "repl" string with the <U-ARW> key, (for "multi-locate and change"), the program will continue on until it has either replaced the number of strings requested, or until it has found the last occurance. When doing "multi-find and replaces", the command will report the total number replaced when it ends.

<ALT> <CTRL> F:
This is the "redo print font" command. You can use this command to "redo" (redefine) any of the default print fonts that are generated using the <F1> 1 through <F1> 5 key sequence.

When you key the command, the top line will read...
Redo (1-5):

Key in the number corresponding to the print font key that you want to change. The screen will then read...
Desc:

Key in a brief description (up to 15 characters) of the print font that is to be assigned to the print font key. I recommend that you end the description with a "-" (dash) and then the number corresponding to the key that you are redefining. In this way, when you move the cursor into a section of text that is of this print font, the top line of the screen will display both the description of the print font, and the key that must be pressed after the <F1> key to create this print font. When you have the description typed in, press <ENTER>. The top line will then read...
On :

Now enter the codes (numbers) necessary to turn the new print font on. Numbers must be separated by commas, and precede hexidecimal numbers with a "**$**" (dollar sign). Terminate the line by keying <ENTER>. Look at the example below...
On : 27,31 <ENTER>
Here, the numbers 27 and 31 must be sent to the printer to turn on the print font.

When you press <ENTER>, the screen will ask...
Off:

Key in the codes (numbers) required to change this print font back to the normal print font. When you have done this, the key will then be redefined as the new print font. All of the places in you text that use this particular font will reflect the new change.

<CTRL> G:
This is the "get text" command. It is used to "get" text from one "window" and move it to the other "window". (See the "WINDOW" command above for information on windows.) If you want to copy a section of text from one window to the other window, first mark off the text to be copied as a block. Then go to the opposite window (using <CLEAR>) and place the cursor where you want the text to be copied to. Then key...

<CTRL>G

Select the block as was done in "<CTRL> B" above. The text will then be transferred from one text file to the other text file. You can create a second copy of you file in the opposite window by marking your entire file as a block, and then using this command to copy the file.

<ALT> <CTRL> G:
This is the "get number" command. When keyed, the program will retrieve the last number stored (if one exists) while using the "Cray-O-Lator", and keys it in at the cursor position.

<CTRL> I:
This is the "insert" toggle command. When keyed, you will change from either the "insert" mode to the "overstrike" mode and the "I" in the upper right corner of the screen will become an "O", or you will change from the "overstrike" mode to the "insert" mode and the "O" in the upper right corner of the screen will change to an "I". When in the "insert" mode, all text keyed will be inserted into the text at the left of the cursor. While in the "overstrike" mode, all text keyed will replace the text at the cursor.

<ALT> <CTRL> I:
This is the "invert case" command. When keyed, inverting characters to all lower case or to all upper case becomes active. If you are in the "upper case only" mode, (a "U" is in the upper right corner of the screen) characters will be changed to upper case. If you are in the "upper and lower case" mode, (an "L" is in the upper right corner of the screen) all text will be changed to lower case. Use the right or left arrow to move the cursor within the line. As you do, the characters under the cursor are changed. If you come to a character that you don't want to change, key in the character as you want it, and continue on.

<CTRL> J:
This command directly sends a carriage return to the printer. It is used to "jump" the page up one line.

<ALT> <CTRL> J:
This command directly sends a form feed to the printer. It is used to "jump" the page up one page.

<CTRL> K:
This is the "block kopy" command. When you key this, the program will search from the beginning of your text for a marked block. Each time it finds a block, it will display the first line of the block at the bottom of the screen. It then checks to see if this is the "Correct block?". If you respond with "n" for "no", it will continue on to the next block. When you respond with "y" for "yes", the program will copy the marked block to the cursor position. If the "verify" mode is on, the program will first ask "Are you sure?" before it copies the block.

This command leaves the original text that was copied exactly as it was, and an exact copy (including print fonts and underlining) of the text in the block will be copied to the cursor position.

<CTRL> L:
This is the "move left one word" command. When keyed, the cursor will move to the beginning of the preceding word in the text.

<ALT> <CTRL> L:
This is the "delete left word" command. When keyed, it will delete the word to the left of the cursor (if one exists on the screen line). If the cursor is inside a word, then the first time the command is keyed, it will delete only the left part of the word that it is in. A "space" at the end of a word is considered part of that word. The deleted text can be recovered by using the <BREAK> key as an "oops" key, provided you have not keyed the <U-ARW> or the <D-ARW> or have not left the line.

<CTRL> M:
This is the "block move" command. When you key this, the program will search from the beginning of your text for a marked block. Each time it finds a block, it will display the first line of the block at the bottom of the screen. It then checks to see if this is the "Correct block?". If you respond with "n" for "no", it will continue on to the next block. If you respond with "y" for "yes", it will move the marked block to the cursor position. If the "verify" mode is on, the program will ask "Are you sure?" before moving the block.

If the cursor is inside the block that is to be moved, an error condition will occur. You can not move a block into itself.

<ALT> <CTRL> M:
This is the "mail-merge" command. A "mail-merge" is where text items from a list are merged into a specified file at specified places, and then the text is sent to a printer. The list consists of groups of items, and each group contains the same number of items. Starting at the beginning of each group in the list, each item is numbered from 1 to the number of items in the group.

With SIMPLY BETTER, the beginning of a list is marked with a "list" marker. The "list" marker is generated by keying...

<F1> L

When you key this, a reverse colored "!" will be output to the cursor position. After this marker, type in the number for how many items there are per group in the list

and then press <ENTER>. Each item in a list consists of a single line terminated by <ENTER>. For example, if you have a list made up of names and addresses, it might look as shown here...

Smith
Robert
1243 South Street
Sumtown
OR
97000
Jones
John
P.O. Box 555
Yurtown
OR
97001

This list is made up of groups of names and addresses, and each group consists of 6 lines. Line 1 of each group is the individual's last name. Line 2 is the first name, etc., and the last item in each group is item number 6 (the zip code). Each group contains 6 items. Therefore, at beginning of this list would be a separate line, keyed in as...

<F1> L6 <ENTER>

After you have created a list, save it as a separate file.

Once you have the list saved to disk, clear all text from memory and key in the text that will later have items from the list merged into it. To do this, key in your text as usual. Each time you come to a place where you want an item from the list, generate a "merge item" marker by keying...

<F1> M

This will place a reverse colored "m" on the screen. After this, key in the number of the item that you want to have merged into your text. In other words if you wanted the persons first name, and if the first name was the second item in each group in the list, you would key in the number 2. After this number, generate a second "merge item" marker by keying...

<F1> M

Later, each copy of the text printed will have the 2nd item from each group merged into the text at the proper place.

When you have the text completed, save it to disk.

To perform a "mail-merge", the text must be in memory, followed immediately by the list. Then, place the cursor where printing is to begin, and key...

<ALT> <CTRL> M

A copy of the text, from the cursor to the "list" marker, will be printed for each group of items in the list.

When doing multiple copies, you may want to end your text with an "end of page" marker (use the "<F1> <D-ARW>" sequence) so that your paper will advance to the next page after each copy of the text is printed.

<CTRL> N:
This is the "next number out" command. When keyed, the program will output a number to the cursor position as specified by the "NUMBER" command. (See the "NUMBER" command above.) The program will then increment or decrement this number according to the increment value setup by the "NUMBER" command, and that number will be output the next time this command is used.

The initial values for this command can be set using the CONFIG program as explained in section "CONFIGURING YOUR SYSTEM". If the initial "next number out" was 100 and the initial increment was -10, then this command, if executed in succession, would output 100, followed by 90, then 80, etc..

<CTRL> O:
This is the "open line for text" command. When you execute it, all of the text from the cursor position will drop down a line so that a hole will be opened for new text. This allows fast entry of text at the beginning of a long file.

<ALT> <CTRL> O:
This is the "Cray-O-Lator" command. When keyed, the program enters a four function pop-up calculator. The top of the screen displays the words "Cray-O-Lator", followed by a reverse colored area for containing the number "window", followed by a three digit abreviation for the base you are in, and then an "M" if a number has been stored.

You may use the "+", "-", "*", and "/" for the usual operation to be performed. When you key one of these, its symbol is placed on the top line, to the right of the base and memory area.

Use the <ALT> key to toggle numbers from plus to minus.

Key in numbers as you would normally with a calculator. You can use the "%" key to change to binary (base 2) and the 3 letter abreviation "Bin" will be placed to the right of the number. If you press the "$", you will change to hexidecimal (base 16) and "Hex" will be dispayed. If you press "T", you once again are in decimal (base 10), and the "Dec" is displayed. You can change bases at any time.

If you press "S", the number in the display will be stored and the letter "M" will appear on the top line. You can use "G" to "get" the last number stored. After you return to the TEXT mode, use <ALT> <CTRL> G to "get" the number stored into your text at the cursor position.

<CTRL> P:
This is the "print text" command. When keyed, your text will be output to the printer, starting with the character at the cursor position.

If you are not sure of how your text will be printed, use the "<CTRL> V" command (described below) to enter the VIEW mode. In this mode, your text will be shown on the screen as close as possible to how it will be printed. (See section "THE VIEW MODE".)

<ALT> <CTRL> P:
This is the "print to memory" command. When executed, text is printed as with the <CTRL> P command above, however, this command appends a copy of the printed text to the end of your current text in memory.

<CTRL> R:
This is the "move right one word" command. When keyed, the cursor will move to the beginning of the next word in the your text.

<ALT> <CTRL> R:
This is the "delete right word" command and it works identical to the "delete left word" (<ALT> <CTRL> L) command above except that the word to the right of the cursor will be deleted.

<CTRL> S:
This is the "sort text" command. When keyed, the program will sort sections of text beginning at the cursor position.

<ALT> <CTRL> S:
This is the "size" command. When keyed, the top line of the screen will display how much buffer has been used, how much buffer is left, and how many granules it will take to store the text on a disk.

<CTRL> T:
This is the "tab set/reset" command. When keyed, the program toggles the status of the tab stop at the cursor position.

<CTRL> U:
This is the "unmark sort markers" command. When keyed, all "sort" markers and "sort end" markers will be removed from the text beginning at the cursor position.

<CTRL> V:
This command puts you in the VIEW mode. Text in this mode will be displayed as close as possible to how it will look when printed on paper.

<ALT> <CTRL> V:
This command is used to turn the "verify" mode on or off. If the "verify" mode is on, a "V" will be present in the upper right corner of the screen. When the verify is on, many of the commands will ask "Are you sure?" before they execute.

<CTRL> W:
This is the "word count" command. Use it to display the total number of words in your text.

<CTRL> X:
This is the "xout" (delete) line command. All text in the line with the cursor will be erased. If you have not keyed the <U-ARW> or the <D-ARW> or have not left the line, the text can be recovered using the <BREAK> key as an "oops" key.

<CTRL> Z:
This is the "zap block markers" command. It erases all block markers from the cursor to the end of your text. If done while the cursor is inside a marked block, the command will not remove the markers that turned on this block, but it will remove all others after the cursor. This results in the block containing the cursor to continue to the end of the text.

<ALT> <CTRL> Z:
Use this command to erase all text from the cursor to the end of your text.

<CTRL> 0 through <CTRL> 9:
These commands execute the task assigned to the corresponding 0 through 9 key pressed. If the key has not been assigned a task, then a task will not be performed. If the "verify" mode is on, the program will ask "Are you sure?" before performing the task. Keying the <BREAK> key while one or more tasks are running will cause an immediate abort of all tasks.

<CTRL> <SHIFT>-=:
This command is used to turn the "mask" on or off. If the "mask" is on, an "M" will be present in the upper right corner of the screen. During a "locate", (executed by the "<CTRL> L" command) upper and lower case letters will be "masked" so that letters will match regardless of whether they are in upper or lower case. Thus the string "maTcHes" would match the string "MatcHeS" during a "locate". When the "mask" is off, all characters must match exactly.

<CTRL> -:
This is the "bar up" command and it can be used to increase the size of the lower window by raising the "bar" that separates the upper window from the lower window. (See section 1.24 for more information.)

<CTRL> <SHIFT>-+:
This is the "bar down" command. It is used to increase the size of the upper window by lowering the "bar" that separates the upper window from the lower window.

<CTRL> <SHIFT>-<:
This is the "erase to line start" command. It will erase all text from the cursor to the line beginning. You can use the <BREAK> key as an "oops" key to restore the erased text if you have not used the <U-ARW> or the <D-ARW> or if you have not left the line.

<CTRL> <SHIFT>->:
This is the "erase to line end" command. It will erase all text starting at the cursor and continuing to the end of the screen line. You can use the <BREAK> key as an "oops" key to restore the erased text if you have not used the <U-ARW> or the <D-ARW> or if you have not left the line.

<CTRL> /:
This is the TEXT mode help command. (See <CTRL> <SHIFT>-? below.)

<CTRL> <SHIFT>-?:
This is the <CTRL> key command help.

The first time that a help command is used after the program is booted, the program will go to the current default drive and look for a file named "HELPFILE.TXT". If the file exists, it will be loaded and then it will not need to be loaded again until the next time the program is booted. The listing for the specific help command will be shown in an overlay window in reverse color. More than one window of help will be available. To move to each successive window listing, press any key. After the last window of help has been shown and the next key pressed, the window will be erased and any overlayed text will still be present.

These "instant" help windows contain only a limited amount of information about the commands. Their purpose is simply to remind you of what you should already be familiar with from having read this manual.

TEXT MODE <F1> COMMANDS

These commands are all used to insert special codes (markers) into your text. Most generate visible markers that are shown in reverse color on the screen. The markers for underlining are hidden inside the text, as are markers for the 5 pre-defined print fonts.

<F1> B:
This is the "block" command, used to mark the start or end of a block. Keying this while in overstrike mode will toggle the "block" condition on or off. If toggled on, all text keyed in will be marked as part of a block. You can then key it again to turn off (end) the block. While in insert mode, if you toggle it on, all text from the cursor to the next block (or to the end of text) will be marked as a block. You can then move the cursor to where you want this block to end and key the command again, this time to toggle the block from on to off.

<F1> C:
This command will output a "force center" marker. It will appear as a reverse colored "c" on the screen. It is used to force a single line to be center justified, regardless of what the default justification (JU) is at that time. After the line is printed, the justification ill return to the default justification parameter (JU).

<F1> E:
This command will output a "sort end" marker, which appears as a reverse colored "e" on the screen. It is used to mark the end of the last item in a group of items to be sorted. When you are sorting sections of text, the beginning of each section to be sorted must be marked with a "sort item" marker (See "<F1> S" below).

<F1> F:
This command will output a "define footer" marker. It will appear as a reverse colored "f" on the screen. It is used to define the "footer". It must be the first character on a printed line. After this marker will be the text that is to be used as a "footer". This text is then terminated by a carriage return character. The marker, text, and carriage return are not part of the normal text. The program will not add a left or right margin to the line when it is printed. Therefore, you must add the left margin spaces to the beginning of the line. Make sure the line will fit on the printed line. You can use "insert page number" markers inside the "footer" to place page numbers on each page.

<F1> H:
This command will output a "define header" marker, which appears as a reverse colored "h" on the screen. It is used to define the "header". It must be the first character on a printed line. After this marker will be the text that is to be used as a "header". This text is then terminated by a carriage return character. The marker, text, and carriage return are not part of the normal text. The program will not add a left or right margin to the line when it is printed. Therefore, you must add the left margin spaces to the beginning of the line. Make sure the line will fit on the printed line. You can use "insert page number" markers inside the "header" to place page numbers on each page.

<F1> I:
This command will output an "index item" marker. It will appear as a reverse colored "i" on the screen. It is used to mark the beginning of an item that is to be placed in the index. The item is then terminated with a carriage return character. The marker, item, and carriage return are not part of the regular text. If you place these items prior to where the actual item is in your text, it is possible that the page number will not have yet been advanced to the page number of the actual item. To prevent this from happening, always place the marker, item, and carriage return in your text immediately after the real item. (See section "CREATING AN INDEX" for more information.)

<F1> L:
This command will output a "list" marker. It will appear as a reverse colored "l" on the screen. It is used to mark the beginning of a "list" that will be used to perform a mail-merge. After the marker, key in the number of items that there are in each group in the list and then end the line with <ENTER>. (See section 1.29 "PERFORMING MAIL-MERGES").

<F1> M:
This command will output a "merge item" marker. It will appear as a reverse colored "m" on the screen. It is used to mark a place in your text where you want to have an item from a mail-merge list inserted. The "merge item" marker is immediately followed by the item number, and then another "merge item" marker is used to terminate the number. When the mail-merge is performed, the "merge item" markers and numbers are replaced by the corresponding item from the list. (See section "PERFORMING MAIL-MERGES" for more information.)

<F1> N:
This command will output a "no-print" marker. It will appear as a reverse colored "n" on the screen. It is used to mark a section of text that is not to be printed. All text that exists between two of these markers will be ignored by the print routine, just as if the text did not exist. You can use the "no-print prints" parameter (NP) to print out these areas by setting the parameter to "y" for "yes".

<F1> P:
This command will output a "pause print" marker. It will appear as a reverse colored "p" on the screen. When the print routine comes to one of these markers, it will halt the printing until you press a key on the keyboard. (Use the "pagination" parameter (PA) to halt printing at the end of each page.)

<F1> S:
This command will output a "sort item" marker. It will appear as a reverse colored "s" on the screen. It is used to mark the beginning of a section of text that is to be sorted. After placing "sort item" markers at the beginning of each item, (They not only mark the beginning of an item, but also mark the end of the previous one.) place a "sort end" marker at the end of the last item to be sorted. (See section "SORTING SECTIONS OF TEXT" for more information.)

<F1> T:
This command will output a "table of contents item" marker. It will appear as a reverse colored "t" on the screen. It is used to mark the beginning of an item that is to be placed in the table of contents. Follow the marker with the text for the table of contents item, and then terminate it with <ENTER>. The marker, text, and carriage return are not part of the normal text. (See section "CREATING A TABLE OF CONTENTS" for more information.)

<F1> U:
This is the "underline" command. It is used to toggle underlining on or off. If, while in the overstrike mode, you toggle underlining on, then all text typed in will be underlined. You can stop underlining by keying the command again. If, while in the insert mode, you toggle underlining on, all text from the cursor to the next section of text that is underlined (or until the end of text) will be underlined. You can then turn off the underlining by placing the cursor where you want the underlining to end, and then key the command again.

<F1> 0 through <F1> 5:
These commands are used to turn the corresponding print font on, where "<F1> 0" is always normal print font. If, while in the overstrike mode, you turn a print font on, all text keyed in will be in the new print font. You can change to a new print font by keying in a new <F1> 0 through <F1> 5 command. If, while in the insert mode, you turn a print font on, all text from the cursor to the next text that is in a different print font will become the new print font. You can easily edit your text print fonts by placing the cursor where you want a new font (while in the insert mode) and then key the desired command (<F1> 0 through <F1> 5).

<F1> <SPACE>:
This is the "hard space" command. It will output a reverse colored "-". It is used between two sections of text where a "space" would normally be used, except that the "hard space" prevents word the wrap from separating the two sections of text. An example of such a place might be between "Mr." and the persons name so that "Mr." will not be printed at the end of a line with rest of the person's name printed at the beginning of the next line.

<F1> <D-ARW>:
This is the "end of page" command. It will output a reverse colored "F" on the screen. It is used to force the end of a page to occur at the "F" character. It is NOT the same as a "form feed" character. If the program were to send a "form feed" character (a number 12) to the printer, the printer would be given the responsibility of advancing the paper to the next page. The problem with doing this is that the printer would not print any of the footers, auxiliary lines, etc., that should be printed below the last line of text on that page.

<F1> <SHIFT>-*:
This command will output a "comment" marker. It will appear as a reverse colored "*" on the screen. It is used to mark the beginning of a comment. The comment is

then terminated by a carriage return character. The marker, comment, and carriage return are not part of the normal text, and will not be sent to the printer. The marker can occur inside a line of text, but it will end at the next carriage return. Thus, if you keyed in...

The printed <F1> * My comment <ENTER>
line of text. <ENTER>

the program would print...

The printed line of text.

Comments can be printed by setting on the "print comments" parameter (PC) to "y" for "yes".

<F1> <SHIFT>-<:

This command outputs a "force left" marker. It will appear as a reverse colored "<". It forces the justification of the line that it is in to be "left" justified. After the line is printed, the normal justification (determined by the "JU" parameter) will resume.

<F1> <SHIFT>->:

This command outputs a "force right" marker. It will appear as a reverse colored ">". It forces the justification of the line that it is in to be "right" justified. After the line is printed, the normal justification (determined by the "JU" parameter) will resume.

<F1> ,:

This command will output a "define auxiliary line A" marker. It will appear as a reverse colored "a" on the screen. It is used to define "auxiliary line A". It must be the first character on a printed line. After the marker will be the text that is to be the auxiliary line. It is then terminated by a carriage return character. The marker, text, and carriage return are not part of the normal text. The program will not add left or right margins to the line when it is printed, so you must add the left margin spaces to the beginning of the line. Make sure the line will fit on the printed line. You can use "insert page number" markers (See <F1> <SHIFT>-# below.) inside an auxiliary line to place page numbers on each page.

<F1> .:

This command will output a "define auxiliary line B" marker. It will appear as a reverse colored "b" on the screen. It is used to define "auxiliary line B". It must be the first character on a printed line. After the marker will be the text that is to be the auxiliary line. It is then terminated by a carriage return character. The marker, text, and carriage return are not part of the normal text. The program will not add left or right margins to the line when it is printed, so you must add the left margin spaces to the beginning of the line. Make sure the line will fit on the printed line. You can use "insert page number" markers (See <F1> <SHIFT>-# below.) inside an auxiliary line to place page numbers on each page.

<F1> <SHIFT>-#:

This command will output an "insert page number" marker. It will appear as a reverse colored "#" on the screen. When the print routine encounters this marker, it will replace the marker with the current page number. This allows you to use the current page number anywhere inside your text.

<F1> <SHIFT>-(:

This is the "begin print codes" command. It will output a reverse colored "(" on the screen. It is used to mark the beginning of one or more print codes that are to be sent to the printer. The markers and codes will not be printed on paper or on the screen while in the VIEW mode. Do not key in the numbers for the codes that you want sent to the printer! Instead, check Appendix A for the correct key or sequence of keys necessary to generate the desired codes. For example, if you want to send the printer a number 4, followed by a 17, you would key in...

<ALT> D <ALT> Q

Terminate the sequence with an "end print codes" marker described below.

<F1> <SHIFT>-):

This is the "end printer codes" command. It will output a reverse colored ")" on the screen. It is used to mark the end of one or more codes that are to be sent to a printer. (See "<F1> (" above for more information.)

<F1> 8:

This is the "begin graphics characters" command. It outputs a reverse colored "[". All characters after this have an offset of 128 added to the character before it is sent to the printer.

<F1> 9:

This is the "end graphics characters" command. It outputs a reverse colored "]" and ends outputing graphics characters started by <F1> 8 above.

<F1> /:

This is the "print parameters" command. It will output a reverse colored "/" on the screen. It is used to mark the beginning of one or more print parameters that are to be changed. Each parameter consists of a two letter abreviation, (Use the COMMAND mode "/" command to view the parameters and their current values.) followed by the new value or setting. Separate each parameter from the next with a "," (comma). The one exception is the "*" parameter. This parameter consists of only the one character and has no value. Its only purpose is to reset all print parameters to their initial status.

<F1> <SHIFT>-?:

This is the <F1> key help command. The first time that a help command is used after the program is booted, the program will go to the current default drive and look for a file named "HELPFILE.TXT". If the file exists, it will be loaded and then it will not need to be loaded again until the next time the program is booted. The help listing will be shown in an overlay window in reverse color. More than one window of help will be available. To move to each new window listing, press any key. After the last window of help has been shown and the next key pressed, the window will be erased and any text overlayed by the window will still be present. These "instant" help windows contain only a limited amount of information about the commands. Their purpose is simply to remind you of what you should already be familiar with from having read this manual.

PRINT PARAMETERS

All of the print parameters and their current settings can be viewed by using the COMMAND mode "/" command.

Placing a print parameter line at the very beginning of your text (use the "<F1> /" command) can have several advantages. First, the parameter settings will be saved with the text. When you load in the text file, the parameter line in the text saves you from having to go to the COMMAND mode to alter the settings. Second, it can save you from forgetting to set up a parameter, or from setting one to the wrong value. The first parameter in this line should be the "*" parameter. Doing this, each parameter is reset to its initial value, since one or more may have been changed when a previous text file was printed. After the "*" parameter, change any parameter that is not to be the default setting.

Each parameter, along with its range, and a brief description is listed here

AL:
This is the "Auxiliary Line A Line Number" parameter. Other than a "header" and a "footer" line, there are also two "auxiliary" lines available. One is "Auxiliary Line A", and the other is "Auxiliary Line B". The "AL" parameter is used to set the line number for "auxiliary line A". Its range is from 0 to the page length (PL) setting. If set to 0, the line will not be printed since a page never has a line number 0.

AS:
This is the "Auxiliary Line A Status" parameter. Its setting will determine which pages the auxiliary line A is to be printed on. You can select "A" for all pages, "E" for even numbered pages, or "O" for odd numbered pages.

BD:
This is the "Baud Rate" parameter. Use the table below to set this to the proper number for your printer.

0 = parallel output

1 = 300 baud 4 = 2400 baud
2 = 600 baud 5 = 4800 baud
3 = 1200 baud 6 = 9600 baud.

BL:
This is the "Auxiliary Line B Line Number" parameter. Other than a "header" and a "footer" line, there are also two "auxiliary" lines available. One is "Auxiliary Line A", and the other is "Auxiliary Line B". The "BL" parameter is used to set the line number for "auxiliary line B". Its range is from 0 to the page length (PL) setting. If set to 0, the line will not be printed since a page never has a line number 0.

BM:
This is the "Bottom Margin" parameter. Its range is from the top margin (TM) plus 1 to the page length (PL) setting.

BS:
This is the "Auxiliary Line B Status" parameter. Its setting will determine which pages the auxiliary line B is to be printed on. You can select "A" for all pages, "E" for even numbered pages, or "O" for odd numbered pages.

FL:
This is the "Footer Line Number" parameter. Set this parameter to the line number that the "footer" is to be printed on. Its range is from 0 to the page length (PL) setting. A value of 0 will turn off the printing of the line since no page has a line number 0.

FS:
This is the "Footer Status" parameter. Its setting will determine which pages the footer is to be printed on. You can select "A" for all pages, "E" for even numbered pages, or "O" for odd numbered pages.

HL:
This is the "Header Line Number" parameter. Set this parameter to the line number that the "header" is to be printed on. Its range is from 0 to the page length (PL) setting. A value of 0 will turn off the printing of the line since no page has a line number 0.

HS:
This is the "Header Status" parameter. Its setting will determine which pages the header line is to be printed on. You can select "A" for all pages, "E" for even numbered pages, or "O" for odd numbered pages.

IC:
This is the "Index Item Column Number" parameter. Its value is used as the column number for the page number for an index item when creating an index. Its range is from 0 to 230. (See section "CREATING AN INDEX" for more information.)

IP:
This is the "Invisible Print" parameter. When set to "y" for "yes", all text will be "invisibly" printed. That is, all text will still be processed by the print routine, (changing margins, etc..) however, no text will be output to the printer.

JN:
This is the "Justify Number" parameter. If the justify spaces parameter (JS) is set to "y" for "yes", then this parameter will determine the maximum number of spaces that can be inserted in between words when the justification parameter (JU) is set to "b" for both. If the print routine comes to a line where it needs to insert more spaces than allowed by this parameter, an error will occur and printing will be halted. It's best to use this while in the VIEW mode, and if the error occurs, you can then hyphenate the last word or modify the text so that an unsightly number of spaces will not be inserted into the line. Its range is from 0 to 10.

JS:
This is the "Justify Spaces" parameter. It can be set to either "y" for "yes" or "n" for "no". See "JN" above for information on its use.

JU:
This is the print "Justification" parameter. It can be set to "L" for left justification, "R" for right, "C" for center, or "B" for both left and right justification.

LF:
This is the "Line Feeds" parameter. If set to "y" for "yes", then a "line feed" (a number 10) will be sent to the printer along with each "carriage return" sent. Some printers require this "line feed" to advance the paper one line in addition to performing the carriage return. If you are unsure about your printer, set this parameter to "n" for "no" and print out a few lines of text. If each line is printed over the top of the preceding line, then you should set the parameter to "y" for "yes".

LM:
This is the "Left Margin" parameter. Its range is from 0 to the right margin (RM) setting less 1.

LS:
This is the "Line Spacing" parameter. If set to 1, then single line spacing will be used. If set to 2, double line spacing will be used, etc.. Its range is from 1 to 10.

NP:
This is the "No-Print Prints" parameter. If set to "y" for "yes", then all text in between "no-print" markers will also be printed.

NU:
The "Nulls" parameter. Increasing the value of this parameter will increase the length of time between when a carriage return is sent to the printer, and when the next character is be sent. Some printers require additional time for carriage returns.

PA:
This is the "Pagination" parameter. If set to "y" for "yes", printing will stop at the end of each printed page and will not resume until a key is pressed. Use this when you are hand feeding single sheets of paper into your printer.

PC:
This is the "Print Comments" parameter. If set to "y" for "yes", then text marked as comments will also be printed.

PF:
This is the "Print Fill Form" parameter. If set to "y" for "yes", then all characters not underlined will be changed to "spaces" before they are printed. If the character is underlined, it will be printed. All carriage returns are printed.

PH:
This is the "Printer Handshake" parameter. If set to "y" for "yes", the print routine will check the "busy" line coming from the printer to see if the printer is ready for the next character. If it shows "ready", the next character will be sent, otherwise printing will halt until the line changes to show "ready". If the printer fails to show a "ready" status, you can abort the printing by using <SHIFT>-<BREAK>.

PL:
This is the "Page Length" parameter. Its value is used to advance the paper to the next page properly. Its range is from the bottom margin (BM) setting to 255.

PN:
This is the "Page Number" parameter. It is the number of the current page being printed.

PP:
This is the "Page Print" parameter. Set it to "a" for "all" pages , "e" for "even" numbered pages, or "o" for "odd" numbered pages. This makes printing both sides of pages easier. First, print the odd (or "even") side, then print the back side with the setting opposite to its first setting.

RM:
The "Right Margin" parameter. Range is from the left margin (LM) setting plus 1 to 250.

TC:
The "Table of Contents Column Number" parameter. It is the column number for the page number for a table of contents item when creating a table of contents. Its range is 0 to 230.

TM:
The "Top Margin" parameter. Its range is from 0 to the bottom margin (BM) setting less 1.

Section 3

Condensed Command Reference

COMMAND MODE COMMANDS CONDENSED

AUTO:	Toggle the "auto save" on or off
BEEP:	Toggle key beeps, alert beeps, and key beeps on or off.
BOTH:	Combine the alternate and primary buffers together.
BS:	Save a block of text to disk.
CLEAN:	Remove all underline and print font codes from all of the text in the buffer.
CS:	Save all text after the cursor position to a disk.
DIR:	Display the disk directory.
FK:	Kill a text file on a disk.
FL:	Load a text file from a disk.
FN:	Display or change the current filename.
FR:	Rename a text file on a disk.
FS:	Save all text in the buffer to a disk.
BYE:	Exit SIMPLY BETTER to BASIC.
EYES:	Toggle hearing impaired mode on or off.
GDIR:	Get directory to memory.
NEW:	Clear the current text buffer of all text.
NUMBER:	View or change the next number out and its increment.
OOPS:	Recover text erased by the CLEAR or AFTER command.
TABOUT:	Erase all tab stop settings.
TL:	Load a task from a disk.
TS:	Save a task to a disk.
WIDTH:	View or change the screen text width.
WINDOW:	Create or end the alternate window.
WRAP:	Toggle word wrap on or off.
40:	Change to a 40 column screen display.
80:	Change to an 80 column screen display.
/:	View or change one or more print parameters.
<SHIFT>-?:	Display an overlay of COMMAND mode help

<CTRL> KEY COMMANDS CONDENSED

<CTRL> A: 'A'ssign a key a task.

<CTRL> B: 'B'lock print text.

<CTRL> C: 'C'ontinue with find/replace.

<CTRL> D: 'D'elete the character at the cursor.

<CTRL> E: 'E'rase block of text.

<CTRL> F: 'F'ind or find & replace text.

<CTRL> G: 'G'et text from the opposite window to the cursor position.

<CTRL> I: 'I'nsert mode toggle.

<CTRL> J: 'J'ump a line (CR to the printer).

<CTRL> K: 'K'opy block of text.

<CTRL> L: 'L'eft move a word.

<CTRL> M: 'M'ove a block of text.

<CTRL> N: 'N'ext number out to the cursor position.

<CTRL> O: 'O'pen text for inserting lines.

<CTRL> P: 'P'rint text to the printer.

<CTRL> R: 'R'ight move a word.

<CTRL> S: 'S'ort text beginning at the cursor position.

<CTRL> T: 'T'ab set/reset at the cursor position.

<CTRL> U: 'U'nmark sort markers.

<CTRL> V: 'V'iew text as it will be printed.

<CTRL> W: 'W'ord count display.

<CTRL> X: 'X'out a line (erase).

<CTRL> Z: 'Z'ap all block markers from the cursor position.

<CTRL> 0-9: Execute one of the tasks from 0 through 9.

<CTRL> <SHIFT>-=: Toggle the mask on or off.

<CTRL> -: Move the "bar" up one line.

<CTRL> <SHIFT>-+: Move the "bar" down one line.

<CTRL> <SHIFT>-<: Erase text to beginning of the line.

<CTRL> <SHIFT>->: Erase text to the end of the line.

<CTRL> /: Display an overlay of TEXT mode help listings.

<CTRL> <SHIFT>-?: Display an overlay of <CTRL> command

<CTRL> <CTRL>: Go to the COMMAND mode from the TEXT mode.

<ALT> <CTRL> A: Erase text to beginning ('A' to Z)

<ALT> <CTRL> B: Print block of text to memory.

<ALT> <CTRL> C: 'C'lear out all text in the buffer.

<ALT> <CTRL> F: 'F'ormat change 1-5.

<ALT> <CTRL> G: 'G'et number saved with Cray-O-Lator.

<ALT> <CTRL> I: 'I'nvert case of text characters.

<ALT> <CTRL> J: 'J'ump printer up a page (FF to printer).

<ALT> <CTRL> L: 'L'eft word erase.

<ALT> <CTRL> M: 'M'ail-merge initiate.

<ALT> <CTRL> O: Cray 'O' Lator (calculator)

<ALT> <CTRL> P: 'P'rint text to memory.

<ALT> <CTRL> R: 'R'ight word erase.

<ALT> <CTRL> S: 'S'ize of text display.

<ALT> <CTRL> V: 'V'erify mode toggle.

<ALT> <CTRL> Z: Erase to text end (A to 'Z').

<F1> KEY COMMANDS CONDENSED

<F1> B: Mark the beginning or end of a block of text.

<F1> C: Output a "force center justification" marker.

<F1> E: Output a "sort end" marker.

<F1> F: Output a "define footer" marker.

<F1> H: Output a "define header" marker.

<F1> I: Output an "index item" marker.

<F1> L: Output a "mail-merge list" marker.

<F1> M: Output a "mail-merge item" marker.

<F1> N: Output a "no-print" marker.

<F1> P: Output a "pause print" marker.

<F1> S: Output a "sort item" marker.

<F1> T: Output a "table of contents item" marker.

<F1> U: Mark the beginning or end of underlining.

<F1> 0-5: Turn on the corresponding 0-5 print font.

<F1> <SPACE>: Output a "hard space" character.

<F1> <D-ARW>: Output an "end of page" marker.

<F1> <SHIFT>-*: Output a "comment text" marker.

<F1> <SHIFT>-<: Output a "force left justification" marker.

<F1> <SHIFT>->: Output a "force right justification" marker.

<F1> ,: Output a "define auxiliary A" marker.

<F1> .: Output a "define auxiliary B" marker.

<F1> <SHIFT>-#: Output an "insert page number" marker.

<F1> <SHIFT>-(: Output a "begin printer codes" marker.

<F1> <SHIFT>-): Output an "end printer codes" marker.

<F1> 8: Output a "begin graphic characters" marker.

<F1> 9: Output an "end graphics characters" marker.

<F1> /: Output a "print parameter list" marker.

<F1> <SHIFT> ?: Display an overlay of <F1> key help listings.

APPENDIX A: STANDARD AND ALTERNATE KEYS"

To Generate Dec	Hex	Keys To Press
1	1	<ALT> A
2	2	<ALT> B
3	3	<ALT> C
4	4	<ALT> D
5	5	<ALT> E
6	6	<ALT> F
7	7	<ALT> G
8	8	<ALT> H
9	9	<ALT> I
10	A	<ALT> J
11	B	<ALT> K
12	C	<ALT> L
13	D	<ALT> M
14	E	<ALT> N
15	F	<ALT> O
16	10	<ALT> P
17	11	<ALT> Q
18	12	<ALT> R
19	13	<ALT> S
20	14	<ALT> T
21	15	<ALT> U
22	16	<ALT> V
23	17	<ALT> W
24	18	<ALT> X
25	19	<ALT> Y
26	1A	<ALT> Z
27	1B	<ALT> 1
28	1C	<ALT> 2
29	1D	<ALT> 3
30	1E	<ALT> 4
31	1F	<ALT> 5
32	20	<SPACE>
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/

To generate Dec	Hex	Keys To Press
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	<SHIFT>-<D-ARW>
92	5C	<SHIFT>-<CLEAR>
93	5D	<SHIFT>-<R-ARW>
94	5E	<ALT> <U-ARW>

To generate Dec	Hex	Keys To Press
95	5F	<SHIFT>-<U-ARW>
96	60	<ALT> 6
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o

To Generate Dec	Hex	Keys To Press
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	<ALT> 7
124	7C	<ALT> 8
125	7D	<ALT> 9
126	7E	<ALT> :
127	7F	<ALT> ;

APPENDIX B: SYSTEM ERRORS EXPLAINED

The following is an alphabetical list of all error messages that will be displayed on the screen's command line.

"Error: BM > PL":You have attempted to set the bottom margin (BM) to a value greater than the page length, or vice versa.

"Error: NAM": The filename specified is not correct.

"Error: bad marker": A print marker is incorrect.

"Error: bad parameter": One of the parameters specified was keyed in incorrectly.

"Error: block not found": The program can not find a block. Either no block exists, or no block was selected from those that do exist.

"Error: BSY": The disk controller returned a "busy" error.

"Error: called running task": One of your tasks called a task that was already running.

"Error: code too long": The print codes between a "print code" marker and a "print code end" marker are too long.

"Error: CRC": The disk controller returned a "crc" error.

"Error: cursor position": The cursor position is bad. Usually when you try to copy or move a block to a place inside the block, or to a place that is not part of your text.

"Error: DRQ": The disk controller returned a "data request" error.

"Error: different drives": This will occur if you try and rename a file to a drive number other than the drive number where the file exists.

"Error: EOL not found": This occurs if lines such as comment lines, footers, etc., are too long. (End Of Line not found.)

"Error: EOT not found": This occurs when the "OOPS" command can not find the "End Of Text" that existed before text was erased using the "CLEAR" or "AFTER" command.

"Error: help not found": The file "HELPFILE.TXT" could not be found on the disk.

"Error: IC/TC <> 0": The "index column number" parameter (IC) or the "table of contents column number" parameter (TC) must equal 0. (Both must be 0 during a "mail-merge".)

"Error: DRV": You have specified a drive number that is not allowed.

"Error: index too long": One of the text lines for the index is too long.

"Error: SEK": The disk controller returned a "seek" error.

"Error: WRT": The disk controller returned a "write" error

"Error: LM >= RM": You have attempted to set the left margin (LM) to a value greater than or equal to the right margin (RM), or vice versa.

"Error: LS <> 1-10": You have attempted to set line spacing to a value other than 1 through 10.

"Error: LOS": The disk controller returned a "lost data" error.

"Error: merge on": You have attempted to create an index or a table of contents, while performing a mail-merge.

"Error: NEF": The filename does not exist on the disk

"Error: not available": The command keyed is not available at the moment. This usually happens when printing restricts certain commands.

"Error: RDY": The disk controller returned a "not ready" error. Usually caused by not closing the disk drive.

"Error: out of memory": This will occur if a procedure can not be performed because of a lack of available memory.

"Error: out of range": One or more numbers are out of range.

"Error: PL < HL/FL/AL/BL": You have attempted to set the page length (PL) to a value less than the "header", "footer", or "auxiliary" line number, or vice versa.

"Error: printer busy": This occurs when you attempt to print two files at the same time.

"Error: print abort": A <SHIFT>-<BREAK> aborted the printing.

"Error: spaces > JN": The number of spaces to be inserted between two words, while printing with both left and right justification, have exceeded the number specified in the justify number parameter (JN).

"Error: TC <> 0": The table of contents (TC) value is not 0.

"Error: TM >= BM": You have attempted to set the top margin (TM) to a value greater than or equal to the bottom margin (BM) setting, or vice versa.

"Error: task buffer empty": The task that you are trying to save is empty.

"Error: task not 0-9": You have attempted to assign a task to a key that is not 0 through 9.

"Error: window not available": This error will occur if there is no "alternate" window for a command that requires the alternate window. (Such as the <CLEAR> key.)

"Error: WPT": The disk you attempted to write on is write protected.

"Exit here": This will occur if you press the <ENTER> key while in the VIEW mode. The error will cause you to exit the VIEW mode to the place being viewed.

"File too large!": This message is displayed if there is not enough room on a disk to hold the file.

"Text buffer full!": This message is displayed if the text buffer fills while keying in text.

AUTO 5, 49, 75
Auxiliary line
- markers 38, 98
- numbers 99

Bar, horizontal 27, 59, 81, 93
Baud rate 37, 42
BEEP 6, 11, 76
Blocks
- copying 28, 89
- displaying 27
- erasing 27, 86
- get from window 61, 88
- marking 28, 95
- max number 27
- moving 28, 89
- printing to memory 29, 85
- printing 29, 42, 85
- removing markers 27, 93
- saving 29, 49, 76
- selecting 28, 95, 106

BOTH 62, 76
BS 30, 76
Buffers
- alternate 48, 50, 59, 62, 76, 78, 81
- current line 13,20,21
- filling 50
- merging 48, 50, 62, 76
- primary 5, 13, 17, 21, 60, 62, 76, 81
- size 48, 92
- type ahead 17

BYE 73, 76
Carriage return 13, 40
Centering lines 34, 37, 95
Changing current filename 47, 79
CLEAN 25, 77
CONFIG.BAS 3
Command mode 11, 15, 73, 75
Comments 35, 39, 97
Constant number 71
Continue locate 45, 85
Cray-O-Lator 70
CS 49, 77
Data modules 3, 8
Default filename 47
Delete
- all after 21, 93
- all before 21, 85
- all text 21, 80, 86
- character 20, 86
- left word 21, 89
- line 21, 93
- right word 21, 92
- to line start or end 21, 94

DIR 47, 77
Directory
- displaying 47
- retrieving 48, 80

Edit
- COMMAND line 75
- tasks 54

End of page markers 97
End print codes 99
End sort markers 37, 95
Epson graphics 39
Exit to BASIC 73, 76
EYES 6, 12, 77
Filename 2, 48
Files
- killing 51
- loading 50, 78
- renaming 51, 79
- saving 48, 79

Filling forms 36, 63, 103
Find & Replace text 43, 45, 86
FK 51, 78
FL 50, 78
FN 47, 79
Fonts
- changing 22, 44, 97
- codes 7, 22
- descriptors 7, 22, 87
- editing 22
- hidden codes 24, 78
- in auxiliary lines 31
- in headers, footers 31
- in sorting 58
- locating 44
- redefining 24, 87
- removing 25, 78

Footers 31, 95
Footer line number 31, 101
Footer marker 32, 95
Footer status 32, 101
Form feeds 36, 97
FR 51, 79
FS 49, 79
GDIR 48, 79
Go to printed page 41
Graphics characters 39, 99
Hard spaces 39, 97
Headers 32, 95
Header line number 32, 101
Header marker 32, 37, 95
Header status 32, 101
Hearing impaired 12, 77
Help
- command mode 83
- text mode 94
- <CTRL> keys 94
- <F1> keys 99

IBM graphics 39, 99
Indexes
- creating 67
- item markers 38, 67, 97
- column number 67, 101

Insert mode 6, 12, 20, 25, 58
Invisible printing 34, 66, 101
Jump paper up line/page 88
Justification
- left & right 34, 102
- center 34, 37, 95
- left or right 34, 39, 98
- spaces 34

Key beep 11, 76
Key repeat 15
Killing files 51, 78
Line feeds 35, 102
Line spacing 35, 102
Loading files 50, 78
Loading tasks 54, 80
Mail-merge
- item 38, 68, 96
- list 38, 68, 96
- description 68
- executing 69, 89
- text body 69

Margins 33, 101, 102, 103
Masks 6, 12, 43, 93
Monitors 3, 10
Move
- to line start or end 18
- to screen bottom 19
- to screen top 19
- one screen up 19
- one screen down 19
- to text start or end 19
- to page number 41
- one word left 18, 89
- one word right 18, 92

Multiple replacements 46
New page marker 40, 97
Next number out 55, 80, 91
No-print areas 35, 38, 96
NUMBER 5, 55, 80
Nulling text 46
Nulls, printer 8, 35, 102
Numbers
- bases 71, 72
- binary 16, 70, 71
- constant 71
- decimal 72
- for keys 16
- getting 72
- hexidecimal 7, 72
- negative 71
- page 36, 39, 103
- saving 70

OOPS 21, 80
Oops key 13, 17, 21
Open for text 20, 91
Overstrike mode 6, 12, 25, 88
Page length 36, 103
Page numbers 30, 36, 39, 98, 103

Parameters
- changing 31, 38, 83, 100
- default 31, 100
- viewing 30, 83
- line markers 38

Pre-printed forms 36, 63
Printer codes 39, 99
Printer handshake 36, 103
Printing comments 35, 103
Printing
- abort 29, 42
- all pages 37, 103
- blocks to memory 29, 85
- blocks 29, 85
- both sides 37, 103
- errors 62
- even pages, odd pages 37, 103
- from cursor 42, 91
- initiating 42, 91
- invisible 34, 66, 101
- priority 62, 82
- spooling 62, 82
- to memory 29, 42, 85, 91
- viewing 40, 91

Print filling forms 36, 63, 103
Print pause markers 38, 96
Removing sort markers 57, 92
Renaming files 51, 79
Repeat <CTRL> key 18
Saving
- all text 48, 79
- blocks 49, 76
- from cursor 49, 76
- tasks 53, 81

Screen column number 12, 19
Screen size 5, 11, 83
Scrolling 15
Shift lock 13
Single Page Output 37, 103
Sorting text 57, 92
Sort end markers 37, 57, 95
Sort item markers 38, 57, 96
Sort markers, removing 57, 92
Spaces, hard 39, 97
Spooling 62, 82
TABOUT 19, 80
Table of contents col. number 66, 103
Table of contents item 38, 65, 97
Table of contents 38, 65
Tab stops 19, 80, 92
Tab key 17, 19
Tasks
- aborting 53
- creating 52, 84
- editing 54
- executing 53, 92
- loading 54, 80
- saving 53, 80
- size 52

Text area size 48, 92
Text colors 5
Text mode 11, 12
Text size 48, 92
TL 54, 80
TS 53, 81
Underlining 17, 22, 24, 44, 56, 63, 97
Verify mode 6, 13, 28, 46, 92
Viewing current filename 47, 79
View mode 11, 40, 91
WIDTH 6, 11, 15, 81
WINDOW 59, 61, 81
Wild card characters 43
Windows 59
changing 60, 82
creating 59, 81
eliminating 60, 83
size of 60, 82, 92
Word count 48, 92
WRAP 4, 11, 83
40, 80 7,15,112
<ALT> arrow keys 52
<ALT> key 16, 41, 70
<ALT> <CTRL> A 21, 85
<ALT> <CTRL> B 29, 85
<ALT> <CTRL> C 14, 23, 85
<ALT> <CTRL> F 87
<ALT> <CTRL> G 72, 98
<ALT> <CTRL> I 88
<ALT> <CTRL> J 88
<ALT> <CTRL> L 18, 89
<ALT> <CTRL> M 69, 89
<ALT> <CTRL> O 70, 91
<ALT> <CTRL> P 42, 92
<ALT> <CTRL> R 18, 92
<ALT> <CTRL> S 60, 92
<ALT> <CTRL> V 13, 92
<ALT> <CTRL> Z 21, 93
<BREAK> key 9, 17, 70
<CLEAR> key 17, 59, 70, 82
<CTRL> key 16, 84
<CTRL> A 54, 84
<CTRL> B 29, 42, 85
<CTRL> C 45, 85
<CTRL> D 20, 85
<CTRL> E 28, 86
<CTRL> F 43, 86
<CTRL> G 61, 81, 88
<CTRL> I 12, 14, 88
<CTRL> J 88
<CTRL> K 28, 89
<CTRL> L 18, 89
<CTRL> M 28, 89
<CTRL> N 6, 55, 91
<CTRL> O 20, 91
<CTRL> P 42, 66, 91
<CTRL> R 18, 92
<CTRL> S 57, 92
<CTRL> T 19, 92
<CTRL> U 57, 92
<CTRL> V 40, 92
<CTRL> W 48, 92
<CTRL> X 21, 93
<CTRL> Z 27, 93
<CTRL> 0-9 54, 93
<CTRL> <D-ARW> 18
<CTRL> <L-ARW 18
<CTRL> <R-ARW> 19
<CTRL> <SHIFT>-<D-ARW> 19
<CTRL> <SHIFT>-<L-ARW> 19
<CTRL> <SHIFT>-<R-ARW> 19
<CTRL> <SHIFT>-<U-ARW> 19
<CTRL> <SHIFT>-< 21, 93
<CTRL> <SHIFT>-= 93
<CTRL> <SHIFT>-> 21, 94
<CTRL> <SHIFT>-? 73, 94
<CTRL> <SHIFT>-+ 60, 93
<CTRL> <CTRL> 75
<CTRL> <U-ARW> 19
<CTRL> - 60, 82, 93
<CTRL> / 73, 94
<F1> B 27, 95
<F1> C 34, 37, 95
<F1> E 37, 57, 95
<F1> F 32, 37, 95
<F1> H 32, 37, 95
<F1> I 38, 67, 96
<F1> key 17
<F1> L 38, 69, 96
<F1> M 38, 69, 96
<F1> N 35, 38, 96
<F1> P 38, 96
<F1> S 38, 57, 96
<F1> T 38, 66, 97
<F1> U 17, 22, 44, 97
<F1> 0-5 24, 97
<F1> 8 39, 99
<F1> 9 39, 99
<F1> <D-ARW> 40, 97
<F1> <SHIFT>-< 34, 39, 98
<F1> <SHIFT>-> 34, 39, 98
<F1> <SHIFT>-? 73, 99
<F1> <SHIFT>-# 33, 39, 98
<F1> <SHIFT>-(39, 99
<F1> <SHIFT>-) 39, 99
<F1> <SHIFT>-* 35, 39, 97
<F1> <SPACE> 39, 97
<F1> , 32, 38, 98
<F1> . 32, 38, 98
<F1> / 31, 38, 99
<F2> key 17, 19
<SHIFT> 2
<SHIFT>-<BREAK> 29, 42, 69
<SHIFT>-? 73
<SHIFT>-<ALT> 18, 19

Copyright Notice