Super "Color" Writer II

For the TRS-80 Color Computer and TDP System 100 Personal Computer.

VIP WRITER

Word Processing Tutorial

TABLE OF CONTENTS

SYSTEM REQUIREMENTS AND LOADING INSTRUCTIONS

SECTION 1 INTRODUCTION

SECTION 2 GETTING STARTED

SECTION 3 EDITING FUNCTIONS

SECTION 4 SAVING AND LOADING TEXT FILES

SECTION 5 ADVANCED FEATURES - TEXT FORMATTING

SECTION 6 WINDOW MODE

SECTION 7 PRINTING AND PAGE FORMATTING

SECTION 7 PRINTING AND PAGE FORMATTING (continued)

SECTION 8 PROGRAMMABLE FUNCTIONS

APPENDICES

SYSTEM REQUIREMENTS AND LOADING INSTRUCTIONS

SYSTEM REQUIREMENTS

The Super "Color" Writer II was designed:

* To work with all TRS-80C and TDP Computers
* For use with 16 to 64K WITHOUT REQUIRING FLEX (tm);
* For use with ANY printer;
* To be compatible with all Super "Color" Library programs;
* With MEMORY-SENSE to automatically sense the memory size of your machine and adapt to it.

There are the following limitations in use:

* It may not be used when the joysticks are plugged in.
* The lowercase display is not available to tape and disk users with less than 32K of memory.

LOADING INSTRUCTIONS

DISKETTE VERSION Mount the Super "Color" Writer II master diskette in Drive 0, type LOADM"WRITER" and press <ENTER>. Almost immediately a billboard will appear and the program will continue to load. After the program is loaded it will first perform a memory test, taking a few seconds, and then it will automatically execute. Press any key to enter the program in the COMMAND MODE. In this mode you may immediately access a disk file. After the program has executed, remove the Master Diskette and insert it in its jacket and refer to Section 1.

ROMPAK VERSION Be sure that the power is OFF. INSERTING THE ROMPAK WHILE THE POWER IS ON MAY DAMAGE BOTH THE COMPUTER AND ROMPAK AND WILL VOID YOUR WARRANTY FOR THE Super "Color" Writer II. Insert the ROMPAK and turn the power on. A billboard will appear. Press any key to enter the COMMAND mode and refer to Section 1, Introduction. (The disk commands in Sections 4-F to 4-O pertain ONLY to Disk users.)

TAPE VERSION Turn on the computer, insert the master tape in the cassette machine, set the volume control to the low-to-medium setting, then type "CLOADM" <ENTER>, and press "PLAY". A billboard will appear, and when the program has loaded, it will perform a memory test, and then automatically execute. After the program has executed press any key to enter the program in the COMMAND MODE from which you may access a tape file. Once the program has executed, rewind the cassette, remove it and put it in its case. Refer to Section 1, Introduction. If an I/O error occurs during loading, use the second copy of the program which follows the first on the cassette. (The Disk commands in Sections 4-F thru 4-O pertain ONLY to Disk users.)

SUPER "COLOR" WRITER II VERSION 3.0 ADDENDUM

R E A D T H I S F I R S T!

The following changes must be included in the Tutorial:

Page 9 To the first full paragraph add: "WIDTH may ONLY be changed when the current line you are on has been terminated with a carriage return."

After the first full paragraph add:

If you decide, for easy editing, to set your WIDTH so that it will be the same as your screen display line length, it is advisable to set it to one character LESS than the actual line length (e.g., 31, 50, 63 or 84). This is because of a "soft" space used at the end of the line of text for wrap around purposes. If you fail to do this, the screen will always window over to the right when you reach the end of the line before your text wraps around. This could be disconcerting to watch, although it is perfectly normal. Thus, to avoid this screen display characteristic, set the WIDTH one character less than the screen display length.

Page 10 At the end of the last paragraph add: "Whenever you have changed text by typing over other text or inserting it, the next time you press the <UP ARROW> or <DOWN ARROW> key to move the cursor up or down on the screen, the cursor will not move but the keystroke will be registered. This keystroke acts to "set" the text in memory. Subsequent pressing of the arrow keys will, however, result in cursor movement."

Page 15 To the end of the last paragraph in Section 2-H add: "The ROMPAK version does not, however, support the exit to BASIC command, so the END command may not be used with that version."

Page 16 To Section 2-F add the following:

NOTE* The <CLEAR>-<Q> command may not be used to go left accross EDIT WINDOW boundaries.

Page 18 UNDO COMMAND. See addition to page 10 above.

Page 28 To the end of the Paragraph in Section 4-B add: "If you wish to break out of a tape load, press <BREAK>. If no file or no leader has been found by the system when loading, the "S" for "searching" will remain on the screen until the system is reset. Press the reset button and the system will return with the cursor on the COMMAND line."

Pages 52, 53 & 86 The Line Delay feature performs the same function as Nulls. The Line Delay feature has thus been eliminated as redundant. If your printer requires Line Delays, you may instead send it the same number of Nulls.

Page 62 To the last full paragraph, add: "The key sequences must never be separated by spaces."

Page 67 The first two examples should read:

EXAMPLE 1: k1=27;31,2=27;32,3=15,4=14,5=27;30,6=27;28,
7=27;20,8=27;19,9=27;14,:=27;15<ENTER>
EXAMPLE 2: k1=$1B;$1F,2=$1B;$20,3=$0F,4=$0E,5=$1B;$1E,
6=$1B;$1C,7=$1B;$14,8=$1B;$13,9=$1B;$0E,
:=$1E;$0F<ENTER>

NOTE: The numeric equates for each key must be separated by semi-colons; the separate key equations must be separated by commas, and MAY NEVER be separated by spaces.

Page 88 The following changes must be made to the chart:

1) Nulls have no representation if generated in either the 32 or the hi-res displays.

2) The representation of CTRL L in the hi-res displays is a double down arrow, one on top of the other.

SECTION 1 INTRODUCTION

1-A LEARNING WORD PROCESSING

If you have a need to write, occassionally or professionally, the Super "Color" Writer II will save you a lot of time, frustration, ink and paper, while giving you a beautiful final printed product. And you can save your textfile and use it over and over again, with or without alteration. Your teachers will be amazed and impressed, your boss will appreciate your economy and good taste, and you will reap unending satisfaction. All this because the Super "Color" Writer II makes one of your most fundamental tasks, communication, infinitely easier.

What exactly is "word processing"? What will it do for you? Word processing is a blend of typing, correcting errors, inserting, deleting and otherwise manipulating text. At least that is what word processing was until just recently with the old "dumb" printers. Now smart printers have arrived. These incredibly versatile machines have allowed word processing to assume still another facet - typesetting. With a smart printer, nearly anything can be done with typefaces.

Now you can all be printing magicians with the Super "Color" Writer II. This program is one of the first in a line of new generation word processors which will revolutionize word processor software technology. You now possess one of the most powerful and versatile word processors available for any computer no matter the price! All these features and power are at your command from a keyboard as uncomplicated as that on the Color computer.

Really, word processing is pretty simple. All you do is type your ideas into your computer and they will appear on the screen. The Super "Color" Writer II then provides powerful and convenient functions that allow you to manipulate and perfect the text in every imaginable way, from insertion and deletion to moving around sentences and paragraphs to where you want them. You can easily move around your file to make all the changes you desire. While getting your ideas down you'll soon start thinking about adding special features for your final product such as centering your text, making commentaries, or putting in headers or footers. When you've done all this you'll want to begin to prepare your new "document" for printing. All this requires is a consideration of such things as margins, placement of headers and footers, line spacing, and decisions about what special features of your printer you want to use. Of course, you'll want to use the FORMAT WINDOW to make sure your decisions are properly implemented.

It's really pretty easy. How far you want to go will depend on your needs and desires. The Super "Color" Writer II will allow you to do anything with your text and your printer, so don't be afraid to experiment.

If you want to become somewhat more familiar with word processing basics, there are several good books available discussing word processing. For example, you could read INTRODUCTION TO WORD PROCESSING by Hal Glatzer, Sybex, Inc., 1981, or THE WORD PROCESSING HANDBOOK by Russell Allen Stultz, Spectrum, 1982. If you are new to word processing and haven't the time to read any of those works, be assured that this Tutorial will teach you what you need to know.

The best way to learn how to use the Super "Color" Writer II is to sit down at the computer, with this Tutorial at your side and your printer at your heels, and try each command to see how it works. In learning to use a word processing system, experience will be your best teacher. There simply is no way to adequately describe the results to be obtained from following the instructions in this Tutorial. You should take some time to become familiar with the commands. Within the constraints of the keyboard we have tried to make all commands easy to use and remember. You must also get to know your printer well, and learn how it interacts with this system. With a little practice, in no time at all you will be proficient.

1-B SYSTEM OVERVIEW

The Super "Color" Writer II is a full featured character (screen) oriented word processing system that combines simplicity and ease of use with features for even the most demanding professional. It was designed to have almost inexhaustible flexibility and power. The longer you use it the more you will be amazed by its great versatility. With the ability to put printer control codes anywhere within the text you have TOTAL CONTROL over all your printer features. With the FORMAT WINDOW you can view your text before printing, even if you have margins set for condensed mode printing at 132 or more characters per line!

The system provides nine different display options and four display color options. Text is manipulated through three modes: the TEXT mode, with its EDIT WINDOW, the COMMAND mode and the FORMAT WINDOW mode. In the TEXT mode, text is entered into the buffer, and a set of commands may be used to insert, delete, locate, etc. In the COMMAND mode, instructions such as clear file, load, save, color change, program function, etc., are used. The FORMAT WINDOW mode is used to display formatted text for final inspection before printing.

The Super "Color" Writer II can be used to edit any ASCII data including files that were generated by the Super "Color" Terminal, Super "Color" Calc and the Super "Color" Database. the system is designed for use with any printer and allows you to enter all control codes from the keyboard. A set of 27 print format parameters allows total control of how your text is formatted and sent to the printer, including margins, headers and footers, page numbering and baud rate. You may create customized printer drivers to control your printer's features and easily insert printer control codes while entering text. The system even has a set of 10 programmable codes to send character codes over decimal 127 as well as nulls, and control A-Z.

1-C CONVENTIONS USED IN THIS TUTORIAL

The syntax used in this Tutorial assumes that you are familiar with the glossary of terms found at the end of this Tutorial. Keyboard notation is consistent throughout, and is as follows: <CLEAR> indicates pressing the CLEAR key. <SHIFT><A> indicates holding the SHIFT key WHILE pressing the A key. <CLEAR>-<A> indicates pressing the CLEAR key and THEN pressing the A key. "CLEAR" and <C-L-E-A-R> indicate pressing the keys C-L-E-A-R.

IMPORTANT

To facilitate the printing of this Tutorial, many of the examples used throughout substitute printable characters, e.g., $,%,&,#,!, and *, and underlined characters for non-printable or command characters described in the text, e.g., format markers, printer codes and other marks having significance to the printer. This was done because if we used the non-printable or command characters, for the example, when printing the Tutorial the printer would either not be able to print the characters, not recognize the characters, or the characters would command the printer to perform the function instead of being printed for illustration. PLEASE, therefore, mentally substitute the mark described in the text for the printable mark we have used to indicate how and where the non-printable or command character should be placed.

SECTION 2 GETTING STARTED

2-A GETTING STARTED

BE SURE THAT THE JOYSTICKS ARE UNPLUGGED.

After the program has executed, a billboard displaying the proprietary information will be displayed. Press any key to enter the program itself. You will then be in the COMMAND MODE. The system awaits your command for such things as selecting your screen display and loading a file for editing or printing. Loading files is discussed below in Section 4. The first tasks you must consider are choosing your screen display size and color, and the size of your text line length. These topics are all discussed in the following sections.

2-B THE SCREEN DISPLAYS

Unless you are using a tape or disk version of the program in a 16K computer, which must use the 32 by 16 screen display, you are in the 51 by 21 display, i.e., you can enter 51 characters per line, with 21 lines per screen. If, after reading the rest of this section you decide that this is the display you want, you may exit from the COMMAND MODE into the TEXT MODE by pressing <BREAK> to begin entering text. The COMMAND and TEXT MODES are discussed below in Sections 2D and 2E.

The 51 by 21 screen display is only one of many display options available with the Super "Color" Writer II (to be distinguished from display WIDTHS in the EDIT WINDOW discussed below in Sections 2-D and 6-A). Super "Color" Library programs offer an astounding variety of display options. There are nine different display sizes and four different display color choices. This variety should satisfy just about every need and taste.

The display size options vary by the number of characters per line and the number of lines per page. You may choose from the standard 32 by 16 Color Computer display and several hi-res displays with lowercase characters: 51 characters per line and 21 lines on the screen (51 by 21), 51 by 24, 64 by 21, 64 by 24, 85 by 21 and 85 by 24. Lowercase is offered in all hi-res displays, with descenders on lowercase characters which extend below the line. Additionally, in the 64 by 21 and 64 by 24 displays, you have the choice of two character sets, a narrow one and a wide one. The difference between the 21 and 24 line per screen options is that some people may find the character set more attractive and easy to read with only 21 lines per screen since there is more space between lines. The color

options are black characters on a green or white background and green or white characters on a dark background (inverse display). (In the 32 by 16 display the color options are green or orange characters on a black background.) NOTE that in the inverted displays, although the background in the display area is dark, the border areas are either green or white (the same color as the characters). This is due to the video display generator graphics chip used in the Color Computer which we cannot change.

The most dense hi-res display, 85 characters per line, may be difficult for some to read well. The 85 by 21 or 24 display is nearly impossible to read when you have selected all uppercase characters. First of all the 85 character display is offered primarily for formatting rather than for editing or reading text. You should use it to look at the shape of your text to see if it was properly formatted. Of course, if your TV or monitor allows your to easily read this display, so much the better.

The possible inability to easily read this display is due to the inadequacies of the common color TV, the background color you have chosen, and/or the amount of interference you are experiencing. You can vary the color control on your Color TV; you also can try to eliminate sources of interference with your TV or monitor by working with the wires, etc. If none of these helps and you still desire to have a clear, crisp display, you may have to consider getting a new TV or monitor. This topic is discussed in Appendix G, "What To Do If Your TV Display Is Bard To Read," which includes a discussion of the limitations of the color TV and references if you find it necessary or desirable to convert your Color Computer for use with a special monitor.

OK, now you know the options. So what would make you choose any particular display? Aside from memory and speed considerations, the choice depends on your mood, your visual acuity, the quality of your TV or monitor and the degree of interference you are experiencing when using your computer. First the speed considerations. The 32 by 16 display is the standard Color Computer display and is built right into the hardware. On the other hand, the other displays are software generated, taking up memory space and requiring software routines to process and display on the screen. The extra processing time for the hi-res displays makes them somewhat slower than the 32 by 16 mode. Thus, for example, you will notice that when you use the 32 by 16 display scrolling is very fast, whereas, when you use the hi-res displays scrolling is somewhat slower. This is due to the extra processing time necessary. We assure you that the processing is as fast as is possible.

The memory aspect is similarly a result of a software, rather than hardware, source. The creation of the hi-res display eats 8K bytes of memory. At times, however, those 8K bytes become necessary so that you can load or work on a textfile. At that time you will have to select the 32 by 16 display and opt not to use the hi-res capability to give you the extra memory you need. This process is discussed in Section 2-I below.

Outside of these constraints you have wide latitude to match the Super "Color" Writer II to your tastes and monitor performance. Monitor performance was discussed above. Here it should be emphasized that the white background does not work very well on a color TV unless the color is turned down to give you a black and white display. This is because white, unlike green, is not one of the primary colors used by the color TV; instead, it is a composite of all three, red, blue and green. If you select the white display on a color TV you get color blooms and shadows which blotch up the characters and make them very difficult to read. It is best to either use the pure green display or turn the color down on your color TV.

As for your tastes, well, there's no accounting for tastes. What we hope is that given the available display options you will find at least one that perfectly suits your needs. The methods to select the different display options will be set out in the following paragraphs of this Tutorial.

The 32 by 16 display, the exclusive display for 16K tape and disk users, is the colorful display, making full use of the color potential of the Color Computer. It does not, however, feature true lowercase characters. Its lowercase characters are displayed just as they are in BASIC: light uppercase characters on a dark background. Numbers, symbols and punctuation will also be displayed in this way. The border and spaces are black. Only uppercase characters, control characters and markers will be light. The LCA-47 Lowercase Adapter sold by Micro Technical Products, Inc., Mesa, Arizona will give you true lowercase with the 32 by 16 mode. The hi-res displays all have true lowercase characters with descenders.

The system starts with the 51 by 21 hi-res mode. You may freely go between displays at any time by specifying a different line length, line number for the hi-res displays, and character set for the 64 display. To select the desired line length press <CLEAR>-<CLEAR> to enter the COMMAND mode, then type the number of the display line length you desire, e.g., 32, 51, 64, or 85, then press <ENTER>. With the 64 display you have one additional option: you may choose between a narrow or a wide character set. This is done by entering from the COMMAND mode either "64N" or "64W" and

then pressing <ENTER>. To select the desired number of lines per screen, either 21 or 24 (for the 51, 64 and 85 displays only), from the COMMAND mode press either 21 or 24, then press <ENTER>. Selection of color options is discussed below.

COMMAND LINE At the top of the screen, no matter the display, will always be a COMMAND line. On the right-hand side of the COMMAND line are two indicators. The "C" tells you the column number of the column position of the cursor from the left side of the page. The "L" on the extreme right-hand side of the COMMAND LINE is the "case status" indicator. "L" represents availability of both upper and lowercase and "U" represents availability of uppercase only. In the "L" mode the keyboard acts like an ordinary typewriter keyboard. Typing <SHIFT><0> toggles the case just as in BASIC. Later on you will learn that this "L" or "U" will turn to lowercase "l" and "u" when you are using the VERIFY feature during LOCATE AND CHANGE. See Section 3-D.

KEY BEEP AND ERROR SIGNAL If you want key beep, turn the volume of your television set up to hear it. (You may toggle key beep off and on by typing <B-E-E-P>-<ENTER> from the COMMAND mode.) Turning up the volume will also allow you to hear the ERROR detect warning beep which will sound whenever the system detects an error, such as in formatting, or when you load a faulty textfile.

Now to select the screen color. . . .

2-C CHANGING THE DISPLAY COLOR

When in the 32 display you may change the display color from green to orange; in the hi-res mode you may change the background color from green to white. To do so first press <CLEAR>-<CLEAR> to enter the COMMAND mode. Next press <C>-<ENTER> to toggle the display color between green and white (orange in the 32 display). You may press <BREAK> to exit to the TEXT mode. In the 32 display the display color option, along with the tint control on your color television set, give you a wide selection of video display colors to suit your mood.

You can also invert the screen in all displays, giving you green or white characters on a dark background in the hi-res displays or the same display as BASIC in the 32 display. This is done from the COMMAND mode by pressing <I>-<ENTER>, which toggles the inverse display. You will note that when the display is inverted, the screen borders will still be white or green. This is due to the Color Computer hardware and cannot be avoided.

After one last selection you will be ready to begin entering text: The line length for the EDIT WINDOW.

2-D TEXT MODE

The TEXT mode is where you'll be spending most of your time, so a thorough understanding of its operation is important. Before discussing text entry, the EDIT WINDOW and text WIDTH concepts will be explained.

THE EDIT WINDOW DISPLAY

There are three distinct but related elements of text display. The first is the screen display dimensions, discussed above. These dimensions are static, although several options are offered. The second is the text WIDTH display in the EDIT WINDOW, to be discussed here. The third is the display of how the text will be printed, discussed in Section 6 and Section 7.

All your text is written with the goal of being printed. Thus, ideally you want to have the line length displayed on the screen the same as that which you will print. Yet, none of the several display modes necessarily exactly fits the length of your finally printed line. Some are too short and some are too long. The text EDIT WINDOW function together with the WIDTH command were designed to solve this problem. The WIDTH command allows you to make the line length which can be displayed on the screen the same as your printed line length, up to 240 characters per line, no matter the display size, from 32 to 85 characters per line, even if you switch from one display MODE to another. This does not mean that you can display 85 characters per line at one time on the screen when in the 64 display. Instead, an EDIT WINDOW is used to view the number of characters allowed per line on the screen for the display, and when you reach the end of the line on the screen the system will automatically move over sixteen more columns to allow entry of more text, and so on until the designated WIDTH has been reached. As the system adds sixteen columns at one side it subtracts them from the other since the display line length is limited. Of course, if you have a WIDTH of 60 in the 64 display the whole line length will be in the EDIT WINDOW. Thus, when in the TEXT mode you are always in an EDIT WINDOW (to be distinguished from the FORMAT WINDOW discussed in Section 6-B). This window works both horizontally and vertically so that you can scroll anywhere within the text. All editing and formatting functions are input through this "WINDOW".

The WIDTH is initially set to 64, the standard printed line length. There is no need to select your printed line

length as your WIDTH. In fact, if you select a WIDTH equal to the display line length you have selected (32, 51, 64 or 85) you will be able to eliminate the need to use horizontal scrolling to read and edit your text, and will only have to scroll vertically to review and edit it. WE ACTUALLY RECOMMEND THAT YOU DO THIS FOR EDITING YOUR FILE. Since your final printed page may also be shorter than the display size you selected, e.g., a 60 character line in the 64 display, you may also wish to set the WIDTH shorter than your display size for a display of your text similar to the final printer product. NOTE, however, that since the text displayed will still contain special printing format information, and since format functions such as justification, centering, headers, etc. are not implemented in the TEXT mode, you will need to use the FORMAT WINDOW to view the text exactly as it will be printed (see Section 6).

To select your text WIDTH, if you desire other than the default WIDTH of 64, enter the COMMAND mode by pressing <CLEAR>-<CLEAR> and type <W-I-D-T-H> followed by the desired WIDTH, then press <ENTER>. Thus, if you desire the EDIT WINDOW display width to be the same as your 60 character printed line you would type "WIDTH60<ENTER>". The maximum WIDTH allowed is 240 characters per line. You may find a WIDTH equal to your display line length to be easy to use for editing your text, e.g., a WIDTH of 32, 51, 64 or 85.

ENTERING TEXT IN THE TEXT MODE

Once you have made these display selections, press <BREAK> to exit the COMMAND mode into the TEXT mode. In the TEXT mode, you can type three classes of characters into the text buffer: The first is PRINTABLE characters (letters, numbers, punctuation, symbols etc.). The second and third, which you will soon learn about, are NON PRINTABLE characters (printer control codes), and FUNCTION markers. PRINTABLE characters appear in the text buffer just as they will appear on the printed page (except in the 32 display where lowercase characters are displayed on the screen as light letters on a dark background as in BASIC). NON PRINTABLE characters appear as inverse characters (in the 32 display as either dark symbols on a light background or as graphics blocks), and will not be printed. Rather, they control special functions within the printer. FUNCTION markers also appear on the screen as inverse characters (in the 32 display as colored blocks) and will not go to the printer at all. NON PRINTABLE characters and FUNCTION markers are discussed in Sections 7-J and 5-B.

<BREAK>, <CLEAR> and <SHIFT><@> have special functions in the TEXT mode. Pressing <BREAK> halts the present command being executed by the system, acts as an escape from some commands, and breaks you out of the COMMAND mode.

<BREAK> is also an "UNDO" instruction in type-over (discussed below) and in the INSERT and DELETE character, word and line commands. This aspect of the use of the <BREAK> key is discussed in those sections (see Sections 3-B and 3-C). <CLEAR> acts as the function control key for the TEXT mode. When <CLEAR> is pressed, the cursor becomes a blinking underscore, indicating that the system is waiting for the next key to be entered (you can abort by pressing <BREAK>). When the function is entered, it is displayed on the command line. <SHIFT><@> is the only means to continuously repeat the last command entered. Commands will not repeat if the command key is held down; this may only be done by holding down <SHIFT><@>.

PRACTICE TEXT

We have devised some practice text for you to use to get to know the system. Before getting started with practice text, however, let's talk about the cursor. This blinking block will always let you know where you are. All your actions take place from the position of the cursor (i.e., they are "cursor oriented"). The cursor has three different appearances. In the normal TEXT mode it is a blinking block. In the CHARACTER INSERT mode, it is a blinking half-block (a blue block in the 32 display) (see Section 3-B). In the COMMAND mode it is a blinking underscore (a white half-block in the 32 display). These transformations have been designed to help prompt you to know which mode you are in.

Since the system was designed so that the cursor always "knows" where it has to be, you will have to learn some of the cursor's idiosyncracies. Some of these idiosyncracies will be stated in rules; others will be left for your discovery and accommodation. To get started type the following:

This is a test.<ENTER>

If you made a mistake or wish to change anything you've typed, just use the arrow keys to position the cursor over the character you wish to change and type the new character. This feature is called "type-over": you can type over and change already entered text. (The opposite of type-over is perpetual INSERT in which characters do not replace the text, but are inserted into it. This function is discussed in Section 3-B). When you reach the end of a line when typing-over you will automatically enter into the LINE INSERT mode (see Section 3-B) and the next line will open up for you to continue entering text. If you make a mistake when typing over a few words and have not gone on to the next line, press <BREAK> and the typed over text will be returned (the UNDO feature - see Section 3-B).

Notice the character after the word "test". This character represents a carriage return and is used to terminate a line. Also notice that the cursor has moved to the beginning of the next line. This is normal whenever you press <ENTER>. A word about carriage returns. Carriage returns are used to end blocks of text, be they one word, one line, or one long paragraph. Thus, if you wish to create lines of text shorter than the length of your print margin (see Section 7-B), be sure to use carriage returns after each shortened line.

Lets try another test (this test assumes that you are using the default 51 by 21 display with the EDIT WINDOW WIDTH of 64). Type the following:

This is a test of the Super "Color" Writer II
word wrap around feature.<ENTER>

Notice that when you reached the end of the screen, the display shifted giving you 16 more columns for text entry. This is the automatic scrolling in the EDIT WINDOW allowed because the default line length is 64 characters per line (WIDTH64), which is longer than the 51 display. Notice also that when you typed the word "feature" it was moved to the beginning of the next line on the screen and that it was not split in half. This is referred to as word wrap around. If the word being typed is too long to fit on the line you're on, the word is "wrapped around" to the next line. There will never be a need to hyphenate a word at the end of a line for a pleasing display unless YOU want to. The word wrap around feature may be toggled off and on for those of you who do not like it. To do so, press <CLEAR>-<CLEAR> to enter the COMMAND mode, then type <W-R-A-P>-<ENTER>.

When wrapping around the system does not move the space preceding the word, however. That space stays on the previous line, and you should remember this when you are manipulating text. If a sentence ends at the end of a line on the screen and is not terminated by a carriage return, the following sentence beginning on the next line will be indented one space (provided you put two spaces between sentences). This is because the previous line claims one of the spaces.

The last two tests were simple sentences and normally shouldn't be terminated with an <ENTER>. You will usually type some form of punctuation (i.e., .,?!) at the end of each sentence, only typing <ENTER> at the end of a paragraph. <ENTER> is also used to put a blank line between paragraphs or anywhere else. Another function for <ENTER>, is to create lines shorter than your margin length.

Now it's time for you to try typing a sentence or two on your own to get a feel for the system. If you wish to

indent, space over the amount you need, or press <SHIFT><LEFT ARROW> once for five spaces and twice for eight. (For other tabs see Section 2-G below.) To terminate the paragraph press <ENTER>. Feel free to move about using the arrow keys to change or correct anything you've typed. Also note that when typing text, if any key except <ENTER>, <BREAK>, <SHIFT> or <CLEAR> is held down for more than one-third of a second, the character or function will repeat on the screen. This feature is called key repeat. It allows you to quickly repeat key entries such as spaces or underline characters.

An additional feature which you will find immensely powerful is TYPE-AHEAD. TYPE-AHEAD is a feature which allows you to continue to type up to 256 keystrokes into the buffer while the screen or system is busy processing text. Once the system has returned to responding to your input it will immediately begin to process what you have previously entered into the TYPE-AHEAD buffer until it catches up to your current input or until you press <BREAK> to stop TYPE-AHEAD. This feature is extremely powerful, and once you become more familiar with the Super "Color" Writer II you will begin to use its power to do such things as type in a new set of commands while the system is saving a file or performing some other task. At times, such as when deleting text, this feature can be both a help and a nuisance, so you must be careful.

Now that you know how to put text on the screen, it's time to learn how to use the system to edit your text. But first. . . .

THE FOLLOWING SECTIONS provide other useful general information which helps you move through the text, get HELP in remembering commands, clear the buffer of text, etc.

2-E COMMANDS IN THE TEXT, COMMAND AND FORMAT WINDOW MODES

To manipulate your text for printing, saving, formatting and other functions you will have to learn the special commands devised for those tasks. The commands were created to be logical and easy to remember, so don't be afraid of being bowled over by a confusing mass of technicalities. The commands you will use fall into different categories.

Many of the commands may be entered directly from the TEXT mode, and require a one or two or sometimes three-keystroke sequence to implement. Single-keystroke commands control cursor movement and are implemented by using the <ARROW> keys. Single keystroke commands will repeat if held down. For two and three-keystroke commands you first press the <CLEAR> key, and then press the key(s)

for the particular TEXT mode command. When you press the <CLEAR> key, the cursor will turn white in the 32 display and become a blinking underscore in the hi-res displays awaiting your command key. Commands which are initiated by pressing <CLEAR> will not repeat if held down. You must press <SHIFT><@> to repeat such commands. These TEXT mode commands are summarized in APPENDIX C. Some of these commands require user input, such as the LOCATE command (see Section 3-D), and the PROGRAMMABLE FUNCTIONS (see Section 8). When input is required, the cursor will move to the COMMAND line at the top of the screen after a prompt and will await your input. The COMMAND line is also used with many commands for prompts requiring you to specify the number of times of execution, or to authorize execution by pressing <Y> for "Yes". Entry on the COMMAND line is limited for the two relevant TEXT mode commands to 21 characters for the LOCATE command (see Section 3-D) and 28 keystrokes for the PROGRAMMABLE FUNCTIONS (see Section 8-A). As in BASIC, <SHIFT><←> will delete any input on the COMMAND line and allow you to begin your input over again.

Another set of commands are COMMAND mode commands. The COMMAND mode is a special mode entered by pressing <CLEAR>-<CLEAR>. When you do this the cursor will appear on the COMMAND line at the top of the screen after the a "COMMAND:" prompt. COMMAND mode commands generally control tape and disk access, display options, text buffer clearing, and alteration of format parameters. COMMAND mode commands are summarized in APPENDIX C. These functions generally require input of character strings for performance and thus the need for the COMMAND line as a convenient place for such input. Input on the COMMAND line for COMMAND mode commands is limited by the screen display: 21 characters for the 32 display, 40 characters for the 51 display, 53 for the 64 display and 74 for the 85 display. As in BASIC, <SHIFT><←> will delete any input on the COMMAND line and allow you to begin your input over again. The COMMAND line will also provide necessary prompts when you are using the COMMAND mode. COMMAND mode commands are progressive in that you do not automatically exit from the COMMAND mode when the current command has been executed; instead, you remain in the COMMAND mode in case you wish to perform other commands. To exit the COMMAND mode you may either press <BREAK> or </>-<ENTER>.

A third mode is the FORMAT WINDOW mode (see Section 6) which also has its own set of commands for scrolling and movement of the text in the FORMAT WINDOW. These commands are summarized in APPENDIX C. The command line is used in this mode to provide prompts. The FORMAT WINDOW mode also has a separate status line at the bottom of the screen to inform the user of text position in the window.

2-F CURSOR CONTROL SUMMARY

The following keys move the cursor in the indicated directions in the TEXT mode (all single-keystroke commands repeat if the key is held down; others require pressing <SHIFT><@> for continuous repeat or re-implementation of the command again for single repeat):

MOVEMENT	KEY(S)
Cursor Up (Scrolling)	<UP ARROW>
Cursor Down (Scrolling)	<DOWN ARROW>
Cursor Left	<LEFT ARROW>
Cursor Right	<RIGHT ARROW>
Cursor Left One Word	<CLEAR>-<Q>
Cursor Right One Word	<CLEAR>-<W>
Cursor To beginning of line	<CLEAR>-<LEFT ARROW>
Cursor To end of line	<CLEAR>-<RIGHT ARROW>
Cursor Home (Top of Screen)	<CLEAR>-<SHIFT><LEFT ARROW>
Cursor Bottom of Screen	<CLEAR>-<SHIFT><RIGHT ARROW>
Page ahead one page of text	<CLEAR>-<DOWN ARROW>
Page back one page of text	<CLEAR>-<UP ARROW>
Cursor Top of Text	<CLEAR>-<SHIFT><UP ARROW>
Cursor End of Text	<CLEAR>-<SHIFT><DOWN ARROW>
Tab 5,8,16,24....72 (programmable)	<SHIFT><LEFT ARROW>

NOTE: Scrolling and paging can only be accomplished if there is at least one page plus one line of text in the buffer. When paging, the system will move ahead or back one full page minus one line, i.e., 14 lines in the 32 by 16 display, 19 lines in the 21 display and 22 lines in the 24 display (one line is devoted to the COMMAND line).

2-G TABS <SHIFT><LEFT ARROW>

To help with your indentation needs, the Super "Color" Writer II provides you with ten TAB positions (5,8,16,24,32,40,48,56,64,72). The cursor is tabbed to successive positions by pressing <SHIFT><LEFT ARROW> the number of times necessary to get to the desired TAB.

The TAB positions are also reprogrammable by you from 0 to 240 in case you have needs other than the default TAB values. Reprogramming is accomplished by replacing any or all of the ten default values. To display the default values press <CLEAR>-<CLEAR> to enter the COMMAND mode type <T-A-B> and press <ENTER>. The default values will be displayed on the COMMAND line. To change the defaults to new values press <CLEAR>-<CLEAR> to enter the COMMAND mode and type <T-A-B> and enter the new values, each separated by a comma, beginning with the lowest value. If you do not wish to change a particular value, replace that value in the sequence with a comma. End the sequence with an <ENTER>. The new values, with the changes, will be displayed.

EXAMPLE 1: <CLEAR>-<CLEAR>TAB<ENTER>
EXAMPLE 2: <CLEAR>-<CLEAR>TAB4,,,18,25,30<ENTER>
EXAMPLE 3: <CLEAR>-<CLEAR>TAB,10<ENTER>

The first example will display the current tab values. The second will reprogram the first, fourth, fifth and sixth tabs to the indicated values while leaving the second, third and seventh thru tenth tabs unchanged. The third example will reprogram the second tab to the indicated value while leaving the other nine tabs unchanged. (NOTE that if you wish to program over six values in the 32 display, you will have to do it in two steps since the display size limits the number of characters which can be programmed. To do this, program the first six, then use commas for these when programming the second part.)

2-H CLEARING THE TEXT BUFFER AND EXITING TO BASIC

There are three options to choose from when clearing the text buffer, plus two other buffer related commands:

"CLEAR"	Clears the ENTIRE textfile.
"AFTER"	Clears all text AFTER the cursor position.
"BEFORE"	Clears all text BEFORE the cursor position.
"OOPS"	Recalls text mistakenly CLEARed or AFTERed.
"END"	Exits the program to go back to BASIC.

To clear all or part of the text buffer, press <CLEAR>-<CLEAR> to enter the COMMAND mode. Now type the desired command, followed by an <ENTER>. The system will respond with the prompt "ARE YOU SURE?". Press <Y> to execute the command. Pressing any other key will abort the command and return you to the TEXT mode. After the command has executed the system will remain in the COMMAND mode so that you can immediately access another file. If you accidentally clear the buffer using the "CLEAR" or "AFTER" commands (BUT NOT THE "BEFORE" COMMAND!) enter the COMMAND mode and type OOPS <ENTER> and the file will return. The first character of the cleared text, however, will have been changed to a space, so you must restore it to its original value.

You may at times desire to conveniently permanently exit the Super "Color" Writer II to BASIC to use another program, etc. To do so, type "END <ENTER>" from the COMMAND MODE. You will then be prompted "ARE YOU SURE?" Press <Y> to exit.

2-I CLEARING THE HI-RES DISPLAYS FOR EXTRA MEMORY

The eight hi-res displays offered by the Super "Color" Writer II are software generated and consume around 8K of

work area which you could use. Although in Color Computers with more memory you will probably never have a need for that extra 8K of work area, in smaller computers you probably will. The system therefore allows you to opt not to use the hi-res displays and instead use the memory devoted to them for your text. When you choose this option it is PERMANENT; you will have to reload the program to re-obtain use of the hi-res displays. You do not have to choose to have the extra memory until the system gives you the "FULL" message when your work area is full. When you want to have the extra memory, enter the COMMAND mode and type "DUMP <ENTER>". You will then be prompted "ARE YOU SURE?". Press <Y> to DUMP the hi-res displays.

2-J DISPLAYING AVAILABLE MEMORY <CLEAR>-<;> AND WORD COUNT

Your text is stored in the buffer in BYTES of data. Each byte essentially amounts to a character, including spaces and markers. Thus, the word "the" is stored in the buffer as three bytes of data. The Super "Color" Writer II allows you to display both the amount of memory bytes remaining to be used and the amount of memory bytes already used. To display the memory, press <CLEAR>-<;>. The number of bytes left to be used will be displayed on the COMMAND line to the left followed by the number of bytes used. In the DISK VERSION the COMMAND line will also show the number of granules the textfile will occupy on the diskette.

You may also find out the the number of "words" in your file, for example, for those picky teachers who want only 500 words. The system actually counts the words in the file and the count is very accurate. To get a word count type <W-O-R-D>-<ENTER> from the COMMAND mode. The word count will then be displayed on the COMMAND line.

2-K HELP <CLEAR>-<?>

A HELP table is built into the system as a quick reference to aid you with the most often used commands and markers. The HELP table comprises several "pages". Type <CLEAR>-<?> to display the first "page" of the HELP table, press any key to go to the next "pages". Pressing <BREAK> will exit to the text mode from any page; pressing any key will exit to the text mode from the last page. HELP may be accessed at any time without destroying any of the text that you have entered. Use HELP as a reminder when you are first getting started but not as a substitute for the manual since it contains only the most important commands, markers and functions available to you.

SECTION 3 EDITING FUNCTIONS

3-A EDITING FUNCTIONS

Now that you are familiar with the screen, the cursor movements and the rudiments of creating a textfile, managing usable memory, clearing the buffer and getting HELP, it's time to learn the editing commands. To be a word processor instead of a text editor requires sophisticated editing functions. The system must allow free insertion and deletion of text, it must allow flexible movements of blocks of text, and it must provide a versatile ability to locate strings of text for manipulation such as changing and deletion. The Super "Color" Writer II meets all of these requirements. The following sections describe this system's sophisticated editing commands. Note that editing is faster in the 32 display than in the hi-res displays. You may therefore consider doing your editing in the 32 display and your formatting in the hi-res displays.

3-B INSERT

The Super "Color" Writer II has two insert commands: INSERT CHARACTER and INSERT LINE. Inserting text is useful whenever you change your mind about what you previously typed or if you would like to add additional text to that which you previously typed. The INSERT LINE command will be welcomed by programmers who constantly insert lines in their programs. A "FULL" message will appear on the COMMAND line if the text buffer is filled during INSERT. With the added feature of fast TYPE-AHEAD you will not have to wait for the computer to catch up each time you wrap around to the next line when in the insert modes.

INSERT CHARACTER <CLEAR>-<I>

The INSERT CHARACTER function in the Super "Color" Writer II is always on until you turn it off, and vice versa. The INSERT mode is useful for many editing tasks, and some people prefer to always have insert on. It should be noted that character insertion, like all editing, is faster in the 32 display than in the hi-res displays.

When the INSERT CHARACTER mode is off you are in the type-over mode. In the type-over mode the cursor is a flashing block. You will be able to type-over text from the cursor position until you reach the end of the line, at which time the text will open up and you will enter the LINE INSERT mode discussed below. Type-over is on when the program begins so you will have to switch to the INSERT mode if you want that mode instead. To enter or leave the character INSERT mode press <CLEAR>-<I>. When in the INSERT

mode the cursor will be blue in the 32 display and a blinking half-block in the other hi-res modes. To INSERT a CHARACTER within the text buffer, position the cursor immediately over the character before which you desire to insert a character and press the key for the character. You may continue typing your text from there. The text will be inserted ahead of the cursor. You will stay in the CHARACTER INSERT mode even after you have wrapped around to the next line. To open a line to enter text, use the LINE INSERT command discussed below.

NOTE: If you desire to add carriage returns (or anything else for that matter) before some block of text, it is best to INSERT them instead of generating them by successively pressing <ENTER>. The latter practice is in reality changing characters of the following text into carriage returns. As you will see if you try it, carriage returns "eat" characters from the following text unless INSERTED.

THE UNDO COMMAND

If you have inserted text and desire to delete the inserted text and restore it as it was, press <BREAK>. This is an "UNDO" edit function command which restores the line you were working on to its former status. This UNDO command works with the INSERT and DELETE commands and the type-over mode ONLY. In other contexts the <BREAK> key will exit from a command or function. The UNDO command only works to restore the original line. If, while inserting text, you wrap around to the next line, the previous line is saved and the original line totally eliminated. The UNDO command will then only serve to clear the line you are working on.

INSERT LINE <CLEAR>-<O>

The LINE INSERT command allows for perpetual line insert. To INSERT a LINE within the text buffer, position the cursor immediately over the character from which you desire to begin inserting new text and press <CLEAR>-<O>. (The LINE INSERT mode is also entered when you have typed over text and reached the end of the line.) The text buffer will open up to allow any amount of text to be inserted. Lines of text will continue to be INSERTED, even when you have pressed <ENTER> to end a paragraph. Pressing <UP ARROW> or <DOWN ARROW> will close up the text buffer and exit back to the TEXT mode. Pressing the arrow keys again will then take you off the line on which you are working. In the LINE INSERT command <BREAK> is an "UNDO" instruction. It will restore the line where you are working to its original state. If you are on the line from which you began INSERTING LINES and have not yet wrapped around, pressing <BREAK> will return you to the original status. If, however, you have already wrapped around, pressing <BREAK> will clear the line on which you were entering text.

If you press the <UP ARROW> or <DOWN ARROW> the text will close up joining your inserted text with the text where you initiated the LINE INSERTION. Frequently you will be using LINE INSERT to add sentences or even paragraphs. While INSERTING this new text you may decide that you want to alter something that you just wrote or you might wish to check to see if what you are going to write will be consistent with what you have already written. If such an alteration or verification will be on the same line that you are on, you may go back in the line without any problem. Yet, usually, to make such alterations or verifications will require that you move up or elsewhere in the file with the up or down arrow keys. If you do, and you are in mid sentence, to avoid the text closing up leave a carriage return where you want to leave off, make your alteration, and then come back to the carriage return and type right over it to continue.

NOTE: While in either the INSERT CHARACTER or INSERT LINE mode, the DELETE commands (see 3-C below) can be used.

PROGRAMMABLE STRING INSERT

The system allows you to insert strings of text throughout your textfile by use of a PROGRAMMABLE FUNCTION or PROGRAMMABLE CHARACTER CODES. See Sections 8-B and 7-I.

3-C DELETE

The system has eight DELETE commands. Six of the commands are for deleting characters or lines from the text file. These are cursor oriented, as they will affect the character or string from where the cursor is located. The seventh and eighth are for deleting blocks and block markers and are discussed in Section 3-H, BLOCK MANIPULATION.

The table below lists the DELETE commands and their respective functions. Once a DELETE command is selected, it can be repeated by holding <SHIFT><@>. All deletions except the DELETE LINE command may be undone immediately after execution with the UNDO command (Section 3-A) by pressing <BREAK>.

FUNCTION	DELETE	KEY(S)
DELETE CHARACTER	Character at cursor	<CLEAR>-<D>
DELETE TO BEGIN	Cursor to beginning of line	<CLEAR>-<A>
DELETE TO END	Cursor to end of line (not CR)	<CLEAR>-<S>
DELETE LINE	Entire line at cursor (not CR)	<CLEAR>-<F>
DELETE LEFT WORD	Word to the left of cursor	<CLEAR>-<H>
DELETE RIGHT WORD	Word to the right of cursor	<CLEAR>-<J>
DELETE BLOCK	Block of text between markers	<CLEAR>-<V>
DELETE MARK	Specified block markers	<CLEAR>-<Y>

The DELETE LEFT and RIGHT WORD commands delete the previous or next word up to the next space, NONSPACE or punctuation mark (,.!?;:). It treats hyphenated words as one word for deletion purposes. The DELETE TO END command will not delete end-line carriage returns.

3-D LOCATE <CLEAR>-<L>

The Super "Color" Writer II has a very sophisticated global LOCATE command. LOCATE allows you to LOCATE, LOCATE AND CHANGE, and LOCATE AND DELETE any string of characters, including characters followed by an <ENTER>, while being able to control the number of changes or deletions to be made. In addition, it uses a MASKing capability that allows you to locate a string regardless of whether it is uppercase, lowercase, or any mixture. The LOCATE command also has a WILD CARD character which, when used in the string to be located, will act as a "don't care" character.

LOCATE STRING

To perform any LOCATE function the cursor must first be moved to a position in the textfile above the string to be located or to the top of the textfile - only strings below the cursor can be located. To LOCATE a character or string of characters, press <CLEAR>-<L>. Now type any string of text, up to 21 characters long, which you wish to locate within the text buffer (this could include <ENTER>s, CONTROL CODEs and MARKERs). When you are finished, press the <DOWN ARROW> to actually locate the string. When the string is located, it will be displayed on the screen with the cursor after it. By pressing <CLEAR>-<C> you will command the system to continue to the next occurrence of the string within the text buffer. For easy LOCATION, hold the <CLEAR>-<C> to continue to the end of the text buffer. If the string is not found, "NONE LOCATED" will be displayed.

LOCATE MASKED STRING

Often you will wish to LOCATE a word but you won't know if all or any part of it is in uppercase (capitalized). For example, you might wish to find all occurrences of the word "therefore", but you know that sometimes it is used at the beginning of a sentence and would thus have the first letter capitalized. With the Super "Color" Writer II there is no need to do two or more separate searches. Instead you may use the MASK capability with the LOCATE command so the system will ignore the case of the string to be searched. The MASK must FIRST be enabled by pressing <CLEAR>-<-> which toggles the MASK. (Pressing <CLEAR>-<-> again will disable the MASK.) The MASK cannot be enabled while actually performing the LOCATE function. When the MASK is enabled an "M" will appear to the left of the case status indicator on the COMMAND line. After enabling the MASK, follow the instructions set out in the above paragraph for LOCATE.

LOCATE STRING USING WILD CARD

Often you will need to LOCATE a word that you think you may have misspelled, or the spelling of which you are not sure. To do this you must be able to replace the question-raising characters with surrogate characters which the system will ignore except for the purpose of determining word length. These surrogate characters are called WILD CARD characters. To illustrate the use of the WILD CARD character during LOCATE, suppose the string to be located is "recieve" (sic.) or "receive". To locate either spelling, use the WILD CARD characters instead of the "ie" or "ei" when typing the string. The <RIGHT ARROW> generates the WILD CARD character while you are using the LOCATE command.

Using the above example, the procedure would be as follows: Type <CLEAR>-<L> for LOCATE. Then type <r-e-c> <RIGHT ARROW> <RIGHT ARROW> <v-e> <DOWN ARROW>. The system will search through the text buffer to locate "recXXve", and locate either spelling.

LOCATE AND CHANGE STRING

The LOCATE AND CHANGE function is extremely powerful. It allows you to change only one occurrence, a specified number of occurrences, or to change multiple occurrences while specifying, upon inspection, whether each consecutive occurrence should or should not be changed.

SINGLE OCCURRENCE EXECUTION

To LOCATE AND CHANGE a single occurrence of a character or string of characters, press <CLEAR>-<L>. Now type the string of text which you wish to locate within the text buffer. When you are finished, press the <UP ARROW>. Now type the string of text (up to 21 characters long) with which you wish to replace the original string. Pressing the <DOWN ARROW> will locate and change a single occurrence of the character or string of characters specified. When the original string is located, it is changed and its replacement is displayed on the screen with the cursor after it. If the original string cannot be found, "NONE LOCATED" will be displayed on the command line. Again, <CLEAR>-<C> is the command which allows you to continue LOCATING and CHANGING the string in the file.

MULTIPLE EXECUTION

To LOCATE AND CHANGE the number of occurrences you specify, follow the instructions for LOCATE AND CHANGE for single occurrences above, except that instead of pressing the <DOWN ARROW> to LOCATE AND CHANGE, press the <UP ARROW>. A "#" prompt will then appear on the COMMAND line asking the number of times you wish the CHANGE AND LOCATE to be performed. After providing the number (999 should do the

whole file), press <ENTER>. The specified number of occurrences will be LOCATED AND CHANGED and the command line will indicate the number of changes made.

VERIFIED MULTIPLE EXECUTION

Perhaps you want to change some but not all occurrences of a particular string in the file. To do this you may enable the VERIFY function and perform a LOCATE AND CHANGE. The VERIFY function allows you to sequentially locate every occurrence of a string in the file and decide whether to implement the CHANGE in each particular instance. TO USE THIS FUNCTION, VERIFY MUST BE ENABLED BEFORE THE LOCATE AND CHANGE COMMAND IS USED. To enable VERIFY press <CLEAR>-<ENTER>. Enabling of VERIFY will be indicated on the command line by a change in the case-status indicator: instead of the L or U being uppercase, it will be lowercase, (light on a dark background in the 32 display).

Once VERIFY is enabled, you follow the normal procedure for LOCATE AND CHANGE of a specified number of occurrences described above. Instead of all changes being automatically made upon pressing <ENTER>, the system will find the first occurrence of the string and wait for your response. Pressing <Y> will cause the CHANGE to be made, and the system will then proceed to the next occurrence; pressing <N> or any other key except <BREAK> will cause the system to go immediately to the next occurrence without making any CHANGE. Pressing <BREAK> at any time will break out of LOCATE AND CHANGE. After changing the final occurrence in your file, the system will indicate on the command line the total number of occurrences that have been changed. Pressing <Y> again will give a "NONE LOCATED" prompt.

WILD CARD IN LOCATE AND CHANGE

The WILD CARD can also be used in the LOCATE AND CHANGE function to change a character or string of characters which could be attached to many different strings of characters. This could be very helpful to correct spelling, grammatical or formatting errors. When a string of characters to be LOCATED is given WILD CARD characters (by pressing <RIGHT ARROW>), and the new string also has WILD CARD characters in the same positions in the string, those positions will not be affected by the change.

For example, suppose you are creating a work with a chapter having section numbers such as "8.1A-8.9A." After creating several other chapters with other numbers, you decide that chapter eight should now be chapter five, that the period between the chapter number and the section number should be a dash, and that the letter A at the end should be preserved (i.e., 8.1A to 5-1A). The easiest and safest way to change all the "8."s to "5-"s and preserve the rest of the number would be to use the WILD CARD.

This would be accomplished as follows: Type <CLEAR>-<L> to begin the LOCATE function and type the string to be searched: <8>-<.>-<RIGHT ARROW>-<A>; then press the <UP ARROW> to make a CHANGE and enter the new string: <5>-<->-<RIGHT ARROW>-<A>; next you press the <UP ARROW>, enter the number of times you want the change made, and then press <ENTER>. NOTE that if you had desired to use VERIFY you would have had to enable it before beginning the LOCATE function. Further, note that if WILD CARD characters are used on both the locate and the change lines, both MUST be of the same length.

When you press <ENTER> the system will locate the specified number of occurrences of "8.XA" and change them to "5-XA". Of course you could just LOCATE AND CHANGE all "8."s to "5-"s without using WILD CARD characters, but it is possible that somewhere in your file you may have an eight followed by a period that is not a chapter and section indicator that you do not wish to change to a "5-".

LOCATE AND DELETE - SINGLE OCCURRENCE

To LOCATE AND DELETE a character or string of characters, press <CLEAR>-<L>. Now type the string of text which you wish to locate within the text buffer. When you are finished, press the <UP ARROW>, which enters a string containing nothing, then the <DOWN ARROW> to delete a single occurrence. When the string is located and deleted, its previous position is displayed on the screen with the cursor after it. If the string cannot be found, "NONE LOCATED" will be displayed on the command line.

MULTIPLE EXECUTION / VERIFY

If you wish to LOCATE AND DELETE a specified number of occurrences, or all occurrences in the file, press <CLEAR>-<L> to enter the LOCATE function, enter the string to be deleted, press the <UP ARROW> twice, answer the "#" prompt with the number of occurrences to be deleted (999 for all in the file) and press <ENTER>. To use the VERIFY function FIRST enable VERIFY by pressing <CLEAR>-<ENTER>, then follow the procedures set out immediately above, and answer the prompt when the system stops at each occurrence.

3-E BLOCK MANIPULATION <CLEAR>-<B>; <CLEAR>-<Bø> TO <B9>

The Super "Color" Writer II allows you to move, copy and delete blocks of text, for example, moving, copying or deleting paragraphs, sentences, titles, etc. You may even deal with up to ten blocks of text at the same time.

You are offered two modes of operation, essentially for your convenience. One, with which the system starts, only allows one block manipulation, and gives you one BLOCK marker; the other gives ten different block markers and allows ten block manipulations. The difference is that in the single BLOCK mode all BLOCK commands, except the BLOCK DELETE command, are not prompted; while in the ten BLOCK mode all BLOCK commands are prompted to determine which block of text is to be affected by the command. As you can see, the single BLOCK mode is more convenient if you are only interested in manipulating one block of text since BLOCK marker entry is easier and there will be no BLOCK number prompting. Still, you may switch at any time to and from the ten BLOCK mode. To toggle between modes, enter the COMMAND mode and type "BLOCK <ENTER>".

To use the BLOCK commands, the text to be manipulated must first be marked with identical BLOCK markers. BLOCK MARKERS are created differently depending on the mode. In the single BLOCK mode, BLOCK markers are created by pressing <CLEAR>-<B>; in the ten BLOCK mode BLOCK markers are created by pressing <CLEAR>-<B> followed by the number of the block you intend to use, from ø through 9. BLOCK markers appear as an inverse "b" plus a "ø" (an ORANGE graphics block plus a "ø" in the 32 in display) in the single BLOCK mode, and are followed by a number from ø to 9 in the ten BLOCK mode.

BLOCK markers must be used in pairs placed before AND after the BLOCK to be manipulated. NOTE that in the ten BLOCK mode block markers MUST be used in IDENTICAL PAIRS. The block of text to be manipulated must be marked with block markers of the same number, i.e., a <Bø> would mark the beginning and end of a block of text to be moved. Remember that spaces at the beginning of a paragraph are recognized by the system, and so should be preceded by the BLOCK marker when doing BLOCK manipulations if you want them included in the move, etc.

BLOCK markers are automatically inserted when created. If there aren't exactly two BLOCK markers of the same type in the text buffer during any BLOCK manipulation attempt, "MARKER ERROR" will appear on the command line. To delete BLOCK markers of a given type from the text buffer, press <CLEAR>-<Y>. If you are in the ten BLOCK mode, you will be prompted for the specific BLOCK number before markers of that type will be deleted. You may also respond with <*>

which will cause ALL BLOCK markers in your textfile to be deleted. There is no "UNDO" command to restore mistakenly performed BLOCK manipulations, SO BE CAREFUL!

As mentioned above, manipulations in the ten BLOCK mode are prompted. Thus, when you press the command keys (<CLEAR>-<M>, etc.) the system will prompt "#" to request the number of block you wish to manipulate, controlled by the BLOCK marker pair you have used. When you press the number the command will be implemented, except with the BLOCK DELETE command which will then elicit an "ARE YOU SURE?" prompt to which you must answer <Y> (plus the BLOCK number in the ten BLOCK mode) to have the command executed.

NOTE: WHEN IN THE TEN BLOCK MODE, BLOCK MARKER PAIRS MAY BE USED WITHIN OTHER BLOCK MARKER PAIRS FOR BLOCK MANIPULATION, BUT CARE MUST BE TAKEN TO AVOID GETTING CONFUSED, WHICH COULD RESULT IN THE MANIPULATION OF THE WRONG BLOCK.

BLOCK COPY <CLEAR>-<X>

To copy (xerox) a block of text, mark the block to be copied with block markers as described above. Move the cursor to the position in the text where the duplicate block is to be located and press <CLEAR>-<X> (plus the BLOCK number in the ten BLOCK mode). The block will be copied at the new position, and the original block will remain where it was, BLOCK markers and all. If you wish to duplicate the block more than once, hold <SHIFT><@> to repeat the function. "FULL" will be displayed on the command line if you don't have enough memory to copy the entire block of text. When this happens try copying the block of text in smaller pieces, piece by piece. Be sure to delete the BLOCK markers after you're done with the BLOCK COPY, unless you want to manipulate that text some more. To delete the BLOCK markers and start over press <CLEAR>-<Y> (plus the number in the ten BLOCK mode).

BLOCK MOVE <CLEAR>-<M>

Do not confuse BLOCK MOVE with BLOCK COPY. The difference between the two is that BLOCK MOVE moves the block of text to the new cursor position AND DELETES THE OLD BLOCK INCLUDING BLOCK MARKERS AT THE SAME TIME; BLOCK COPY moves the block to the new cursor position but leaves the old block intact, including block markers. To move a block of text, follow the instructions for BLOCK COPY above but press <CLEAR>-<M> (plus the number in the ten BLOCK mode).

BLOCK DELETE <CLEAR>-<V>

BLOCK DELETE is easily accomplished. Mark the block to be deleted as described above and press <CLEAR>-<V> (plus the BLOCK number in the ten BLOCK mode). The command line will then say "ARE YOU SURE?" Pressing <Y> will delete both the text marked and the block markers. Any other response will exit to the TEXT mode.

SECTION 4 SAVING AND LOADING TEXTFILES

TAPE FILE ACCESS

4-A SAVING TEXTFILES TO TAPE

The Super "Color" Writer II allows you to save textfiles using a portable cassette recorder. The entire textfile, or any portion of it, may be saved to tape to be reloaded at any time. (See Appendices A and B for details about loading, creating, editing and saving BASIC programs and ASCII files usable by many utility programs including Super "Color" Library programs.)

For your convenience the Super "Color" Writer II uses progressive commands. This means that after any tape command is executed you do not automatically go back to the TEXT mode, but you can either enter the next command directly or press <BREAK> to return to the TEXT mode.

Saving is a cursor-oriented function - you can save all text from the current cursor position to the end of the buffer. For example, if you want to save the bottom half of your textfile, move the cursor to that position and save; if you want to save the entire file contents, be sure to move the cursor to the beginning of the file and save.

To save, first move the cursor to the position from which you wish to save the file, and press <CLEAR>-<CLEAR> to enter the COMMAND mode. Place the recorder in the record mode, type "CS" followed by a filename not to exceed eight characters and press <ENTER>. If no filename is specified, the textfile will be saved with the last filename used. If no filename has been used so far, the textfile will be saved without a filename. The file contents, beginning with the current cursor position, will be saved to the tape. If you do save a partial file, the system will indicate this by sounding the ERROR detect signal and giving a "(FILENAME) PART SAVED" message on the COMMAND line to give you a warning in case you did not wish to save a partial file. When the textfile has been saved, the system will return to the COMMAND mode, prompting "(FILENAME) SAVED." To display the last filename used, press <CLEAR>-<CLEAR> to enter the COMMAND mode, then type <C>-<N>-<ENTER>. The last filename will be displayed on the COMMAND line.

Sometimes after you have begun to save a textfile you may decide that you want to abort the save. You may abort the save by pressing <BREAK>.

4-B LOADING A TEXTFILE FROM CASSETTE

You may load any ASCII (not binary) file into the Super "Color" Writer II. To load a textfile from tape, press <CLEAR>-<CLEAR> to enter the COMMAND mode. Ready the cassette recorder, type "CL" followed by <ENTER>, then press "PLAY". The filename, if any, will appear on the COMMAND line. When the file is loaded, the COMMAND line will show the message "(FILENAME) LOADED" with the cursor after it. Press <BREAK> to enter the TEXT mode. If an error occurs during loading, "ERROR" will be displayed on the COMMAND line, the error detect warning will sound and the system will return to the COMMAND mode displaying the partially loaded textfile. To start over you should clear the buffer and try loading again. If you are still unsuccessful, see Section 4-E.

4-C APPENDING TEXTFILES

The system allows you to load as many textfiles as will fit in the text buffer. To append one textfile after another, load the first then load the second. If the file to be loaded is too large, the system will accept all of the file that will fit in the buffer, and "FULL" will be displayed on the COMMAND line.

4-D TAPE FILE LINKING

Tape file linking is discussed completely in Section 8-A, PROGRAMMABLE FUNCTIONS.

4-E RECOVERING TEXTFILES FROM BAD TAPES

Sometimes, because of a bad tape, bad line connection, or some other problem, your file saved to tape will be bad. The system allows you to recover most of your saved file. The Super "Color" Writer II uses the same ASCII tape format as the Color Computer. The tape format consists of one start block followed by separate data blocks. If a particular data block causes an error during loading, the rest of the textfile can be recovered by proceeding as follows:

1) After loading the textfile and encountering the error, note the position of the tape counter to mark the spot.

2) Rewind the tape to the beginning of the file and re-load the tape JUST to the point where the filename appears on the COMMAND line.

3) Fast-forward the tape to the place where the error occurred.

4) Unplug the earphone plug to listen to the tape to position it at the beginning of the next data block.

5) Plug the earphone plug back in and press "PLAY". The remainder of the textfile will load in with the exception of the block of data with the error. You may then go back and repair your textfile.

If the error occurs in the initial start block which serves to tell the system that a file has begun, you may still load the file by doing the following:

1) Take the tape containing any textfile created and saved with the Super "Color" Writer II and load it JUST the point where the filename appears on the command line.

2) Return to the original tape and locate the bad initial block. Unplug the earphone plug to listen to the tape to position it at the beginning of the next data block.

3) Begin to load the defective textfile with the first good block.

4) Once the file has loaded, you may go in and make any repairs if necessary, and then resave the file with the original filename.

DISK FILE ACCESS (DISK VERSION ONLY)

4-F DISK COMMANDS AND SYSTEM DEFAULTS

For your convenience the Super "Color" Writer II uses progressive commands. This means that after any disk command is executed you do not automatically go back to the TEXT mode, but you can either enter the next command directly or press <BREAK> to return to the TEXT mode.

The Super "Color" Writer II initializes with a default extension of "/SCW" and a default drive number of ":Ø". These defaults can be changed at any time. The default extension assumes the last extension used and is changed by entering a new extension during any disk access ("DS" to SAVE, "DL" to LOAD, "DN" to check the DISKFILE NAME, "DR" to RENAME or "DK" to KILL). The same is the case with the default drive number. Changing defaults will be discussed more in later sections.

4-G DISPLAYING A DISKETTE DIRECTORY

To display a disk directory, mount a diskette in Drive Ø (default drive), type <CLEAR>-<CLEAR> to enter the COMMAND mode, then press <D>-<I>-<ENTER>. The directory for Drive Ø will be displayed in a two-column format with the drive number and number of free granules displayed on the COMMAND line. If the number of entries exceeds 1 page the display will pause. To continue to the second page of entries press any key except <BREAK>. A maximum of 68 entries are allowed on the disk. When you have found the file that you are looking for, press <BREAK> to escape the directory and freeze the last directory page displayed. Each entry on the directory will be followed by a number indicating the number of granules allocated to the file on the disk, and the letter "A" for ASCII or "B" for Binary to indicate the nature of the file.

Sometimes, due to disk operating system failure or some other reason, your disk may have been "crashed". The Super "Color" Writer II permits you to LOAD files from a crashed disk. When you call up the directory of a crashed disk some of the files of the disk will be followed by an asterisk. Those files MAY NOT be loaded. The asterisk indicates that those files are bad. The other files may be loaded. Be sure to load these files and save them on another disk. Once all the good files have been saved from the bad disk, you then should bulk erase the disk, re-initialize it and use it again to save files - if the disk has not been totally ruined. DO NOT SAVE FILES ONTO A BAD DISK SINCE THEY WILL NOT BE SAVED PROPERLY! If you have Super "Color" Disk-ZAP you should be able to fix the underlying error.

If you wish to display the directory for a drive other than Drive 0 type <D>-<I> followed by the drive number and then press <ENTER>. The directory will be displayed for the drive number that you specified.

EXAMPLE: DI3<ENTER>

This example will display the directory for the diskette mounted in Drive 3.

4-H SAVING A TEXTFILE TO THE DISKETTE

The Super "Color" Writer II saves textfiles to the diskette in the ASCII format for compatibility with Super "Color" Library files, BASIC programs saved in ASCII (see your BASIC manual) and all other programs using pure ASCII.

When the Super "Color" Writer II saves a textfile to a specific drive it first checks to see if the file already exists on the diskette mounted in the drive specified. If the file exists the system will save the new textfile over the old one. If the file does not exist the textfile will be saved to the diskette mounted in the drive specified. The Super "Color" Writer II always VERIFIES when it saves to disk to assure that your textfile has been saved error free.

IMPORTANT

From time to time the Color Computer operating system may ERROR while saving the textfile. The exact error encountered will be displayed on the COMMAND line and the ERROR warning signal will sound. If an error such as this should occur, resave the textfile on both the same diskette and a backup diskette as well. Errors during the save, if undetected, would cause you to lose all of your hard work. To avoid a calamity of this sort it is recommended that you ALWAYS save your files on two separate diskettes. Then if you have difficulty with one of your files, you will still have a copy of your work. If you own the Super "Color" Disk-ZAP you should be able to repair the underlying disk error.

DISKETTES ARE CHEAP; YOUR TIME ISN'T!!

Saving is a cursor-oriented function - you can save all text from the current cursor position to the end of the buffer. For example, if you want to save the bottom half of your textfile, move the cursor to that position and save; if you want to save the entire file contents, move the cursor to the beginning of the file and save. Since more often than not users do not wish to save partial files, whenever you attempt to save a partial file the error detect signal will sound and the message "(FILENAME) PART SAVED" will

appear on the COMMAND line. This will make you check to be sure that you want to save a partial file, and we hope it will save you from a potential "disaster".

To save, first move the cursor to the position from which you wish to save the file, and press <CLEAR>-<CLEAR> to enter the COMMAND mode. Mount a formatted diskette in Drive Ø. DO NOT USE THE Super "Color" Writer II MASTER DISKETTE. Type <D>-<S> followed by the filename (not to exceed 8 characters) and press <ENTER>. The prompt "ARE YOU SURE?" will appear ONLY when a file with the same filename exists on the disk. Pressing <Y> will cause the SAVE; pressing any other key will abort the SAVE. In all other cases the SAVE is automatic.

EXAMPLE: DSTESTFILE<ENTER>

In this example the textfile "TESTFILE/SCW" will be saved on the diskette mounted in Drive Ø, and "TESTFILE SAVED" will be displayed. Press <BREAK> to return to the TEXT mode.

If you wish to use a different extension such as "/BAS", type the extension after the filename.

RULE: Extensions must be preceded with a "." or a "/".

EXAMPLE: DSTESTFILE/BAS <ENTER>

In this example, the textfile "TESTFILE/BAS" will be saved on the diskette mounted in Drive Ø.

If you wish to save your textfile to a diskette mounted in a drive other than Drive Ø, type ":" followed by the number of the drive on which you wish your file to be saved.

RULE: Drive numbers must be preceded by a ":".

EXAMPLE 1: DSTESTFILE:1<ENTER>

EXAMPLE 2: DSTESTFILE/BAS:2<ENTER>

In example 1, the file TESTFILE/SCW will be saved on the diskette mounted in Drive 1. In example 2, the file TESTFILE/BAS will be saved to the diskette in Drive 2.

USING AND CHANGING DISKNAME DEFAULTS

After you have saved a textfile to disk, the filename, extension, and drive number are stored in a buffer and are retained for later use. To display the filename press <CLEAR>-<CLEAR> to enter the COMMAND mode, then type <D>-<N> <ENTER>. The last name, extension and drive number will be displayed on the COMMAND line.

Because the filename is retained, you can save the same textfile without entering the filename, extension, or drive number each time. To save a textfile using the existing filename type <CLEAR>-<CLEAR>-<D>-<S> and press <ENTER> <Y>. Your file will be saved to the diskette mounted in Drive Ø with the previous filename, here "TESTFILE/SCW".

To save a textfile using the existing filename to a diskette mounted in a drive other than Drive Ø, type <CLEAR>-<CLEAR>-<D>-<S>-<:> then the Drive number, for example, <2>, and press <ENTER> <Y>. Your file will be saved to the diskette mounted in Drive 2 with the previous filename, here "TESTFILE/SCW".

You may change the default diskname, extension and/or drive from the DISKNAME command by entering the new name after typing <D-N>, and then pressing <ENTER>. This will change the filename for automatic diskfile command execution.

4-I DETERMINING FREE SPACE ON THE DISKETTE

To determine whether or not there is enough free space remaining on a diskette to save your file, determine the granule allocation figure for your textfile by pressing <CLEAR>-<;> to display memory (see Section 2-J), and compare it with the number of granules remaining on your disk as shown when you call up the directory (see Section 4-G).

4-J LOADING TEXTFILES FROM THE DISKETTE

The Super "Color" Writer II allows you to load or append, and edit any ASCII (not binary) file from diskette, including Super "Color" Library files, BASIC programs saved in ASCII (consult your BASIC manual), editor/assembler source files, etc.

To load any textfile saved in ASCII press <CLEAR>-<CLEAR> to enter the COMMAND mode then type <D>-<L> followed by the filename and drive number, then press <ENTER>. If no extension is specified the default extension will be used. When loaded the message "(FILENAME) LOADED" will be displayed on the COMMAND line. Pressing <BREAK> will exit to the TEXT mode.

EXAMPLE: DLTESTFILE/SCW:Ø<ENTER>

The Super "Color" Writer II will look for the file TESTFILE/SCW on Drive Ø. If the file is found it will be loaded from the diskette. A warning will sound and the

"(FILENAME) NOT FOUND" message will appear on the COMMAND line if no such file exists.

As mentioned earlier, the filename, extension and drive number are retained, allowing you to load the previously entered textfile by typing <D>-<L> and pressing <ENTER>.

EXAMPLE 1: DLTESTFILE/BAS:2<<ENTER>

EXAMPLE 2: DL<ENTER>

Example 1 will load the file TESTFILE/BAS from the diskette mounted in Drive 2. Example 2 will load the previously entered textfile, in this case TESTFILE/BAS, from the diskette mounted in Drive 2.

4-K APPENDING TEXTFILES

The system will allow you to append as many textfiles as will fit in the text buffer. To append one textfile after another, load the first, then load the second. When appending one textfile to another, if the file to be appended is too large, the system will accept all that the buffer will hold, and a "FULL" message will be displayed on the COMMAND line. If this happens, you may have to break your textfile to be loaded down into smaller pieces and save each smaller file separately.

4-L RENAMING DISK FILES

The system will allow you to rename your disk files without having to resave your files. All file name rules discussed above apply concerning filenames and extensions. To rename your file enter the COMMAND mode, type <D>-<R> plus the file name you want changed and then press <ENTER>. After you have pressed <ENTER> the system will respond with the prompt: "NEW FILE NAME?" You may then type in the new file name and press <ENTER>. The system will then rename your file.

EXAMPLE: DRTESTFILE <ENTER> NEWNAME <ENTER>

The above example renames the file "TESTFILE/SCW" TO "NEWNAME/SCW."

4-M KILLING TEXTFILES

To kill a textfile from the diskette, press <CLEAR>-<CLEAR> to enter the COMMAND mode, then type <D>-<K> followed by the filename, extension and drive number, and press <ENTER>. The system will respond with "KILL

(FILENAME) Y/N?" A <Y> response will kill the file. Any other response will exit to the COMMAND mode.

EXAMPLE: DKTESTFILE/SCW:0 <ENTER> Y

This example will kill the file TESTFILE/SCW from the diskette mounted in Drive 0.

As with all other commands, "DK" will kill the previously entered textfile if you type <D>-<K>-<ENTER> and answer the "KILL (FILENAME) Y/N?" prompt with a "Y".

EXAMPLE: DK<ENTER> Y

This example will kill the previously entered textfile TESTFILE/SCW from the diskette mounted in Drive 0.

4-N TRANSFERRING TAPE FILES TO THE DISKETTE

The Super "Color" Writer II supports cassette I/O. This allows you to load any textfile from tape and transfer it to the disk, or vice versa. See Section 4-B Loading Textfiles From Tape.

4-O LINKING DISKETTE FILES FOR CONTINUOUS PRINTING

Disk file LINKING is discussed in Section 8-C in the chapter devoted to PROGRAMMABLE FUNCTIONS.

SECTION 5 ADVANCED FEATURES - TEXT FORMATTING

5-A ADVANCED FEATURES

The Super "Color" Writer II has many special features to aid you in producing professional looking results. The advanced features in this section are aids to text formatting. That is, functions are introduced which allow you to control arrangement of elements of text, such as centering, flush right, non-breakable spaces, and the likes of headers, etc. All of these features and many others introduced later make use of FUNCTION MARKERS which are discussed in the first section below.

5-B FUNCTION MARKERS

The Super "Color" Writer II uses function markers such as BLOCK markers to set aside different types of text within the textfile or to represent special characters that couldn't ordinarily be placed in the textfile. Many of the markers are discussed immediately below in their respective sections. A complete list of the markers used can be found in Appendix D.

RULE: Only NONSPACE (see Section 5-G), PROGRAMMABLE CHARACTER CODES, PRINTER DRIVER CODES, P CODE (see Sections 7-I and 7-J) and PAUSE PRINT markers (see Section 7-L) may be placed within a line of text if it is to be printed properly. All others MUST be placed at the beginning of a new line (the previous line being terminated by a carriage return unless they are being placed at the beginning of the file).

IMPORTANT

THE FOLLOWING SECTIONS discuss and give examples of the use of FUNCTION MARKERS. To facilitate the printing of this Tutorial the examples given to illustrate the proper use of FUNCTION MARKERS use underlined printable characters (e.g., n,/,<,u,>,k) in place of the actual FUNCTION MARKERS described for each function (the underline indicates that the character underlined is the function key and must be preceded by pressing <CLEAR>). This is because FUNCTION MARKERS are either non-printable characters, or are characters which would command the printer to perform the function itself while printing the Tutorial. Therefore, when reading examples PLEASE mentally substitute the FUNCTION MARKER as DESCRIBED in the text for the printable character used in its place.

5-C CENTERING <CLEAR>-<U>

The Super "Color" Writer II takes the hassle out of centering text on the printed page. Before setting up text for centering, you first MUST establish the margins to be used for the centered text if the text to be centered will differ in line length or in print font from that used so far. (See Section 7-C and Appendix C below.) Centering is accomplished with a CENTER marker. The CENTER marker is represented on the video display as a an inverse "u" (a BLUE block in the 32 display), and is generated by pressing <CLEAR>-<U>. To set aside a line of text to be centered, press <CLEAR>-<U> at the beginning of a new line (the previous line must have been terminated by a carriage return), then type the line to be centered followed by an <ENTER>. The system will do the rest. If P CODE or other markers, described in Section 7-J below, are being used, the CENTER marker must come before the first P CODE pair or marker.

EXAMPLE 1: µThis is centered text.<ENTER>
EXAMPLE 2: µØXØThis is emphasized, centered text.ØYØ<ENTER>
Ø = P CODE marker
µ = CENTER marker.
X = Control code to turn on emphasize.
Y = Control code to turn off emphasize.

5-D FLUSH-RIGHT <CLEAR>-"<"

While formatting text, there are often times when you might want to have a string of characters flush to the right side of the page, such as when typing a letter with a date. Word processor designers have had to develop an easy means to place text FLUSH-RIGHT, aside from the obvious one of counting spaces.

The Super "Color" Writer II uses a FLUSH-RIGHT marker to accomplish this task. The FLUSH-RIGHT marker alternates with the AUXILIARY LINE A command key sequence (see Section 5-E): <CLEAR>-"<". The default marker for this key sequence is the AUXILIARY LINE A marker. To toggle the marker to be generated with this key sequence, press <A>-<ENTER> from the command mode. Then, to generate the FLUSH-RIGHT marker press <CLEAR>-"<". (This was necessary because of the small keyboard on the Color Computer.) The FLUSH-RIGHT marker is represented on the screen by an inverse "f" (a RED diagonal block in the 32 display). FLUSH-RIGET markers must be placed at the beginning of a new line (the previous line must have been terminated by a carriage return) directly before the string of text to be flushed to the right and must be terminated with an <ENTER>. You need not manually count and generate the spaces. The system will do the rest. If PROGRAMMABLE CHARACTER CODES or

P CODE markers, described in Sections 7-I and 7-J below, are being used, the FLUSH-RIGHT marker must come before the PROGRAMMABLE PRINTER CODE or P CODE pair (see centering example 2 above).

EXAMPLE ≤This text will be FLUSHed RIGHT.
≤ = FLUSH-RIGHT marker.

5-E HEADERS, FOOTERS AND AUXILIARY LINE MARKERS <CLEAR>-<,>,<.>,"<" AND ">"

HEADERS, FOOTERS and AUXILIARY lines are lines of text used for various purposes. HEADERS usually are lines of text which you would want printed at the top of every page, such as the chapter name of a book or the origin of a business letter. FOOTERS are the same, except that they usually go at the bottom of the page. They can be used for footnotes and the like. AUXILIARY lines are additional lines available to you for the same purposes if the HEADER and FOOTER are not enough.

The Super "Color" Writer II allows HEADERs, FOOTERs and AUXILIARY lines within the textfile. These lines are set aside from other text with HEADER, FOOTER and two AUXILIARY LINE markers. The HEADER marker is represented as an inverse "," (a YELLOW block in the 32 display), and is generated by pressing <CLEAR>-<,>. The FOOTER marker is represented as an inverse "." (a RED block in the 32 display), and is generated by pressing <CLEAR>-<.>. The AUXILIARY marker alternates with the FLUSH RIGHT and LINK marker command key sequence (see Sections 5-D and 8-C): <CLEAR>-"<" and ">". The default markers for this key sequence are the AUXILIARY LINE A and B markers. To toggle the marker to be generated with this key sequence, press <A>-<ENTER> from the command mode. The AUXILIARY A marker is represented by an inverse "<" (a YELLOW block absent its lower right quarter in the 32 display), and is generated by pressing <CLEAR>-"<". The AUXILIARY B marker is represented by an inverse ">" (a RED block absent its lower right quarter in the 32 display), and is generated by pressing <CLEAR>-">". Headers, footers and auxiliary lines can be as long as you wish, can be centered and can be set to begin on any line on the page.

To use HEADER lines proceed as follows: Go to the beginning of a new line (the previous line must be terminated by a carriage return unless the line begins the file) and press <CLEAR>-<,>. Next, type the text which you wish to appear as the HEADER. Terminate the line with an <ENTER>. The HEADER will be printed on line four of the printed page. You can, however, change the line on which the HEADER will be printed by setting the "HL" parameter

(see Section 7-C) to the desired line. To suppress the header, set "HL" to zero. Unless otherwise specified, when not suppressed, the HEADER will be printed on every page. The system also permits you to specify that the HEADER be printed only on odd or even pages. To specify Odd, set the HS (HEADER STATUS) parameter to O; to specify Even, set it to E; to return to printing the HEADER on All pages, set "HS" to A.

EXAMPLE: ⊥This is a header.<ENTER>
⊥ = HEADER marker.

To use FOOTER lines proceed as follows: Go to the beginning of a new line (the previous line must have been terminated by a carriage return) and press <CLEAR>-<.>. Now type the text which you wish to appear as the FOOTER. Terminate the line with an <ENTER>. The FOOTER will be printed on line sixty-two of each printed page. You can also change the line on which the footer will be printed by setting the "HL" parameter to the desired line (see Section 7-C). To suppress the footer, set "FL" to zero. The FOOTER STATUS "FS" parameters A, E, and O govern printing FOOTERs on All, Even, or Odd pages, with the default being All as with HEADERS.

EXAMPLE: ⊥This is a footer.<ENTER>
⊥ = FOOTER marker.

Creation of AUXILIARY LINES A and B follow the procedures above for HEADERs and FOOTERs. Go to the beginning of a new line (the previous line must have been terminated by a carriage return) and press <CLEAR>-"<" for an AUXILIARY A marker, type the text and press <ENTER>. Press <CLEAR>-">" for an AUXILIARY B marker, type the text and press <ENTER>. The line number on which the auxiliaries are to appear MUST be specified by setting the parameters, "AL" for AUXILIARY line A and "BL" for AUXILIARY line B, to their appropriate values. The default is zero to suppress them. The AUXILIARY STATUS parameters "AS" for AUXILIARY status A and "BS" for AUXILIARY status B must be set at <A>ll, <E>ven or <O>dd in a FORMAT line.

EXAMPLE: kAL56,BL56,ASo,BSe,HL4,FL61,HSa,FSa
k = FORMAT marker

The above example illustrates how HEADER, FOOTER and AUXILIARY LINE parameters are set in a FORMAT line. AUXILIARY A will be printed on line 56 of the ODD numbered pages. AUXILIARY B will be printed on line 56 of the EVEN numbered pages. The HEADER will be printed on line 4 of ALL pages and the FOOTER will be on line 61 of ALL pages.

Headers, footers and auxiliary lines can be centered by placing a CENTER marker directly AFTER the HEADER, FOOTER or AUXILIARY line A or B marker, but before the line of text.

EXAMPLE: ↑uThis is a centered header.<ENTER>
↑ = HEADER marker.
u = CENTER marker.

As the system prints the textfile, when it comes to the line on which the HEADER, FOOTER or AUXILIARY line is to be printed, it will print the most recent HEADER, FOOTER or AUXILIARY line that was encountered. This allows you to change HEADERS, FOOTERS or AUXILIARY lines by placing additional HEADER, FOOTER or AUXILIARY lines elsewhere within the textfile.

If you suppress headers, footers or auxiliary lines, they remain suppressed until you set the "HL", "FL", "AL" or "BL" parameter to a value other than zero (Ø).

RULE: HEADERs, FOOTERs and AUXILIARYs MUST be placed before any printable text in order to appear on the page on which they are entered. "HL", "FL", "AL" and "BL" cannot be set to appear on the same line if they are to be printed on the same page. Use the WINDOW (see Section 6) if you are not sure these lines are properly placed.

5-F COMMENT MARKERS <CLEAR>-<+>

The Super "Color" Writer II permits you to make a commentary to what you have written by putting non-printable COMMENTS in the text. Such a commentary can often be helpful if you are writing a document which will be used several times and you want to remember why you put some piece of information in, or you want to remind yourself of other alternatives, etc. For example, you may wish to create a file of comments to remind you of the keys to press to implement your printer driver. You could then load that before beginning work on a new file so that you could check it when you needed a reminder. COMMENT markers must be placed at the beginning of the line directly before the text to comprise your comment (the previous line must have been terminated by a carriage return), and the COMMENT must be terminated by an <ENTER>. The COMMENT marker, an inverse "+" (a GREEN diagonal block in the 32 display), is generated by pressing <CLEAR>-<+>. Your COMMENT may be any length. When the printer comes to a COMMENT marker, it ignores everything that comes after it up to and including the first <ENTER>.

EXAMPLE: ±This is a comment.<ENTER>
± = COMMENT marker.

5-G NON-BREAKABLE SPACE <CLEAR>-<SPACE>

The NONSPACE marker is represented as an inverse space (a BLOCK of a dark hue of the foreground color selected by you when using this program, i.e., green or orange in the 32 display), and is generated by pressing <CLEAR>-<SPACE>. NONSPACE markers can be very useful, especially for tying words together when using the justification feature.

The NON-BREAKABLE SPACE marker is just like a space, in that the system treats it as a space, but it differs from the space in one important respect: spaces are "breakable." You've noticed that when characters are typed to form a word, the system will not split them apart when wrapping words around on the screen or when wrapping words around for printing. The system waits until it encounters a space to split up the text for wrap around. Spaces also are an area in which the system can insert more spaces to space words out during justification.

NON-BREAKABLE SPACES, on the other hand, are binding spaces. They bind the text together just like normal letters so that the text cannot be split where the space would be or so that spaces cannot be inserted during justification. NONSPACES are thus used to bind words together with only one or a specified number of spaces in between. Words joined by NONSPACE markers are treated as one long word for word wrap around and justification purposes.

WORD SPACING WITHIN JUSTIFIED TEXT

Justified text provides a handsome printed product with even and neat margins on both the left and right sides of the page. Justification does, however, occasionally cause problems since it works by inserting spaces next to other spaces in a line to extend the line to make the margins even. Sometimes, however, you do not want spaces inserted between some text. For example, you rarely would want the units of a proper name to have extra spaces inserted between them like this: John J. Doe; and you usually want to ensure that there are exactly two spaces after a colon (unlike the above example). To cure this problem (which only occurs when you are using the justification feature) you must use NONSPACES to bind the text together.

EXAMPLE:__John_Q._Smith<ENTER>
_ = NONSPACE marker.

(Since a space is not a visible character, it is shown as the underline character alone representing <CLEAR>-<SPACE>).

5-H FORMFEEDS <CLEAR>-</>

The FORMFEED marker is represented by an inverse "/" (a CYAN block in the 32 display), and is generated by pressing <CLEAR>-</>. The FORMFEED marker must be placed at the beginning of a new line (the previous line must have been terminated by a carriage return). (Formfeed characters intrinsically act as carriage returns in the display, i.e., they are the only character on the line. This makes them easier to find.) Formfeeds are commands to the printer to skip lines until it reaches the top of the next page for printing. Formfeeds are used in two situations:

1) Where you wish to print a partial page and automatically feed the paper to the top of the next page. The FORMFEED should be placed at the beginning of the line following the point at which you wish that page to end. This point could be determined using the WINDOW MODE (see Section 6).

2) FORMFEEDS are also used where the last printed page of a multiple-paged document is not a full page and you want the page number and/or footer (if located at the bottom) to be printed. Normally, printing is stopped at the last character in the file. Footers and page numbers are not technically part of the file, but are added automatically by the system from a special buffer. Without a FORMFEED at the end of the text, the page number and footer, which are placed at the bottom of the page by the system, are not printed there since the system has not been allowed to get to where they were to be printed. Again, you will see this if you use the FORMAT WINDOW for final formatting. The FORMFEED should be placed at the beginning of the line immediately following the end of text on the page.

SECTION 6 WINDOW MODES

SECTION 6-A THE EDIT WINDOW

The edit window was introduced in Section 2-D. There is little that needs to be added about how the edit window works, since its workings are pretty simple and clear. What needs a small discussion is how to easily create columns and hyphenate text using the EDIT WINDOW.

COLUMN CREATION IN THE EDIT WINDOW

Column creation is so simple, it hardly needs discussion. Of course, you first have set the WIDTH of your text to the width of your printed line. Now all you have to do is go across the first row and set up the columns. All you have to remember is to use a carriage return after each line so that no spaces are inserted if you are using justified text. Once you are satisfied with the spacing with the first row, you can then go and quickly program TABS (see Section 2-G) for the columns stops and create the remaining rows with the first row as a model.

SIMPLE HYPHENATION IN THE EDIT WINDOW

Although justified text is extremely pleasing to read, some prefer to combine justified text with hyphenation, or to use hyphenation without justification ("ragged right"). With the EDIT WINDOW, hyphenation, even within justified text, is a breeze. IT IS IMPORTANT TO NOTE, however, that if you are using printer codes of any type (see Sections 7-I thru 7-J) which take up line space in the TEXT mode, your hyphenation will be thrown off by these codes unless they are stripped from the file by you prior to hyphenation. If you cannot strip these codes, you will HAVE TO use the FORMAT WINDOW, discussed below, to determine where to hyphenate your text, as it does strip such codes.

The first step is to create a WIDTH the same as your printed line width so that your text will be displayed with your print margins, no matter the screen display you are using. Now, when you view your lines you will see exactly what the last word of the line will be and how many spaces are left from the last word to the true end of the line. It is those spaces which you have to fill with a hyphenated portion of the word beginning the next line. The aim is to split the first word in the next line at a syllable break (adding a hyphen and a space) so that the word will rewrap up to the previous line and use the extra spaces.

Of course, if the word beginning the next line is a single syllable word it cannot be split so you cannot use it to fix the previous line. You will have to either insert

spaces in the previous line if you are not using justification or you may leave the space insertion to the automatic justification feature. If the first word of the next line is multi-syllabic, you may break the word at the syllable break which comprises the number closest to the number of spaces left on the line minus one space for the hyphen you will add. Split the word, add the hyphen and a space, and the system will rewrap the word to the previous line. Do this for all the lines of your text. If you want justified text, the system will insert any remaining spaces in the line to give you an even right-hand margin. Otherwise you will have created a "ragged right" margin.

6-B THE FORMAT WINDOW

The Super "Color" Writer II offers hi-res displays with 51, 64 and 85 characters per line and the WIDTH command to set the display to the line length which will be printed (see Section 2-D). These modes, without more, will display your normal text as a close facsimile of what will be printed. With use of the EDIT WINDOW you can view a reasonable facsimile when creating text with a line length greater than 85 characters per line, such as when creating special reports or diagrams using the compressed mode or the 15 pitch option of your printer to print between 120 and 132 characters per eight-inch line on the standard 8 1/2 by 11 inch page. With the EDIT WINDOW you can even view special formats for larger paper sizes such as 11 by 14 inch paper, or even larger, which allow more characters per line.

Still, the hi-res displays and EDIT WINDOW do not solve one essential hindrance to replicating the text EXACTLY as it will be printed. The Super "Color" Writer II allows you to freely enter and edit printer control codes, printer parameter FORMAT lines, centered text, HEADERS and other special features to be used for printing when entering text. These features are dynamic. That is, you indicate by use of a MARKER the function to be performed when printing and the system or printer automatically implements the function WHEN YOU PRINT THE TEXT.

Until the text is printed, however, the FORMAT lines, printer codes and MARKERS, which will not be printed, remain in the EDIT WINDOW display. This characteristic prohibits EXACT replication in the TEXT mode of what you are writing in its final printed form, even with the hi-res displays using the WIDTH command, since the FORMAT lines and printer control codes necessary to tell you what you will be having your printer do take up text space on the screen.

We were sensitive to this need for replication of the final formatted text, so we included a formatted display WINDOW in the Super "Color" Writer II. This FORMAT WINDOW

displays the text the way it will look when printed, but you get to see it BEFORE it is committed to hard copy. Your text will be "printed" to the screen. All functions controlled by FORMAT parameters and function markers are implemented and masked out of the display. BLOCK markers and improperly used markers are not, however, masked, and the FORMAT WINDOW may not be used with such markers in the file. Printer functions implemented by printer control codes and character codes are not implemented since they are created by the printer, not by the Color Computer; they are, however, masked from the display.

Thus, if the text to be displayed is to be centered, it will be displayed, centered, in its proper position on the line (normal print fonts ONLY). If a header is to be displayed on the fourth line from the top, that is where you will find it. And so with page numbers, indented quotes, auxiliary lines, justification, flush right, footers and any other special formats you have selected. You can check to see that your columns are properly spaced and that there are no orphan lines. The WINDOW does NOT, however, display printer-controlled format features such as double-wide, italics, emphasized or underlining the way they will appear since the screen display cannot represent them. In other words, if the text to be centered is in the double-wide (elongated) print font with a left margin of zero and a right margin of 40, the text to be centered will be displayed offset from the rest of your text and will not appear as it will be printed.

SECTION 6-C USING THE FORMAT WINDOW MODE <CLEAR>-<P>-<W>

To enter the WINDOW mode first move the cursor to the portion of the text to be displayed and then press <CLEAR>-<P> to initiate printing. The system will ask whether you want to PRINT or use the FORMAT WINDOW. Press <W> to enter the WINDOW. When you wish to exit to the TEXT mode for editing, press <E>. The cursor will be at the same position from which you entered the WINDOW mode.

When entering the FORMAT WINDOW, you will notice a slight delay before the screen is updated. This is normal. The COMMAND line will indicate that you are in the WINDOW mode. The screen display will vary with the display mode. You will see either the first 32 characters of 14 lines of text or the first 51, 64, or 85 characters (provided your line is that long) of 19 lines of text (or 22 in the 24 lines/page option), beginning from the left margin, of the page of formatted text on which the cursor was positioned. Carriage returns will not appear on the screen unless you want them to be there. To display them, while in the WINDOW mode, press <ENTER>. To get rid of them, press <ENTER> again. Note that when you select carriage returns, they

will appear at the end of every line representing the carriage returns sent to the printer.

A STATUS LINE at the bottom of the screen indicates your current position in the text. The STATUS LINE provides you with the following information from left to right:. PN (Page Number), LN (Line Number of the top line on the screen), CN (Column number in number of columns from number one of the left margin to the leftside of the display), LM (the current Left Margin), RM (the current Right Margin).

NOTE: If you don't define the page number (PN) parameter at the beginning of your textfile, the window will assign the number "1" to the first "page" of your textfile. If you have changed page numbers or interrupted the numbering sequence anywhere in the textfile, when the system encounters it in the WINDOW mode, it will try to use the original number sequence it assigned your text and WILL NOT function properly.

If a format error, such as incorrect margins, page length, footer placement, etc. (see Section 7-B) occurs while in the WINDOW mode, an error message diagnosing the problem will be displayed on the COMMAND line and the cursor will indicate the position of the error in the FORMAT LINE (see Section 7-B). Correct the error and press <CLEAR>-<P>-<W> to continue. You will find this is a great check on your page formatting.

SECTION 6-D SCROLLING IN THE FORMAT WINDOW

Once you have entered the FORMAT WINDOW mode, you may exit to the TEXT mode by pressing <E>. The following keys or combinations of keys perform the functions indicated in the FORMAT WINDOW:

<UP ARROW>.....Scrolls up 12/17/20 lines (from mid-page).
Or scrolls to top of previous page (from top of page).
<DOWN ARROW>...Scrolls down in the text file 12/17/20 lines.
<LEFT ARROW>....Scrolls left in the text file 16 characters.
<RIGHT ARROW>..Scrolls right in the text file 16 characters.
<SHIFT><UP ARROW>.....Displays the first page.
<SHIFT><DOWN ARROW>.....Displays the next page.
<SHIFT><LEFT ARROW>.....Displays left side of current page.
<ENTER>.....Toggles the display of carriage returns.

All FORMAT WINDOW functions repeat if the keys are held.

SECTION 7 PRINTING AND PAGE FORMATTING

7-A PRINTING WITH THE SUPER COLOR WRITER II

Up until now you have been primarily concerned with putting text in the buffer and editing your text. The end goal, however, has always been to print your work using your printer. Before you print your text you generally need to prepare it to meet your needs. Preparation of your text can be divided into two segments: system oriented page formatting functions (FORMAT PARAMETERS) and printer oriented functions (PROGRAMMABLE CHARACTER CODES, PRINTER DRIVERS and PRINTER CONTROL CODES). Before we go into print preparation in depth, however, you should be aware of the fundamental PRINT mode commands.

FUNDAMENTAL PRINT COMMANDS

The Super "Color" Writer II will automatically print your text PERFECTLY CENTERED on 8 1/2 by 11 inch pages of continuous-feed paper, and has a SINGLE SHEET PAUSE option to allow the use of single sheets of paper as well (see Section 7-C). THE MARGINS AND PRINT LENGTH ARE PRESET FOR PERFECTLY FORMATTED TEXT. For properly printed text, move the cursor to the top of the textfile, move your paper in your printer to top of form (the system will give you your top margin) and press <CLEAR>-<P>. The system will then ask if you wish to PRINT or use the FORMAT WINDOW. Press <P> to PRINT, <W> for the FORMAT WINDOW, or any other key to abort. Everything will then be printed except COMMENTS, see Section 5-F, or text enclosed by NO-PRINT markers, see Section 7-M. You can even print those if you use the PRINT COMMENT and NO-PRINT PRINT features (see Section 7-C). You may print any lesser portion of text below the cursor position by pressing <CLEAR>-<P>-<P> after you have positioned the cursor.

Holding the <SPACE BAR> will pause printing. Printing can be resumed by pressing any key. Holding down <BREAK> will stop printing, and exit to the TEXT mode. You may use the INVISIBLE PRINT feature (see Section 7-C) to recommence printing on the page from which you broke away from printing.

7-B PAGE FORMATTING

INTRODUCTION

The Super "Color" Writer II system controls the various elements of page formatting such as margins, line spacing, pagination, headers, justification and page length. These formatting elements are called PRINT FORMAT PARAMETERS. These PARAMETERS have been preset with default values to

allow the novice a perfectly printed page without even having to consider the FORMAT PARAMETERS (see Section 7-A). Still, the more experienced user will periodically desire to change these default values to implement more sophisticated formatting, and the system also allows for this.

These parameters are listed below in Section 7-C along with the default values initially defined by the Super "Color" Writer II. This is your great chance to use the WINDOW mode. The WINDOW will help you spot formatting errors and allow you to make better formatting decisions.

ELEMENTS OF PAGE FORMATTING

CHANGING MARGINS

Before entering into a discussion of the method for altering default values, it would be helpful for you to understand a little bit about the reasons the original default values for the left and right margins were chosen. Printers offer several different options for printing text. Most SMART printers offer several print fonts to allow different means of emphasis. To do this, the printer uses a different amount of space to print the same character, and this affects how you format your text.

The normal 8 1/2 by 11 inch sheet of paper allows the printer eight inches of usable space per line. When in the NORMAL print font, the printer prints ten characters per inch (cpi). With the total line length of eight inches, the printer is thus capable of printing 80 characters per line. The sizes of the other print fonts alter these figures. In the ELONGATED (DOUBLE-WIDE) font the printer can type 5 cpi, thus giving a total of 40 characters per eight inch line. In the ELONGATED-COMPRESSED font the printer is capable of printing 8.3 cpi, for a total of 66.4 (67) characters per eight inch line. The twelve pitch option is a 12 cpi format, for a total of 96 characters per eight inch line. The fifteen pitch option is a 15 cpi format, for a total of 120 characters per eight inch line. Finally, in the COMPRESSED font, printers can handle 16.7 cpi, for a total of 132 characters per eight inch line.

Since you are usually in the NORMAL font, we had to choose the proper default margins to use within the 80 character per line maximum. The common practice is to have a 64 character line, leaving an eight character space to the left and right of the text for a margin. Thus, we have set the left and right margins to 8 and 72 (8-64=72; 72+8=80).

If you want to print CENTERED or FLUSH RIGHT text using other fonts, or if you desire a different line length, the margin for the NORMAL font is not always the proper one. Since the new line length is up to you, we will let you figure out the exact margins you require.

Whenever you wish to alter the print font (see Sections 7-I and 7-J dealing with printer drivers and P CODEs) which is often the case with centered text which is often in a different print font, you must first alter the default left and right margins to reflect the number of characters per line that the printer can print in that font. Thus, when you want to switch from NORMAL to ELONGATED and then back to NORMAL, the default values must be changed both before going from NORMAL into ELONGATED and again before going back into NORMAL. The proper margins to be used for CENTERING using each of the standard print fonts are as follows:

```
ELONGATED.................(5 C.P.I.).....................LM0,RM40
ELONGATED-COMPRESSED.....(8.3 C.P.I.).................LM0,RM67
NORMAL....................(10 C.P.I.).....................LM0,RM80
TWELVE PITCH..............(12 C.P.I.).....................LM0,RM96
FIFTEEN PITCH.............(15 C.P.I.).....................LM0,RM120
COMPRESSED................(16.7 C.P.I.).....................LM0,RM132
```

CHANGING DEFAULT PARAMETERS

There are several important things which must be known about FORMAT PARAMETERS. Every time the system is started up the FORMAT PARAMETERS contain their default values. The current value of all PARAMETERS will be displayed if you press <.>-<ENTER> from the COMMAND mode. All FORMAT PARAMETERS will stay with the system until actually CHANGED. Merely turning off a PARAMETER will not change it. Thus, if you have altered the FORMAT PARAMETERS in a textfile, printed that textfile, and then cleared the buffer and started a new textfile, unless you change the parameters as they existed in the textfile that you just printed, the FORMAT PARAMETERS will remain as they were set in the printed textfile until changed or until the system is turned off. This could cause problems since you may not remember your parameters, such as when you had turned justification off for a particular letter, and the effects may show up only when you use the FORMAT WINDOW mode or worse yet when you print your textfile. This is another reason to USE THE FORMAT WINDOW.

Alteration of the default values is accomplished in either of two ways:

1) You may at any time alter any parameter by going into the COMMAND mode and entering the parameter syntax (listed below in Section 7-C). This is a very quick and convenient way to change any FORMAT PARAMETER. It does, however, have one major drawback: if FORMAT PARAMETERS are changed in this manner, they do not become part of the textfile buffer, and therefore are NOT saved when you save the textfile to tape or disk.

EXAMPLE: <CLEAR>-<CLEAR> <L>-<M>-<2> will change the default left margin PARAMETER, or the previous left margin, to a new left margin of zero.

2) Parameters may also be changed by entering a FORMAT line. A FORMAT line is composed of a FORMAT marker, which is represented as an inverse "k" (a MAGENTA block in the 32 display), followed by FORMAT parameters and terminated by an <ENTER>. FORMAT lines can be placed anywhere within the textfile to change margins, line spacing, justification, etc. FORMAT lines are not printed in the text.

FORMAT lines are created by generating a FORMAT marker at the beginning of a new line (the previous line must have been terminated with a carriage return). To generate a FORMAT marker, press <CLEAR>-<K>. Next type the parameter desired using the syntax listed in the table below. Then type the desired value to be assigned (within the limits listed under the same table heading). Many parameters can be typed on a single line, but each parameter MUST be separated by a comma; there is, however, no required order for entering FORMAT parameters. The FORMAT line MUST be terminated with an <ENTER>.

The following example is a FORMAT line which sets the LEFT MARGIN to 20, the RIGHT MARGIN to 60, the LINE SPACING to 2, the TOP MARGIN to 10, the BOTTOM MARGIN to 62, JUSTIFY off, and SINGLE SHEET PAUSE on.

EXAMPLE: kLM20,RM60,LS2,TM10,BM62,JUn,PAy<ENTER>
k = FORMAT marker.

The above example of a FORMAT line is the first of many examples to follow that use a "k" to represent a FORMAT marker. Remember that the FORMAT marker is generated by pressing <CLEAR>-<K> and is represented on the video display as an inverse "k" (a magenta block in the 32 display.

The syntax of the FORMAT parameters, and the <Y>, <N>, <A>, <E>, <O>, <L>, <C>, <R> and <T> electives can be typed in either upper or lower case. FORMAT lines can be block moved and copied. FORMAT lines will not be printed. When the textfile is saved to tape or disk, the FORMAT lines will be saved as well, so the print format you've chosen will not be lost.

NOTE: While printing or while entering the WINDOW mode, if the system encounters a FORMAT parameter that is beyond the limits listed below, or the FORMAT line has been incorrectly entered such as by not separating PARAMETERS with commas, one of several error messages will be displayed on the COMMAND line, the error detect siglal will sound and the system will return to the TEXT mode. The cursor will indicate the location of the error. The error message displayed will diagnose the error encountered, such as

"TM>BM", "BM>PL" and "BD NOT 1-8". Try again after making the necessary correction.

When a FORMAT error is encountered in the middle of a page, printing must be resumed from the top of the page on which the error occurred because the system line counter is always reset to one when you press <CLEAR>-<P>-<P> to print. To resume printing, first correct the error encountered. Next go to the top of the file and turn the INVISIBLE PRINT parameter on (in a FORMAT line), then locate the first word printed at the top of the page on which the error occurred and turn the INVISIBLE PRINT parameter off. Finally, go to the top of the file and commence printing. The text up to where INVISIBLE PRINT is turned off will be processed without being sent to the printer, then the text will be sent to the printer with the correct page number, etc.

RULE: When you intend to rely on the DEFAULT value of a parameter for most of your textfile, but need to use a different value for a special purpose such as setting the bottom margin to a different length for just one page of your file, YOU MUST SET THE DEFAULT VALUE IN A FORMAT LINE PRIOR TO SETTING THE NEW VALUE IN THE FILE. If you don't do this, the system will apply the newly programmed value to the whole textfile, for example, applying your special bottom margin to all pages in your file. Once programmed, the new parameter will control until reprogrammed.

7-C FORMAT PARAMETERS

The following is a list of the FORMAT parameters, all of which can be changed. The status of all parameters will be displayed by pressing <CLEAR>-<CLEAR> to enter the COMMAND mode, then typing <.>-<ENTER>, or the syntax of any parameter followed by <ENTER>.

EXAMPLE: <CLEAR>-<CLEAR>BD<ENTER>

The current status of all parameters will be displayed on the screen.

Parameters can also be changed if, instead of pressing <ENTER> after entering the syntax, you type the substitute parameter.

EXAMPLE: <CLEAR>-<CLEAR>BD5<ENTER>

This example changes the printer baud rate to 2400 baud.

ALL PARAMETERS CAN BE RESET TO THEIR DEFAULT VALUES EITHER BY PUTTING AN "@" IN A FORMAT LINE, OR BY PRESSING <@>-<ENTER> FROM THE COMMAND MODE.

PARAMETER	SYNTAX	DEFAULT VALUE	LIMITS
AUXILIARY A LINE	AL	0	1-255 0=SUPPRESS
AUXILIARY B LINE	BL	0	1-255 0=SUPPRESS
AUXILIARY A STATUS	AS	A	A, E, or O
AUXILIARY B STATUS	BS	A	A, E, or O
BAUD RATE	BD	3	1=110,2=300,3=600 4=1200,5=2400,6=4800,7=9600
FOOTER LINE	FL	62	1-255 0=SUPPRESS
FOOTER STATUS	FS	A	A, E, or O
HEADER LINE	HL	4	1-255 0=SUPPRESS
HEADER STATUS	HS	A	A, E, or O
INVISIBLE PRINT	IP	N	Y or N
JUSTIFY	JU	Y	Y or N
LINE DELAY	LD	0	1-255 0=SUPPRESS
LINEFEEDS(after CR)	LF	N	Y or N
LINE SPACING	LS	1	1-255
MARGIN BOTTOM	BM	60	2-254
MARGIN LEFT	LM	8	0-255
MARGIN RIGHT	RM	72	10-255
MARGIN TOP	TM	6	1-253
NO-PRINT PRINTING	NP	N	Y or N
NULLS	NL	0	0-255
PAGE LENGTH	PL	66	3-255
PAGE NUMBER	PN	1	1-65535
PAGE NUMBER LINE	PG	0	1-255 0=SUPPRESS
PAGE NUMBER PLACEMENT	PP	C	L C R or T
PRINT COMMENT	PC	N	Y or N
PRINTER HANDSHAKING	PH	Y	Y or N
SINGLE SHEET PAUSE	PA	N	Y or N

A = All; E = Even; O = Odd; Y = Yes; N = No
L = Left; C = Center; R = Right; T = Toggle

Some of these parameters need explaining:

The **BAUD** parameter allows you to alter the speed with which you can send your file to the printer. It is governed by the ability of your printer to handle the selected BAUD rate (see your printer manual).

The **INVISIBLE PRINT** function is used when, during printing a multi-paged document, you must stop printing midway because of format error, printer error, etc. Instead of starting printing all over, or resetting all parameters and printing from the beginning of the page on which printing stopped, you may set IP to "Y" at the beginning of the file and set IP to "N" at the beginning of the page on which printing stopped. Start printing at the beginning of the file. The system will invisibly "print" all the text up to the IPN indicator, and then send text to the printer.

The LINE DELAY parameter is provided for those whose printers require a delay after each carriage return and before the next character to be printed is sent to the printer to allow the printer time to move the carriage back to the beginning of the next line (see your printer manual). Unlike the NULL parameter, nothing is sent to the printer with this parameter. Instead, the system waits the amount of time it would take for the specified amount of NULL characters to be sent to the printer.

LINEFEEDS are required by some printers (see your manual) to advance the paper before printing the next line.

The NO-PRINT PRINTING parameter, when enabled by putting "NPY" in the format line, commands the system to print all text between NO-PRINT markers (see Section 7-M).

NULLS are empty characters required by SOME PRINTERS to ive them time to move to the beginning of the next line for printing (see your printer manual).

The PAGE NUMBER (PN), PAGE NUMBER LINE (PG), and PAGE NUMBER PLACEMENT (PP) parameters are discussed below in Section 7-D, Pagination.

The PRINT COMMENT parameter, when enabled by putting "PCY" in the format line, commands the system to print all comments along with the rest of the text (see Section 5-F).

The PRINTER HANDSHAKE feature allows you to send your text to a printer that does not require handshaking (see your printer manual) or through the RS-232 port directly to a terminal without waiting for handshaking signals which are used by printers. The default is PH on ("Y") for the normal transmitting of text to a printer.

The SINGLE SHEET PAUSE function is useful for anyone having a printer which requires or allows the user to insert single sheets of paper for printing instead of using continuous feed paper. This function is enabled by putting "PAy" in the format line. When enabled the system will stop printing at the end of each page. Printing will not be resumed until you press any key (except <BREAK>) after inserting more paper.

7-D PAGINATION

Initially, the system suppresses pagination. If you wish to print page numbers, set the PAGE NUMBER LINE (PG) parameter (within a FORMAT line) to the line on which you wish the page numbers to appear (e.g., PG62). You may specify placement of the page number (initially centered) at the left or right hand side or the center of the page by setting the PAGE NUMBER PLACEMENT (PP) parameter to "L", "R"

or "C". You may also alternate left and right sides of the page by setting the PAGE NUMBER PLACEMENT parameter to "T". Page numbers will then appear on the right side of odd numbered pages and the left side of even numbered pages.

Page numbers will always start with number 1 unless you specify another page number with the PAGE NUMBER (PN) parameter (within a FORMAT line e.g., PN7) and will be printed centered on the page on the line you specified with the PG parameter. If you wish to start with a different page number, set the PAGE NUMBER (PN) parameter to the desired number (less than 65535). If you wish to temporarily suppress page numbers, set PG to zero (Ø). Page numbers will still be incremented, but will not be printed. If you wish to re-enable page numbers, reset "PG" to the number of the line on which you want the page number printed on the page from which pagination is to resume.

EXAMPLE 1: kpn3,pg4,ppr
EXAMPLE 2: kpgØ
k = FORMAT marker

Example 1 sets the line on which the page number is to be printed to line four, specifies that the page number will be printed at the right-hand side of the page, and sets the page number to 3. Example 2 suppresses page numbers.

7-E PRINTER ORIENTED FUNCTIONS

Text formatting with FORMAT PARAMETERS allows you to fully control margins, HEADERS, FOOTERS, justification and several other page formatting features. However, there are many printer features available, depending on your printer, which are not system controlled, but are printer controlled. These printer controlled features, such as bold face, italics, compressed characters, elongated characters, superscripts, subscripts, underlining, special characters and backspacing, are implemented by transmitting printer control codes to the printer. Because these control codes are different from the ordinary letters of the alphabet, punctuation, numbers, etc, they must be "imbedded" into your textfile in special ways when you need them. This is so you can use any printer to do anything.

Printer controlled features can be divided into three categories: programmable printer codes in your printer driver; programmable character codes; and printer control code controlled features. Each of these three will be explained so you can get started making full use of your printer. Before discussing these three printer control methods, it will be useful to discuss some fundamental concepts about printers and how they interact with the Super "Color" Writer II.

7-P A PRIMER ON PRINTERS

Because the Super "Color" Writer II is so versatile, allowing it to be used with and control any printer, a discussion of how to use the program with all those printers is almost impossible to do well. This section will attempt to discuss the rudiments of using a printer to its best advantage, and using this program to make your printer shine. Since you already own a printer, you already own a printer manual. After a survey of the various printer manuals we have to conclude that you are surely lost. The manuals are a poor lot; it's a miracle that people make the printers work at all. We hope this section helps.

Printers can be divided into "impact" printers, including letter quality, and smart printers. "Impact" printers are those which usually lack "brains". They are sometimes very limited, lacking many features such as emphasized, compressed and double-strike fonts. Letter quality printers are extremely sophisticated "impact" printers which produce letter-quality hard copy and allow a change of print font by changing to a different thimble. The more expensive letter quality printers are actually smart printers, with many special features for the user. On the other hand, those with smart dot matrix printers can use many different print fonts without changing the print head.

"IMPACT" PRINTERS

As stated above, "impact" printers often only provide the bare bones. Owners of letter quality printers, however, have reason to be proud. Their printers will produce beautiful letter-quality hard copy, and because they have many different thimbles or daisy wheels at their disposal, they have a variety of print fonts and styles to choose from. Nevertheless, letter quality printers will not produce many of the fonts available to the dot matrix printers. Thus, letter quality printer owners will have different concerns than those of smart printer owners. Particularly, letter quality printer owners will be interested in the PAUSE PRINT function (see Section 7-L) which can be used to stop the printer to change the thimble or daisy wheel. The other functions will also be of use, as limited by the capabilities of the printer.

SMART DOT MATRIX PRINTERS

Dot matrix printers are getting increasingly cheap and smart. Such printers offer an astounding variety of features to allow different print fonts, several modes of emphasis, superscript and subscript, and even graphics printing. The quality of print is also getting increasingly

better, with a greater number of dots being used to represent a single character. Dot matrix printer owners will find all the print functions to be applicable.

7-G MATCHING THE SUPER "COLOR" WRITER II WITH YOUR PRINTER

THE ASCII SYSTEM

The Super "Color" Writer II, like most other good word processing programs, is totally ASCII compatible. This means that everything that you put into the buffer is in the ASCII form. ASCII is a standard for symbols used in communications, the acronym standing for American Standard Code for Information Interchange. The ASCII standard potentially contains 256 symbols which are represented by numeric values from 0 to 255 decimal (0 to FF hexadecimal).

Before going on with this discussion you should become familiar with the chart in Appendix E. This chart shows the Extended ASCII character set. The regular ASCII character set is comprised of ASCII symbols from decimal 0 through 127. The Extended ASCII character set is comprised of the standard ASCII character set plus the symbols for decimal 128 through 255. The first column gives the first 128 ASCII symbols in order; the second column states the manner in which the ASCII symbol is displayed in the 32 display when properly generated; the third columns gives the display in the hi-res screens; the fourth column gives the decimal numeric equivalent of the ASCII symbol, and column five gives the hexadecimal equivalent; the sixth column explains common functions of some of the control codes; and the final column is the MOST IMPORTANT: it is the column which explains how to generate the ASCII symbol and its numeric equivalent with the Super "Color" Writer II through your Color Computer keyboard. Thus, from Appendix E you can see that to generate the ASCII symbol "A", with the numeric value 65 decimal (41 hex) which is stored in the buffer, you press the "A" key. The screen display in all displays shows the letter "A".

The first 32 ASCII symbols, with numeric values from 0 to 31 decimal, are control characters. They were specifically devised to be used to control functions of devices which accept ASCII data. The next 96 ASCII symbols, from 32 to 127 decimal, are the alphabet, in upper and lower case form, the numbers, and the conventional symbols seen on the typewriter keyboard such as the colon, the ampersand, the dollar sign, etc. In every system adhering to ASCII the ASCII symbols represented by the numbers from 32 to 127 decimal will be the same.

The Extended ASCII symbols from decimal 128 to 255 are different. There is no standard symbolic equivalent for them. Each system (computer, printer, etc.) may use these decimal numeric equivalents to produce different symbols. It is very possible, therefore, that what will be represented on the screen and what will be printed when you generate decimal 128 to 255 will be different. You must therefore be sure to refer to your printer manual to see what you will be printing when you generate one of these numeric equivalents. In the Super "Color" Writer II the numeric equivalents from decimal 128 to 255 are generated by programming them through FORMAT lines instead of directly by pressing the key sequence found in column seven in Appendix E. For a discussion of how to generate decimal 128 to 255 see Section 7-I dealing with PROGRAMMABLE CHARACTER CODES. Because of this lack of standard ASCII symbols, printer functions and monitor display for the numeric equivalents from decimal 128 through 255, the Extended ASCII character set in Appendix E does not show anything for those decimal equivalents.

As you can see, when you press a key on the keyboard, what you are really doing is putting a number into the buffer, that number being the numeric equivalent of the ASCII symbol you have generated. It just so happens that the letters of the alphabet and other standard keyboard symbols ARE the ASCII symbols themselves. Thus, when you press the "7" key, you have generated the ASCII symbol "7", which is represented on the screen as a "7". How is "7" represented in your buffer? By looking in Appendix E you can see that the numeric equivalent for the ASCII symbol for "7" is decimal 55 (37 hex). (Actually the buffer contains the binary equivalent of the decimal number 55. For convenience we will refer to decimal numbers when referring to buffer contents.) Thus when you press "7", your buffer receives a decimal 55 in the appropriate memory location. Since "7" is itself a number, why isn't it sent to the buffer as a decimal 7? Because the "7" on your keyboard and the "7" on your screen are merely symbols, the ASCII symbol "7" to be precise, and this symbol has been assigned the numeric equivalent of decimal 55 in the ASCII system. Decimal 7, on the other hand, is not the numeric equivalent of a "7" at all, but, as you can see from Appendix E, is the numeric equivalent of the ASCII symbol Control G, which is represented on the screen by an inverse apostrophe, and is not generated by pressing "7", but by pressing <CLEAR>-<SHIFT><G>. Well then, instead of pressing the 7 key to get an ASCII 7, can you press its numeric equivalent 55? No! By pressing 55 you generate TWO bytes of data for your buffer, both being the numeric equivalent of the ASCII symbol for "5", which is decimal 53 (35 hex). Get it now? To generate any given decimal equivalent of an ASCII symbol, go to Appendix E, find the decimal column and locate the number, then go to the last column to see how that decimal

equivalent is generated. Follow the instructions in the last column to send the proper decimal value to the buffer.

You should now understand how ASCII symbols relate to their numeric equivalents, and how numeric equivalents are generated. This is IMPORTANT for proper use of your printer since you will have to generate decimal equivalents to implement many printer functions (see Sections 7-I and 7-J).

FROM BUFFER TO PRINTER

If you have your buffer chock full of text you want to have printed, your buffer will contain oodles of bytes, each containing a number from decimal Ø to 255 representing an individual ASCII symbol. Your printer is also programmed to work with the ASCII system, so when you hit <CLEAR>-<P>-<P> to commence printing, your buffer sends its numbers one by one to the printer. The printer receives the numbers and interprets them for printing.

Many of the numbers, specifically decimal 32 to 127 (the alphabet, etc.), are printed by the printer as their ASCII equivalents. Not all numbers sent to the printer, however, are intended for printing. Take the ordinary space for example. When you press the space bar, you generate the ASCII symbol for space, which ain't much. Into your buffer goes a decimal 32 (2Ø hex). When the space goes to the printer, your buffer sends the decimal 32 to the printer. The printer receives the decimal 32, and in response it skips a space during printing. Thus, the decimal equivalent for the ASCII symbol for space does not have the printer "print" anything; instead, decimal 32 commands the printer to do something, i.e., skip a space.

Other ASCII symbols, and their numeric equivalents also serve as commands to the printer. Recall that the ASCII symbols from Ø to 31 are called control symbols. These symbols were designed to be used to control functions. With few exceptions, smart printer manufacturers have therefore taken these control symbols and adopted them to control certain functions within the printer. They are called CONTROL CODES. This is great! By generating these control codes and putting them in your buffer you can send them to the printer while printing and control the printer functions! Now for the bad part: Although the numeric equivalents will always bring forth the same ASCII symbol, printer manufacturers do not use the same ASCII symbol and its numeric equivalent for the same printer function. This lack of uniformity requires that you carefully read your printer manual to see what the proper numeric equivalents are to implement the desired functions. This lack of standardization also stops us from supplying a chart showing how to uniformly affect certain printer functions.

A look at a typical smart printer shows that the user may choose from normal, elongated, compressed and other print modes and fonts, not to mention using superscripts, graphics and special character sets that the printer provides. To implement these functions requires that the printer be sent the proper control codes. The printer manual for your printer should have a control code summary chart which tells you which control codes will implement which functions. The chart should list the functions, and in columns next to the functions give the decimal and/or hexadecimal equivalents for the ASCII Control symbol which will implement the function. It is from this chart that you derive the control codes to use to make your printer dance.

The Super "Color" Writer II allows you to imbed these printer control codes right into your textfile. Control codes are normally imbedded using the printer driver (see Section 7-I) or are PROGRAMMABLE CHARACTER CODES (see Section 7-I). Yet, occasionally you will have to use P CODE markers for imbedding text (see Section 7-J). Imbedding control codes with P CODE markers requires entry of the actual ASCII or numeric equivalent of the code. An example should help here. The DMP 200 Printer manual states that to start an underline requires that the printer be sent the control code which is a decimal 15. This does not mean that you type "15" for the printer code. This would only send the printer two bytes of data, one a decimal 49 (hex 31) for the ASCII symbol for "1", and a decimal 53 (hex 35) for the ASCII symbol for "5". Instead, you must go to Appendix E, look in column four for decimal 15, and look in column seven for the correct manner of generating a decimal 15 to be sent to the printer. Appendix E shows that to generate a decimal 15, which is the numeric equivalent of the ASCII Control O, you must press <CLEAR>-<SHIFT><O>. This will produce a § (an inverse "/" in the 32 display) on the screen. When you generate a decimal 15 the system will store the decimal 15 in the buffer, and it will be sent to the printer when you print your buffer. This is how printer control codes between P CODE markers are handled. Use the procedures outlined above when you opt to use P CODEs to imbed control codes rather than using the printer driver or programmable character codes.

Before going on to the specifics of programmable character codes, creation of a printer driver, P CODE markers and other print commands, a word about a few common print control code practices. Many printers require that certain control codes ALWAYS be preceded by an ESCAPE code. Read your printer manual to see which, if any, of your printer functions require the use of the ESCAPE code. When the ESCAPE code is required, its decimal equivalent, 27 (generated by pressing <CLEAR>-<@> in the lower case mode ONLY) MUST be placed before the control code for the new print font or mode to be used. Another common control code

usage is to have one control code to turn on a function and another to turn off the same function. Thus, with the TRS-80 DMP 200 printer a decimal 15 starts underlining while a decimal 14 ends underlining. Some printer manuals explain that an ESCAPE plus a letter will work as a control code. Note that the letter MUST BE uppercase (uppercase characters have different decimal numeric equivalents than their lowercase counterparts - see Appendix E).

7-H AN INTRODUCTION TO IMBEDDING CONTROL CODES

The discussion on printers should have helped you understand the nature of printer control codes and how they fit in the ASCII system. Now it's time to learn how the Super "Color" Writer II "imbeds" those printer control codes so that you can control your printer.

Since printer control information is really only numbers being sent to the printer to have it perform a function, there is really only one form for a control code: a number between 0-31 and 128-255. Yet, logically, control codes can be split into two kinds. The first kind are control codes, usually found in the Standard ASCII Character Set, which actually command the the printer to change a print font, or use a feature such as underlining, superscripts, or elongated characters. The second kind are codes which tell the printer to print a character not available with the Standard ASCII Character Set, but are from the Extended ASCII Character Set of your printer, such as a copyright symbol or a British pound sign. Each of these is imbedded somewhat differently by the system.

The first kind are called generally "printer control codes." These may be imbedded in two different ways: as programmable printer codes from your printer driver, and as the codes themselves placed between special markers called P CODE markers. These two methods are not exclusive; either method may be used. The reason two methods were supplied was because of convenience and necessity. The Super "Color" Writer II allows you to customize a printer driver to control up to ten printer functions. The printer driver makes imbedding control codes into your text extremely easy. From this the convenience. Yet, because you might need to control more than ten printer functions, and because some printer functions, such as backspacing, are not easily implemented from a printer driver, another means to imbed control codes was necessary. The printer driver method is a procedure whereby you program keys of the keyboard to always implement one printer function. This bypasses the P CODE marker method in which you must enter the control code between P CODE pairs. Since these methods differ so, they will be discussed in separate sections.

The second kind of codes mentioned above are "programmable character codes," generally used to have the printer print special characters not available in the Standard ASCII Character Set. As the name implies, these character codes are programmed by you so that certain keys on the keyboard always implement each desired character code. The method for programming your character codes is essentially the same for programming your printer driver, so they will be discussed together.

7-I PROGRAMMING PRINTER DRIVERS AND CHARACTER CODES

Programming your printer driver and programming a set of character codes are complementary operations. With each there are ten keys which may be programmed. Furthermore, they are similarly imbedded in your textfile. They differ in that character codes are represented on the screen by the \SCII equivalent of the code programmed, whereas the control codes are represented by an inverse "d" (a magenta block in the 32 display) followed by the number of the printer driver code used. Also, printer driver codes are supplied P CODES by the system when you imbed them in your text, whereas if you are programming a printer control code into a character code you MUST supply P CODES yourself (P CODES are discussed below in Section 7-J). Finally, character codes may be used for programmable string insert, but driver codes MAY NOT.

Programming of character codes and driver codes is nearly identical. Both are done through FORMAT lines, although each may also be done from the COMMAND line. They both may be reprogrammed at any time. COMMAND line programming is inadvisable since the codes so programmed will not be saved when you save your file, nor are they easy to check. Nevertheless, it is convenient if you are in a hurry. FORMAT line programming allows your programmed codes to be saved for later reuse. They are also easier to refer to.

ONE ESSENTIAL THING TO REMEMBER about programming PROGRAMMABLE CHARACTER CODES (but NOT driver codes) FROM THE FORMAT LINE is that they will NOT be able to be implemented until you "WINDOW" through the FORMAT lines ONCE (see Section 6, FORMAT WINDOW). This is required to change the default values for programmable character codes. Therefore, you should place your programmed FORMAT lines at the beginning of your file, and WINDOW through them after you are done programming them and before you begin to use the individual codes. You only have to WINDOW once to set the codes for all further use in the file until any of the codes is reprogrammed, you load a file with different programmed codes, or turn your computer off.

Programming character codes will be discussed first. The programming principles discussed there will then be applied when discussing how you can customize your printer driver.

PROGRAMMABLE CHARACTER CODES

Many users of the Super "Color" Writer II will be interested in printing characters in their text which cannot be generated by the keyboard and are not part of the standard ASCII character set but which are available in the Extended ASCII Character Set supported by SOME printers. For example, DMP 200 Printer owners may want to print a copyright symbol; Epson owners may want to print German or French characters; Microline owners may want to print graphics in the text or may even want to use the Super "Color" Writer II to print a diagram along with text.

The Super "Color" Writer II supports all 128 ASCII characters from the keyboard. In addition, you are able to generate any character, including Extended ASCII Characters above 128, with any of the ten programmable characters. You must check your printer manual to see what symbols correspond to the numbers from 128-255 in the Extended ASCII Character Set.

To print these characters, the system supports ten programmable character codes that can be programmed to any value between 0 and 255 to create characters not normally available to you. These codes are used to send graphics or special characters to the printer. The screen display of PROGRAMMABLE characters is discussed below.

The ten PROGRAMMABLE CHARACTER CODES are generated in the text file using <CLEAR>-<SHIFT><1> through <CLEAR>-<SHIFT><9> and <CLEAR>-<SHIFT><:>. The <SHIFT><1>, etc., are the command keys (<SHIFT><1> equals "!", etc.). Initially the PROGRAMMABLE CODES are defaulted to generate some of the symbols in the Standard ASCII Character Set (see the asterisked key sequences in Appendix E). Until you program the key sequences to represent different character codes you may use them to represent their defaults.

Programming of the PROGRAMMABLE CODEs is done just as the FORMAT parameters are altered: through a FORMAT line, or through the COMMAND line. You may program UP TO TEN character codes into a single key sequence if you separate each by a semicolon. You may also place one or all ten programmed key sequences in the same format line, but each must be separated by a comma.

To program character codes, first generate a FORMAT marker (<CLEAR>-<K>). Now you simply create equations where the command key (!, ", #, $, %, &, ', (,), or *) equals (=)

the character code(s) to be programmed for that key. If more than one character code is being equated to the key, each must be separated by a semicolon.

The character codes may be entered either as the numeric equivalent (hex or decimal) or as the ASCII equivalent. CHARACTER CODES for printing Extended ASCII characters may ONLY be programmed with the numeric equivalent obtained by you from your printer manual. MARKERS and other characters may either be programmed using the ASCII or the numeric equivalent. Decimal numeric equivalents are directly entered after the equal sign; hex equivalents must be preceded by a dollar sign; a single ASCII equivalent MUST be preceded by an apostrophe (or alternatively may be enclosed in quotation marks); and if you are programming more than one ASCII equivalent they must be preceded and followed by quotes, but they are NOT separated by commas or semicolons.

As an example: The DMP 200 Printer prints a copyright character when it receives a decimal 171. To program the Super "Color" Writer II to send this character to your printer, first press <CLEAR>-<K> for a FORMAT marker. Next, press <SHIFT><1> to create an exclamation point, followed by an equal sign. Next press "171" for the character code. (Instead of pressing decimal 171, you can enter hexadecimal if you precede it with a "$".):

EXAMPLE 1: k!=171<ENTER>
EXAMPLE 2: k!=$AB<ENTER>

The character code 171 (a copyright symbol in the DMP 200 Printer) is now programmed. In the first example it was programmed using a decimal equivalent; in the second example it was programmed using the hexadecimal equivalent (AB hex = 171 decimal). Before this code may be implemented in your textfile you must WINDOW through the format line so that the default value is changed (defaults are listed in Appendix E). The copyright symbol is then implemented in your textfile not by generating an exclamation mark <SHIFT><1>) as for programming the character code, but by pressing <CLEAR>-<SHIFT><1>. Thus, to generate a copyright symbol for your DMP 200 Printer you would press <CLEAR>-<SHIFT><1> at the appropriate place in the file. The display would show an inverse "+" in the hi-res displays (a graphics block in the 32 display). Whenever the system encounters this character in the textfile it will send a decimal 171 to your DMP 200 Printer.

To program this example using the COMMAND line you would enter the COMMAND mode and press "!=171<ENTER>. COMMAND line programmed codes, just like COMMAND line altered FORMAT PARAMETERS, are immediately effective and may immediately be implemented. Character codes may be

reprogrammed at any time by changing the original FORMAT line, by entering a new FORMAT line elsewhere in the file, or through the COMMAND line.

The display of the Extended ASCII Character Set in the hi-res displays are the inverse of the characters from Ø to 127 in the Standard 128 ASCII Character Set. That is, the display for decimal 161 is the inverse of the display for 33 decimal (161-128=33), or an inverse exclamation point. To find the display of Extended ASCII Characters, 127 to 255, go to the Standard ASCII Character Set in Appendix E and find the correct number and its corresponding letter using a similar calculation. The display of Extended ASCII Character Set in the 32 display is the same as in BASIC (see your BASIC manual).

The following is an example of a FORMAT line which sets one programmable code do a decimal 201 (hex C9) and another programmable code to decimal 255 (hex FF).

EXAMPLE 1: k!=201,"=255<ENTER>
EXAMPLE 2: k!=$C9,"=$FF<ENTER>

k = FORMAT marker
! = <SHIFT><1>
" = <SHIFT><2>
$ = Signals that what follows is in hex.

To generate the first (after WINDOWING the FORMAT line) you would press <CLEAR>-<SHIFT><1> ("!"); to generate the second you would press <CLEAR>-<SHIFT><2>('"'). The first would be displayed as an inverse exclamation mark; the second as an inverse quotation mark.

One very beneficial use for PROGRAMMABLE CHARACTER CODES is to create special characters not available from the Standard ASCII Character Set but which may be created using other characters from that Set. The Standard ASCII Character Set does not contain a symbol for a left arrow; yet you can easily create one by pressing "<", backspacing and then pressing "-". You can program a CHARACTER CODE to make your printer "automatically" do this for you at the press of one key sequence. The following example does this and illustrates all the programming rules:

EXAMPLE: k!='<;'Ø;8;"-Ø"<ENTER>

! = <SHIFT><1>
' = Signal that what follows is an ASCII symbol.
8 = Printer code for a backspace character.
" = Signals that contents are ASCII equivalents.
Ø = P CODE marker.

This example programs the key sequence <CLEAR>-<SHIFT><1> to create a left arrow symbol. Note that MARKERS are programmed as ASCII equivalents. Note also that when you

program more than one ASCII character they must be placed between quotation marks, and they are not separated by commas or semicolons. The P CODES (discussed in Section 7-J) are required because whenever you backspace and "print over" a character, if the backspace character and character to be printed are not within P CODES the character count for justification would be thrown off (see Section 7-J on P CODES and Section 7-K on underlining and backspacing). When you press this sequence at the appropriate place in you textfile (after WINDOWING through the FORMAT line) you will see five characters: "<", "Ø", "←" and "-" (the "8" is represented by a left arrow - see Appendix E) and "Ø". When the printer receives these codes it will create a left arrow for you.

As you can see, the PROGRAMMED CHARACTER CODE allows you to enter many characters by pressing just one key sequence. You could yourself enter this same sequence of characters with the same result, but PROGRAMMABLE CHARACTER CODES make the job simple, and if you have to use the character over and over, the trouble is well worth it. You can use this same method to create foreign language letters, such as a circumflexed vowels to be entered with one simple key sequence.

In fact, you can actually use PROGRAMMABLE CHARACTERS (but not driver codes - discussed below) to program any character string with up to TEN characters per code. The characters can be any regular character and even markers. They may NOT, however, be COMMANDS. (For programming a series of commands see Section 8, PROGRAMMABLE FUNCTIONS.) Thus, once programmed you can, with the press of one key sequence, have whole words, even P CODE strings, placed in the file. And if you have toggled perpetual character insert, you essentially have a programmable string insert (see also Sections 3-B and 8-B). If you are going to program markers, such as P CODE markers (discussed below), however, you MUST place the ASCII character in the equation, and not the numeric equivalent, i.e., for a P CODE you must press <CLEAR>-<Ø> (see the last example).

EXAMPLE 1: k!="String"<ENTER>
EXAMPLE 2: k$="Ø:XØ"<ENTER>

k = FORMAT marker.
: = ESCape code.
Ø = P CODE marker.
! = <SHIFT><1>
$ = <SHIFT><4>
X = Any printer code.

The first example programs the key sequence <CLEAR>-<SHIFT><1> to be the word "String". The second example programs the key sequence <CLEAR>-<SHIFT><4> to be a printer control code. Now, when you press either of these

sequences, (after you have WINDOWED the FORMAT line) the specified programmed string will be placed in your file, and subsequently sent to the printer. If you had perpetual INSERT on, you could also INSERT the "String" or the printer control code in your file.

As you will see, PROGRAMMABLE PRINTER CODES could be used to create a file of additional programmed printer codes to supplement your programmed printer driver. The difference between the character codes and driver codes lies in how the programmed codes will be displayed. PROGRAMMED CHARACTER CODES are actually displayed as the string programmed; the programmed printer driver code strings are represented by a two character code (see discussion below).

PROGRAMMING PRINTER DRIVERS

The Super "Color" Writer II provides the ability to customize a printer driver to control up to ten of the usual printer control functions such as turning on and off bold, underlining, superscripts, subscripts, a different print font, or any unusual printer function you wish to use. Programming and implementation of your printer driver is nearly identical to that for PROGRAMMABLE CHARACTER CODES discussed above. Once you have created your customized printer driver, you can save it for permanent use with your printer. To use it again all you need do is load it at the beginning of your file and print your formatted file.

Programming of printer codes may also be done through FORMAT lines or the COMMAND line. To create your printer driver you program each of the ten programmable driver codes, if you desire that many. (It helps to COMMENT your printer driver format lines for easy reference when you need to remember what the codes are for.) You have ten printer code strings you can program, controlled by <CLEAR>-<1> through <CLEAR>-<9> and <CLEAR>-<:>. Once programmed, these key sequences will make your word processing much simpler.

To create your printer driver you will first have to determine the ten printer control functions you most wish to include. Don't forget that you may need different codes for turning on and turning off certain functions such as underlining. The programming rules are the same as for PROGRAMMABLE CHARACTER CODES discussed above.

Some examples should help you learn how to customize your own printer driver. The following commented examples show how to program a printer driver for the DMP 200 Printer to turn on Bold, off Bold, on Underlining, off Underlining, Half Reverse Linefeed, Half Forward Linefeed, Select Condensed Characters, Select Standard Characters, Elongation on, and Elongation off:

± This DMP 200 printer driver performs as follows: 1 Bold on; 2 Bold off; 3 Underline on; 4 Underline off; 5 Half Reverse Linefeed; 6 Half Forward Linefeed; 7 Select Condensed Characters; 8 Select Standard Characters; 9 Elongation on; and : Elongation off.

```
EXAMPLE 1:  k1=27;31, 2=27;32, 3=15, 4=14, 5=27;30, 6=27;28,
             7=27;20, 8=27;19, 9=27;14, :=27,15<ENTER>
EXAMPLE 2:  k1=$1B;$1F, 2=$1B;$20, 3=$0F, 4=$0E, 5=$1B;$1E,
             6=$1B;$1C, 7=$1B;$14, 8=$1B$13, 9=$1B;$0E,
             :=$1B,$0F<ENTER>
```

± = COMMENT marker.
k = FORMAT marker.
1 = <1> key.
2 = <2> key.
3 = <3> key.
4 = <4> key.
5 = <5> key.
6 = <6> key.
7 = <7> key.
8 = <8> key.
9 = <9> key.
: = <:> key.
27= Escape code.
$ = Signal that a hex number follows.

(THIS PRINTER DRIVER GIVES THE DEFAULT VALUES FOR THE KEYS IN THE SUPER "COLOR" WRITER II. THUS, IF YOU OWN A DMP 200 PRINTER, AND IF YOU ARE SATISFIED WITE THIS PRINTER DRIVER, YOU DO NOT HAVE TO CUSTOMIZE YOUR OWN. THE KEYS ARE ALREADY SET TO PERFORM THE FUNCTIONS INDICATED.)

The examples program ten number keys, <1> through <9> and <:>, one complete printer driver, to perform the functions indicated. Example 1 was programmed strictly in decimal numbers; example 2 was programmed strictly in hex. You could have programmed key <2> in an ASCII equivalent instead of in the numeric equivalent as follows:

```
EXAMPLE 1:  k2=27;'<SPACE><ENTER>
EXAMPLE 2:  k2=27;"<SPACE>"<ENTER>
```

The space is the ASCII equivalent for 32 decimal and 20 hex. As you can see, you can either precede an ASCII equivalent with an apostrophe or you can surround it with quotation marks. Thus, if your printer manual gives you ASCII equivalents for the printer codes you can use the ASCII programming rules to create your printer driver. On the other hand, you can go to Appendix E to convert the ASCII characters into their numeric equivalents for programming. We recommend the latter for convenience and simplicity.

Programming driver codes may also be done from the COMMAND line. Using the space example above you would type from the COMMAND mode: .2=27;'<SPACE><ENTER>.

Any of the driver codes may be reprogrammed by changing the FORMAT line. You may also alter it for part of your file by creating a new FORMAT line elsewhere in your file.

To implement any of these programmed printer codes and imbed them into your textfile all you have to do is press the key sequence for the printer function programmed. In the above example, to turn on BOLD, you would, at the appropriate place in your text, press <CLEAR>-<1>. This key sequence will be represented in your display by an inverse "d" (a graphics block in the 32 display), which stands for "Driver", followed by a "1", and surrounded by system supplied P CODE markers. Other keys will be represented by the inverse "d" and the number of the key, plus the P CODE markers.

NOTE: It must be emphasized that although P CODES will be automatically inserted in the system for your ten programmed printer codes, P CODES will still have to be used for printer codes not in your printer driver and other special functions such as using backspace characters. Thus, for those rare instances when you must go outside your customized printer driver to use a special feature, P CODES are in order. P CODES are discussed below.

These examples should give you an idea of just how easy it is to create one or more printer drivers. You could create multiple printer drivers for different editing functions you do, or for different printers. If you own two different printers the printer drivers can be made a matching set. The number keys can be programmed to always represent the same function, e.g., <1> always turns on BOLD and <2> always turns on UNDERLINING. Of course, the numeric or ASCII equivalent for the printer functions in the two printers will probably differ, and thus this will have to be considered when programming each printer driver. This method assures that the same text can be printed on both printers without changing printer codes in the text, provided you load the correct driver before printing.

If you find ten driver codes insufficient, remember that you can program up to ten character codes as printer code strings, giving you a total of twenty programmable, alterable, saveable and reloadable printer and driver codes. This should handle just about every need.

7-J CONTROL CODES IN THE TEXT <CLEAR>-<Ø>

As was explained above, the ten functions programmable with your printer driver plus ten potential programmable character codes may not be enough to handle those occasional special needs. Moreover, some printer functions are not easily handled using the printer driver (i.e., underlining with some printers and functions requiring extensive backspacing). Therefore, the Super "Color" Writer II provides another means for imbedding printer control codes in your file: Printer control codes can be imbedded in the textfile with special markers called P CODE markers (see Section 7-G). These markers are represented as an inverse "Ø" (a BUFF block in the 32 display) and are generated by pressing <CLEAR>-<Ø>. Technically, these markers set aside a STRING of codes which is sent to the printer, but is not treated as printable text. This is important to note, as the system counts all PRINTABLE TEXT for justification purposes. If the codes were counted along with the printable text, the justification count would be wrong.

P CODE marker pairs can be used ANYWHERE within the textfile as long as they are used in PAIRS! P CODE pairs need not be terminated by an <ENTER> unless you desire a carriage return to end a line. If P CODEs are used at the beginning of a paragraph and you wish to indent, the indented spaces must follow the P CODE pair. NOTE that when you are using programmed driver codes from your customized printer driver (see Section 7-I) the system will automatically supply the P CODE pair to save you the trouble. If, however, you are programming character codes to function as a supplemental printer driver, you MUST program P CODES around the printer codes.

Below are three examples of how P CODE markers might be used to imbed control codes within the textfile.

REMEMBER! THESE ARE ONLY EXAMPLES. PLEASE substitute your particular printer's codes as generated according to Appendix E in place of the symbols used.

The example below uses codes to underline text. Notice that the P CODE pairs are placed directly before and after the text to be underlined. This is IMPORTANT!

EXAMPLE 1: This is an ØXØunderlineØYØ test.<ENTER>
Ø = P CODE marker.
X = Control code to turn on underline.
Y = Control code to turn off underline.

The example below uses codes to boldface text. Notice that more than one code can be placed within P CODE pairs. It is advisable to switch on the new font to be used, if any, right after you have switched off the print font just used between the same P CODE pair. It's convenient too.

EXAMPLE 2: This is a ∅:X∅boldface∅:Y∅ test.<ENTER>
∅ = P CODE marker.
: = ESCAPE code.
X = Code to turn on boldface.
Y = Code to turn off boldface.

The next example switches print sizes with codes. Note that P CODEs need not be followed by an <ENTER>.

EXAMPLE 3: ∅:Z∅
∅ = P CODE marker.
: = ESCAPE code.
Z = Code to turn on alternate print size.

Certain printers require that some or all control codes be preceded by an ESCAPE code in the control sequence. ESCAPE is generated from the keyboard, in the lowercase mode ONLY, by pressing <CLEAR>-<@>.

RULE: P CODE markers must be used in PAIRS...OR ELSE!

7-K UNDERLINING, BACKSPACES, SUPERSCRIPTS AND SUBSCRIPTS

UNDERLINING

Underlining has obvious advantages for emphasizing text, underlining book titles, or other uses. The Super "Color" Writer II supports underlining for all printers, and allows you to underline within justified text. Underlining is a printer-controlled feature. There are essentially two ways that printers control underlining: automatic and with backspaces. If your printer supports automatic underlining, this can be incorporated in your printer driver (see Section 7-I), and there is no need to read this section. If your printer does not support automatic underlining, this section will prove helpful. In fact, some printers do not support underlining at all, in which case this section will be of no avail. (See your printer manual.)

Automatic control of underlining is through use of printer control codes to turn on and off underlining before and after the text you want underlined. This is easily accomplished by generating the numeric equivalents of the correct control codes in your printer driver or between P CODE markers.

The second method, used with older printers that do not support underlining directly, requires that your printer supports backspace characters because the print-head must be backspaced over the text to be underlined the exact number of spaces of the text being underlined, and then the same number of underline characters must be sent to the printer (underline characters are generated by pressing

<SHIFT><UP ARROW> - see Appendix E). To do this with the Super "Color" Writer II you must place all backspace and underline characters between P CODE markers. This is required to stop the system from counting the backspace and underline characters when calculating line length for justification, and thereby throwing off justification.

EXAMPLE 1: Ø≠ØUnderlining is turned on and off.Ø$Ø
EXAMPLE 2: Underlining|with|backspacesØ<<<<<<<<<<<<<<<<<<<<
<<<<<<<_____Ø

Ø = P CODE marker.
≠ = Code to turn on underlining.
$ = Code to turn off underlining.
< = Backspace character.
_ = Underline character.
| = NONSPACE marker.

(NOTE that a NONSPACE marker is generated with a <CLEAR>-<SPACE>, and is normally represented in this manual by an underlined space. Here this is changed to avoid confusion with the underline character.)

RULE: Backspacing and underlining functions may not be performed on words to be located on more than one line unless you turn underlining off at the end of one line and turn it on at the beginning of the next.
RULE: When underlining more than one word using backspace characters, the words must be joined with NONSPACE markers, as in Example 2 above.

BACKSPACING

Backspacing, a printer controlled feature, can have other uses besides underlining. Placement of tilde's and other diacritical marks usually will require you to backspace the printhead for proper placement. (You may also program PROGRAMMABLE CHARACTERS to make letters with their special diacritical marks.) Owners of some printers will want to use backspaces to make use of the printer's double strike capability if your printer does not have an automatic double-strike capability. Whatever the purpose, backspace characters and the text to be printed in the backspaced area MUST be put between P CODE markers. NONSPACE markers (see Section 5-G) must be used for spaces where the printer will not be printing anything in the backspace area. The backspace character (decimal 8) is generated by pressing <CLEAR>-<SHIFT><H> (see Appendix E).

EXAMPLE: Double_strikeØ<<<<<<<<<<<<<Double_strikeØ
Ø = P CODE marker.
< = Backspace character.
_ = NONSPACE marker.

In this example the backspace characters are used to allow printers which do not support it directly to double strike the phrase "Double strike". Between P CODE markers are placed the same number of backspace characters as there are characters in the string "Double strike", and then that string is sent to the printer again so that it will type over the first string to affect double strike. NOTE that NONSPACE markers must be placed between words in the backspace function.

SUPERSCRIPTS AND SUBSCRIPTS

Superscripts and subscripts are printer controlled features. You can only have them if your printer allows you to. If your printer supports superscripts and subscripts (see your printer manual), they are enabled by incorporating the control codes in your printer driver or between P CODE markers (see Sections 7-I and 7-J) just like other printer functions enabled by printer control codes. Note that some printer manuals refer to superscripts as half-reverse linefeeds, and to subscripts as half-forward linefeeds.

7-L PAUSE PRINT MARKERS <CLEAR>-<=>

The PAUSE PRINT function is a special feature which allows you to pause the printer for such special applications as changing thimbles or daisy wheels on letter quality printers. This PAUSE PRINT function is intended for use with DAISY-WHEELs, TTY's, and other non-buffered printers. The PAUSE PRINT function WILL USUALLY NOT work properly on any printer that uses a buffer to store text. Experiment with your printer to see if the PAUSE PRINT function will work with it.

The PAUSE PRINT function is enabled by using the PAUSE PRINT marker BETWEEN P CODE MARKERS. The PAUSE PRINT marker is generated by pressing <CLEAR>-<=> and is represented by an inverse "=" (an ORANGE diagonal block in the 32 display). When the system encounters the PAUSE PRINT marker (between P CODE markers) it stops printing at that point. After you have completed the task of changing the thimble or whatever, you may restart printing by pressing any key.

EXAMPLE: The printer will stop here.Ø=Ø
Ø = P CODE marker.
= = PAUSE PRINT marker.

7-M NO-PRINT MARKER <CLEAR>-<N>

The Super "Color" Writer II provides a means to segregate portions of your text from that which you want to

be printed, without deleting the segregated text from the file. Text to be segregated is placed between NO-PRINT markers. These markers MUST BE at the beginning of a new line (the previous line must have been terminated by a carriage return). To generate a NO-PRINT marker, press <CLEAR>-<N>. The NO-PRINT marker is represented by an inverse "n" (an ORANGE block absent the lower right-hand corner in the 32 display). When the printer encounters the first NO-PRINT marker it ignores the following text until it encounters another NO-PRINT marker which toggles the system back to resume printing from that point.

EXAMPLE:
nAll of this text, which was preceded by the first NO-PRINT marker, is text that WILL NOT BE printed.<ENTER>
nThis text, which was preceded by a the second NO-PRINT marker, WILL BE printed.

n = NO-PRINT marker.

SECTION 8 PROGRAMMABLE FUNCTIONS

8-A PROGRAMMABLE FUNCTIONS <CLEAR>-<Z>

PROGRAMMABLE FUNCTIONS are powerful functions which can be used to create any commands you can dream up. You may create up to three "newly created commands" (PROGRAMMED FUNCTIONS). All of these new commands are stored in buffers and can be used whenever you wish. These PROGRAMMED FUNCTIONS are created by incorporating any combination of up to 28 of the commands, functions, modes or keystrokes in the Super "Color" Writer II, and each PROGRAMMED FUNCTION can be executed up to 65535 times.

Some examples will greatly clarify the usefulness of this feature and the mechanics of its use. But first an outline of how to "create a command." The three PROGRAMMED FUNCTIONS you can create are an E FUNCTION, an R FUNCTION and a T FUNCTION. Programming of each PROGRAMMABLE FUNCTION option is initiated by pressing <CLEAR>-<Z>, which puts the prompt "PROG FUNCTION" on the COMMAND line. In response, you must indicate which of the three functions you wish to PROGRAM: either <E>, <R> or <T> (these letters were arbitrarily chosen to be conveniently located on the keyboard). After you have chosen the particular FUNCTION you wish to program by pressing <E>, <R>, or <T>, you may select either instant execution of your function on request or authorized execution (i.e., you have to answer <Y> to the "ARE YOU SURE?" prompt) by pressing an <UP ARROW> for instant or a <DOWN ARROW> for prompted execution. After you have selected execution by pressing one of the arrows, a blank COMMAND line will be displayed awaiting entry of up to 28 commands, characters, etc. Once you have entered your FUNCTION, you may press <DOWN ARROW> and the "#" prompt will appear. You then enter the number of times you desire the function to be performed (up to 65535 times) and press <ENTER>. This completes the programming of your desired FUNCTION. Reprogramming uses the same procedures.

When you want any of your PROGRAMMED FUNCTIONS to be executed, place the cursor in the desired position, and press <CLEAR> and then the letter of your function, i.e., E, R or T. If you have selected instant execution the function will immediately occur; if you selected prompted execution the system will respond with the "ARE YOU SURE?" prompt. Pressing <Y> will execute the specified PROGRAMMED FUNCTION the designated number of times. Pressing any other key or <BREAK> will exit to the TEXT MODE.

CASSETTE TEXTFILE LINKING

EXAMPLE 1: An excellent example of how this function can be used is the programmed function we call "cassette file linking". This function is a combination of the following commands: "CLEAR" the textfile, "CL"oad a

textfile from cassette and <P>rint the textfile that was just loaded. This programmed function can be executed (repeated) as many times as you specify. Before trying this example you need at least two consecutive formatted textfiles (see Section 8-C Disk File Linking) on a cassette. You will also need to use the remote jack on your cassette player. Ready the cassette and printer, and let's go.

To program the PROGRAMMABLE FUNCTION to link consecutive cassette files type <CLEAR>-<Z>. (For this example you'll probably want to CLEAR the text buffer first.) The system will respond with the prompt "PROG FUNCTION", asking for you to designate the FUNCTION name, E, R or T. Let's press <E> here. Now press <DOWN ARROW> for prompted execution. The system will now display a blank command line with an "L" on the right. Any command, function or keystroke (except <BREAK> - see rule 7 below) can be typed including <ENTER>. Type the commands just as they are ordinarily entered (up to 28 keystrokes) (subject, however, to some rules which will be covered in a later example). For this example, enter the keystrokes just as they appear below. Remember that <CLEAR> represents the clear key while <C-L-E-A-R> represents pressing the keys C-L-E-A-R. On the blank command line type the following command string:

<CLEAR> <CLEAR> <C-L-E-A-R> <ENTER> <Y> <C-L> <ENTER>
</> <ENTER> <CLEAR> <P> <P> <DOWN ARROW>

This sequence will CLEAR the buffer, enter the COMMAND mode, Cassette Load a file into the buffer, exit the COMMAND mode and then PRINT the buffer contents. The <DOWN ARROW> tells the system that entry of the "command string" is complete. The system will respond with the prompt "#". Enter the number of times that you want the function executed. For this example, we will use the two consecutive textfiles on tape, so type <2>-<ENTER>. Any number from 1 to 65535 can be used. FUNCTION E is now programmed and ready for prompted execution.

Set the cursor to the desired position and we'll be ready to execute the function. Press <CLEAR>-<E> to call PROGRAMMED FUNCTION E. The system will respond with an "ARE YOU SURE?" prompt. Press <Y>. If your entries are correct the system will execute your command. If you press any other key the system will exit to the TEXT mode.

If you would like to try "cassette file linking" again, there is no need to re-program the function. Your new command will be stored in the buffer as PROGRAMMED FUNCTION E until you re-program it or turn the computer off. Just move the cursor to the position you want your function to start and press <CLEAR>-<E>. The system will again respond with an "ARE YOU SURE?" prompt on the command line. Pressing <Y> will execute the function.

But what if you want to change the number of times an existing PROGRAMMED FUNCTION is executed? Press <CLEAR>-<Z> as if to reprogram a function and answer the "PROG FUNCTION" prompt with the name of the FUNCTION you desire to change, here <E>, and <DOWN ARROW>. Since you do not wish to completely reprogram the function, but only change the number of times it is executed, next press <UP ARROW>. This will keep the PROGRAMMED FUNCTION the same, and will elicit the prompt "#" so that you can change the number. Now enter the new number and press <ENTER>. PROGRAMMED FUNCTION E is ready for prompted execution the new number of times.

CREATE A COLUMN

EXAMPLE 2: Another example, which covers the other programming rules governing this command, consists of two separate programmed functions. Let's call the first newly programmed function "create a column". CLEAR the text buffer, move the cursor to TOP, press <CLEAR>-<Z>, select your PROGRAMMED FUNCTION, here <R>, and press <DOWN ARROW> for prompted execution. On the blank command line type the following command string exactly:

This test creates a column<ENTER>

Press <DOWN ARROW> to tell the system that you are done entering the command, respond to the "#" prompt with a "15" and <ENTER>. Next execute the FUNCTION by pressing <CLEAR>-<R> and answering the "ARE YOU SURE?" prompt with a <Y>. This command creates a column of 15 lines of text.

Let's call the second programmed function "column insert". To perform this new command you must understand the arrow functions. All commands and keystrokes other than the arrow keys perform the same functions when PROGRAMMING FUNCTIONS as in the text mode. When PROGRAMMING a FUNCTION the ARROW keys, however, perform triple duty. First, they move the cursor right or left on the command line, their usual functions. The ARROW keys also serve to tell the system what to do with the cursor during the newly created FUNCTION (e.g., move up, down, left or right). The arrow keys also are used in this system to complete parts of a stepped command, such as the use of the <DOWN ARROW> in the creation of the PROGRAMMED FUNCTION to end the entry of the functions and to elicit the "#" prompt. "Column insert" is a good example to illustrate how they work.

Move the cursor to the TOP of the file again. You should now see the column of text you just created. Press <CLEAR>-<Z>, then <T>, a <DOWN ARROW> for prompted execution, and on the blank command line enter the following command string as it appears:

<CLEAR> <I> <SPACE> <SPACE> <SPACE> <SPACE> <SPACE>

Now press <LEFT ARROW>. Notice that when you pressed the <LEFT ARROW>, the cursor backspaced over what you just typed. Restore the <SPACE> and press <SHIFT><Ø> to change the arrow mode. In this mode the case indicator will change to a "U". Now if you press the <LEFT ARROW>, a representation of the <LEFT ARROW> will appear. (In the "U" arrow mode in the 32 display ONLY the only confusing symbol is the <DOWN ARROW> which is represented by an asterisk.)

Let's continue where we left off entering functions. In the "U" mode enter the following:

<LEFT ARROW> <LEFT ARROW> <LEFT ARROW>
<LEFT ARROW> <LEFT ARROW> <DOWN ARROW>

Now press <SHIFT><Ø> to change the mode and then a <DOWN ARROW> to end the programmed line, answer the "#" prompt with "15" and an <ENTER>, press <CLEAR>-<T>, answer the "ARE YOU SURE?" prompt with a <Y> and watch! If everything went according to plan, the new FUNCTION should have inserted five spaces at the beginning of the line, backspaced to the beginning of that line and moved down to the next line, 15 times. Note that this INSERT function did not have to be programmed in; instead, you could have executed the five space insert while in the PERPETUAL INSERT mode (see Section 3-B).

To re-execute this example you need only set the cursor at the desired position, turn off perpetual INSERT by pressing <CLEAR>-<I>, press <CLEAR>-<R> and then <Y>, move the cursor back to the beginning after the R FUNCTION is through and press <CLEAR>-<T> and then <Y>. You could also reprogram the R FUNCTION to move the cursor to the HOME position and then execute the T FUNCTION itself.

To clear the PROGRAMMED FUNCTIONS, press <CLEAR>-<Z>, <R>, press the <DOWN ARROW> twice, and then press <ENTER>, and do the same for the <T> FUNCTION. Of course, reprogramming of a FUNCTION will clear the old function too.

The examples above should serve to illustrate just how powerful this command really is. Other helpful uses for the PROGRAMMED FUNCTIONS include creating functions to INSERT and DELETE spaces when formatting columns. If you experiment some on your own you'll find the possibilities to be endless. To amuse yourself, try animating a little cartoon using the graphics blocks and the paging commands.

THE FOLLOWING ARE SOME SPECIAL RULES:

1) PROGRAMMABLE FUNCTIONS begin execution from the TEXT mode; thus COMMAND mode commands must be preceded by <CLEAR>-<CLEAR>. </>-<ENTER>. exits the COMMAND mode.

2) PROGRAMMABLE FUNCTIONS start from the cursor position.

3) A <DOWN ARROW> in the "L" mode after entry of the desired function letter elicits prompted execution; an <UP ARROW> elicits IMMEDIATE execution.

4) <SHIFT><Ø> toggles "L" and "U" for arrow modes.

5) Programmable character codes (see Section 7-I) must be entered from the "U" mode.

6) Control codes <CLEAR>-<SHIFT><A-Z> (decimal 1-26) <SHIFT><@> and <CLEAR>-<SHIFT><CLEAR> must be entered by pressing <CLEAR> from the "U" mode, and then entering the remaining keystroke from the "L" mode.

7) When programming a function you may need to execute a <BREAK> from the COMMAND mode. Pressing <BREAK) itself when programming the FUNCTION will not program the FUNCTION to break, but instead will break out of the programming procedure altogether. To program the function to break, <BREAK> (the same as <CLEAR>-<SHIFT><C>) MUST be entered as follows: In the "U" mode press <CLEAR>; then return to the "L" mode and press <SHIFT><C> (see Rule 6).

8) You MAY program PROGRAMMED FUNCTIONS into other PROGRAMMED FUNCTIONS, i.e., an E into an R. When you do this you do not have to follow the imbedded <CLEAR>-<E> (or <R> or <T>) with a <Y> to authorize use of the imbedded PROGRAMMED FUNCTION. YOU MAY NOT, however, program a function into itself, or program another function containing a particular PROGRAMMED FUNCTION into that function (i.e., if your R has a T, you MAY NOT program the R into the T).

SECTION 8-B PROGRAMMABLE STRING INSERT

As was seen in the Column Insert example in Section 8-A, the INSERT command can be used to great advantage as part of a PROGRAMMABLE FUNCTION. When so used, the INSERT command becomes a great supplement to the LOCATE AND CHANGE command. LOCATE AND CHANGE is used to change, add to, or subtract from a particular string of text (see Section 3-D). LOCATE AND CHANGE requires, however, that the same string throughout the textfile be altered. For example, you might wish to change the contents of every P CODE pair which turns on the elongated print font so that it will instead turn on the italics font. To do this you would LOCATE AND CHANGE the desired strings from Ø$Ø to Ø#Ø (Ø illustrates the P CODE MARKER, $ illustrates the printer code for elongate and # illustrates the printer code for italics).

But what if you wish to add a particular string of text at different places in your textfile unrelated to any

particular string of text; for example, what if you decided that several portions of your text currently in the normal print font should be printed in the emphasized print font? LOCATE AND CHANGE would not be of much help since the strings by which you would want to place the appropriate P CODE pairs would not be the same. Without some other means to insert the P CODE pairs, you would have to laboriously insert every character of every P CODE pair everywhere you desired to change the print font.

To the rescue the PROGRAMMABLE STRING INSERT FUNCTION. Instead of that laborious task, you can instead create two PROGRAMMABLE FUNCTIONS, one to turn on emphasize by LINE INSERTING the P CODE pair with the appropriate printer code just before the text to be emphasized, and one to turn off emphasize by LINE INSERTING a P CODE pair with the appropriate printer code after the text to be emphasized (if this is the method prescribed by your printer manual). Of course, character insert may also be used instead of line insert. To create the first PROGRAMMABLE FUNCTION:

Press <CLEAR>-<Z>, select the desired FUNCTION (here let's use E), press <UP ARROW> for immediate execution, and then enter on the blank command line: <CLEAR>-<O> <CLEAR>-<Ø> <(your printer code for turning on emphasize)> <CLEAR>-<Ø>, go into the "U" mode and press <UP ARROW> and return to the "L" mode. These last keystrokes instruct the FUNCTION to BREAK out of the LINE INSERT function - see Section 3-B. Next press the <DOWN ARROW> and respond to the "#" prompt with a 1 and press <ENTER>. (Instead of using the LINE INSERT function, you could program the P CODE and execute the function with PERPETUAL INSERT on.)

The second FUNCTION to turn off emphasize would be exactly the same except that you select a different FUNCTION letter (here let's use R) and you would put your printer code for turning off emphasize between the P CODE MARKERS. Both FUNCTIONS are programmed to be performed only once.

Now for the labor saving! Move your cursor to the position where the P CODE string is to be inserted and execute the E FUNCTION by pressing <CLEAR>-<E>-<Y>. The E FUNCTION will be executed once, INSERTING the P CODE string. To INSERT the P CODE string to turn off the emphasize font, similarly execute FUNCTION R.

PROGRAMMABLE CHARACTER CODES (Section 7-I) can also be used for programmable string insert, in fact, for the identical function described above, except with character insert instead of line insert. You would choose a PROGRAMMABLE FUNCTION when you wish to combine several COMMANDS, such as cursor movements. Thus, PROGRAMMABLE FUNCTIONS are the choice for INSERTING and DELETING spaces when altering columns.

8-C LINKING DISKETTE FILES FOR CONTINUOUS PRINTING

The Super "Color" Writer II has the ability to link diskette textfiles together for continuous printing of lengthy documents. This feature is totally programmable for hands-off operation once the files have been properly formatted.

Diskette file linking is accomplished by using the PROGRAMMABLE FUNCTION to successively load and print as many textfiles as desired and is similar to cassette file linking outlined in Section 8-A with two differences: 1) Textfiles stored on the diskette have filenames and unlike tape files do not require sequential access, so a list of the filenames desired ("LINKING LIST") must be created at the top of the text buffer to tell the system which files are to be loaded and in what order; and 2) The special Disk Load command for linking, "DL;", is used instead of "CL" for tape file linking.

Linking can be done in many ways, but for simplicity, we suggest the following procedure. Each textfile to be linked should be saved as a group of full pages of formatted text. Your many textfiles comprised of such grouped pages can then be easily linked. To group your pages and create a linkable textfile, first use the FORMAT WINDOW to determine the beginning and end of each page of text for the group of pages constituting a textfile. Be sure to use FORMFEEDS (see Section 5-H) at the end of your files if you want each formatted file to begin at the top of a new page. A format line containing all of the pertinent format information such as page number (PN), page number line (PG), margins (LM,RM) and so on, should be placed at the top of each of the textfiles to be linked. Once the textfile is properly formatted, save the file with the appropriate name and continue to the next group of pages. After the linkable textfile groups are created, you next generate a "LINKING LIST".

The linking process can be explained best by using an example. Assume that four formatted textfiles have been saved to the diskette. Each textfile has six formatted pages of text for a total of 24 pages. The first file has the name "FILE1/SCW" (pages 1-6). The second file has the name "FILE2/SCW" (pages 7-12). The third file has the name "FILE3/SCW" (pages 13-18). The last file has the name "FILE4/SCW" (pages 19-24).

A "LINKING LIST" must now be created using these files. The "LINKING LIST" will be the only resident text in your buffer when you begin linking. To create the "LINKING LIST" a LINK marker must be used. This marker sets aside textfile names. The LINK marker is represented on the video display as an inverse lowercase "L" (a BLUE diagonal block in the 32

display). The command key sequence for the LINK marker is alternated with that for the AUXILIARY B marker: <CLEAR>-">" (there are only so many keys on the keyboard!). The default marker generated is the AUXILIARY B marker. To toggle to generate the LINK marker, press <A>-<ENTER> from the command mode. Each LINK marker must be followed by a filename (same rules apply as in disk access) and terminated by an <ENTER>. The LINKING LIST MUST be at the TOP of the text buffer. So, for our example:

EXAMPLE:
LFILE1/SCW<ENTER>
LFILE2/SCW<ENTER>
LFILE3/SCW<ENTER>
LFILE4/SCW<ENTER>
X (marks the spot where the cursor must be located when the PROGRAMMED FUNCTION is implemented).

NOTE: An underlined "L" is used to represent the LINK marker.

The above "LINKING LIST" constitutes the list of filenames that the system will use while in the linking function. The order in which they appear, from top to bottom, is the order in which the files will be loaded and linked. (Once the "LINKING LIST" has been created, it can be saved for future use.)

Diskette file linking is accomplished by using the PROGRAMMABLE FUNCTION to perform the following functions:

1) Clear the buffer AFTER the "LINKING LIST" (this preserves the "LINKING LIST" through the entire linking process).

2) Load a file, using the next filename in the "LINKING LIST" sequence, from the diskette.

3) Print the textfile.

4) Execute this programmed function for the number of filenames in the "LINKING LIST".

The special linking command, "DL;" is used rather than "DL" so that the system knows to take the filenames, in order, from the "LINKING LIST" at the beginning of the text buffer.

Using the "LINKING LIST" example above, the procedure for diskette file linking is as follows:

1) Position the cursor at the line below the last entry in the "LINKING LIST" and all the way to the left (X marks the spot in the example above).

APPENDICES

APPENDIX A

ASCII FILE HANDLING

You are able to load, create, edit or save any kind of text file saved in ASCII with the Super "Color" Writer II. This means that you can load BASIC programs that have been saved using the ASCII option (CSAVE"filename",A<ENTER> for tape or SAVE"filename",A<ENTER> for disk), ASCII files and BASIC programs that were created with Super "Color" Library programs, or any ASCII file that was created using the Color Computer, right into the textfile buffer to be edited, saved to tape or disk, or printed!

APPENDIX B

EDITING BASIC PROGRAMS

ADHERE TO THE FOLLOWING RULES WHEN EDITING BASIC PROGRAMS:

1) All commands, strings and variables MUST be in UPPER CASE.
2) Program lines MUST be preceded by a line number and a space,
3) Line numbers must be in sequential ascending order (from the smallest to the biggest number).
4) All lines must be no longer than 240 characters followed by an <ENTER> for them to properly load into BASIC.

The Super "Color" Writer II is an excellent tool for changing variables, PRINT@'s, etc., by using the LOCATE AND CHANGE command. Refer to Section 3-D.

When you save a BASIC program, in the ASCII format, to tape or disk, the BASIC interpreter inserts a CARRIAGE RETURN as the first character to reside within the textfile. This CARRIAGE RETURN is unnecessary for BASIC loading and will create problems if you load the textfile to another computer system. Therefore, once you have loaded your BASIC program into the Super "Color" Writer II for editing, the first thing you should do is delete this CARRIAGE RETURN.

BASIC programs generated by other programs may sometimes have text that lies outside of line numbers causing "Direct statement in file" (DS) errors when loaded into BASIC. This problem can be corrected using the Super "Color" Writer II and proceeding as follows. Load the program into the Color Computer and LIST the program to locate the last line number loaded. The error will lie just after this line. Load the Super "Color" Writer II into the computer. Now load the program into the text buffer and scroll to the suspected line number. Edit out the erroneous text and resave the program to tape or diskette using the Super "Color" Writer II. If necessary, repeat the process to find additional errors.

APPENDIX C

COMMAND SUMMARIES

TEXT MODE COMMAND SUMMARY

(SEE THE HELP TABLE <CLEAR> <?>)

COMMAND	KEY(S)
BLOCK COPY (xerox)	<CLEAR>-<X>
BLOCK DELETE	<CLEAR>-<V>
BLOCK MOVE	<CLEAR>-<M>
COMMAND mode	<CLEAR>-<CLEAR>
CONTINUE LOCATE	<CLEAR>-<C>
Cursor down	<DOWN ARROW>
Cursor left	<LEFT ARROW>
Cursor left one word	<CLEAR>-<Q>
Cursor right	<RIGHT ARROW>
Cursor right one word	<CLEAR>-<W>
Cursor up	<UP ARROW>
Cursor to left of screen	<CLEAR>-<LEFT ARROW>
Cursor to right of screen	<CLEAR>-<RIGHT ARROW>
Cursor to top of textfile	<CLEAR>-<SHIFT><UP ARROW>
Cursor to end of textfile	<CLEAR>-<SHIFT><DOWN ARROW>
Cursor to home	<CLEAR>-<SHIFT><LEFT ARROW>
Cursor to bottom of screen	<CLEAR>-<SHIFT><RIGHT ARROW>
DELETE CHARACTER	<CLEAR>-<D>
DELETE TO END OF LINE	<CLEAR>-<S>
DELETE LINE	<CLEAR>-<F>
DELETE BLOCK MARKERS	<CLEAR>-<Y>
DELETE TO BEGINNING OF LINE	<CLEAR>-<A>
DELETE WORD LEFT	<CLEAR>-<H>
DELETE WORD RIGHT	<CLEAR>-<J>
FUNCTION E	<CLEAR>-<E>
FUNCTION R	<CLEAR>-<R>
FUNCTION T	<CLEAR>-<T>
HELP	<CLEAR>-<?>
INSERT character	<CLEAR>-<I>
INSERT line	<CLEAR>-<O>
LOCATE or CHANGE	<CLEAR>-<L>
MEM display memory LEFT, USED (GRANules disk only)	<CLEAR>-<;>
MASK case (during LOCATE only)	<CLEAR>-<->
Page back 14/19/22 lines	<CLEAR>-<UP ARROW>
Page ahead 14/19/22 lines	<CLEAR>-<DOWN ARROW>
PRINT textfile from cursor position	<CLEAR>-<P>-<P>
PROGRAMMABLE FUNCTION	<CLEAR>-<Z>
REPEAT last function	<SHIFT>-<@>
TAB 5,8,16,24,32,40,48,56,64,72	<SHIFT><LEFT ARROW>
VERIFY LOCATE	<CLEAR>-<ENTER>
WINDOW MODE	<CLEAR>-<P>-<W>

COMMAND MODE COMMAND SUMMARY

The following commands must be entered from the COMMAND mode. To enter to the COMMAND mode press <CLEAR>-<CLEAR>. To exit the COMMAND mode press <BREAK> or </>-<ENTER>.

COMMAND	KEY(S)
ALTERNATE MARKERS (LINK, FLUSH RIGHT/AUXILIARIES)	<A>-<ENTER>
CASSETTE LOAD	<C-L>-<ENTER>
CASSETTE NAME	<C-N>-<ENTER>
CASSETTE SAVE	<C-S>-<ENTER>
CHANGE DISPLAY COLOR	<C>-<ENTER>
CHARACTER SIZE IN 64 DISPLAY	<6-4-W/N>-<ENTER>
CLEAR TEXTFILE	<C-L-E-A-R>-<ENTER>
CLEAR TEXTFILE AFTER CURSOR	<A-F-T-E-R>-<ENTER>
CLEAR TEXTFILE BEFORE CURSOR	<B-E-F-O-R-E>-<ENTER>
DISKETTE DIRECTORY	<D-I>-<ENTER>
DISKETTE KILL	<D-K>-<ENTER>
DISKETTE LOAD	<D-L>-<ENTER>
DISKETTE LOAD (LINKING)	<D-L-;>-<ENTER>
DISKETTE NAME	<D-N>-<ENTER>
DISKETTE RENAME	<D-R>-<ENTER>
DISKETTE SAVE	<D-S>-<ENTER>
DISPLAY PRINT PARAMETERS	<.>-<ENTER>
DUMP HI-RES DISPLAYS FOR EXTRA MEMORY	<D-U-M-P>-<ENTER>
EXIT COMMAND MODE	</>-<ENTER>
EXIT TO BASIC	<E-N-D>-<ENTER>
INVERT DISPLAY	<I>-<ENTER>
OBTAIN WORD COUNT OF TEXTFILE	<W-O-R-D>-<ENTER>
"OOPS" BUFFER RECALL	<O-O-P-S>-<ENTER>
RESET DEFAULT FORMAT PARAMETERS	<@>-<ENTER>
RESET LINKING LIST	<;>-<ENTER>
SCREEN DISPLAY MODE SELECT	<32/51/64/85>-<ENTER>
TAB DEFAULTS (PROGRAMMABLE, see Section 3-D)	<T-A-B>-<ENTER>
TOGGLE BETWEEN SINGLE & TEN BLOCK MODE	<B-L-O-C-K>-<ENTER>
TOGGLE KEY BEEP	<B-E-E-P>-<ENTER>
TOGGLE WORD WRAP AROUND	<W-R-A-P>-<ENTER>
WIDTH OF EDIT WINDOW	<W-I-D-T-E> (#) <ENTER>

WINDOW MODE COMMAND SUMMARY

The FORMAT WINDOW is entered by pressing <CLEAR>-<P>-<W> and exited by pressing <E>. The following keys or combinations of keys perform the functions indicated in the FORMAT WINDOW (functions repeat if the key is held):

<UP ARROW>.....Scrolls up 12/17/20 lines (from mid-page).
Or scrolls to top of previous page (from top of page).
<DOWN ARROW>.....Scrolls down in the text file 12/17/20 lines.
<LEFT ARROW>.....Scrolls left in the text file 16 characters.
<RIGHT ARROW>.....Scrolls right in the text file 16 characters.
<SHIFT><UP ARROW>.....Displays the first page.
<SHIFT><DOWN ARROW>.....Displays the next page.
<SHIFT><LEFT ARROW>.....Displays left side of current page.
<ENTER>.....Displays carriage returns.

FORMAT PARAMETER COMMANDS

The following FORMAT parameters can be changed either by using the COMMAND mode or by using a FORMAT line. Their status can be displayed by pressing <CLEAR>-<.> from the COMMAND mode. The parameters may be reset to their default values by pressing <@>-<ENTER> in the COMMAND mode, or an <@> in a FORMAT line.

PARAMETER	SYNTAX	DEFAULT VALUE	LIMITS
AUXILIARY A LINE	AL	Ø	1-255 Ø=SUPPRESS
AUXILIARY B LINE	BL	Ø	1-255 Ø=SUPPRESS
AUXILIARY A STATUS	AS	A	A, E, or O
AUXILIARY B STATUS	BS	A	A, E, or O
BAUD RATE	BD	3	1=110,2=300,3=600 4=1200,5=2400,6=4800,7=9600
FOOTER LINE	FL	62	1-255 Ø=SUPPRESS
FOOTER STATUS	FS	A	A, E, or O
HEADER LINE	HL	4	1-255 Ø=SUPPRESS
HEADER STATUS	HS	A	A, E, or O
INVISIBLE PRINT	IP	N	Y or N
JUSTIFY	JU	Y	Y or N
LINE DELAY	LD	Ø	Ø-255
LINEFEEDS(after CR)	LF	N	Y or N
LINE SPACING	LS	1	1-255
MARGIN BOTTOM	BM	60	2-254
MARGIN LEFT	LM	8	Ø-255
MARGIN RIGHT	RM	72	10-255
MARGIN TOP	TM	6	1-253
NO-PRINT PRINT	NP	N	Y or N
NULLS	NL	Ø	Ø-255
PAGE LENGTH	PL	66	3-255
PAGE NUMBER	PN	1	1-65535
PAGE NUMBER LINE	PG	Ø	1-255 Ø=SUPPRESS
PAGE NUMBER PLACEMENT	PP	C	L, C, R or T
PRINT COMMENT	PC	N	Y or N
PRINTER HANDSHAKING	PH	Y	Y or N
SINGLE SHEET PAUSE	PA	N	Y or N

A = All; E = Even; O = Odd; Y = Yes; N = No
L = Left; C = Center; R = Right; T = Toggle

MARGINS FOR CENTERING

The proper margins to be used for CENTERING with each of the standard print fonts are as follows:

ELONGATED.....(5 C.P.I.).....LMØ,RM40
ELONGATED-COMPRESSED.....(8.3 C.P.I.).....LMØ,RM67
NORMAL.....(10 C.P.I.).....LM8,RM72
TWELVE PITCH.....(12 C.P.I.).....LMØ,RM96
FIFTEEN PITCH.....(15 C.P.I.).....LMØ,RM120
COMPRESSED.....(16.7 C.P.I.).....LMØ,RM132

APPENDIX D SUMMARY OF MARKERS

MARKER	KEY	DISPLAY*	DECIMAL (value of marker)	FUNCTION
AUXILIARY A	"<"	yellow 3/4	158	Marks line to be used as Auxiliary A on printed page.
AUXILIARY B	">"	red 3/4	190	Marks line to be used as Auxiliary B on printed page.
BLOCK (0-9)	<b>	orange	255	Marks block of text for block manipulation (2 req'd).
CENTER	<u>	blue	175	Marks line to be centered between left & right margins.
COMMENT	<+>	green diag.	137	Marks line to be masked during printing.
FLUSH RIGHT	"<"	red diag.	185	Marks line to be flushed to the right margin.**
FOOTER	<.>	red	191	Marks line to be used as Footer on printed page.
FORMAT	<k>	magenta	239	Marks line of format parameters to be altered.
FORMFEED	</>	cyan	223	Represents formfeed character used to bring paper to top.
HEADER	<,>	yellow	159	Marks line to be used as Header on a printed page.
LINK	">"	blue diag.	169	Marks line containing filename to be used during linking.**
NO PRINT	<n>	orange 3/4	254	Marks start and end of text not to be printed.
NONSPACE	<SPACE>	dark hue	128	Represents space that won't be broken during justification.
PAUSE PRINT	<=>	orange diag	249	Marks position for print pause (non-buffered printers ONLY).
P CODE	<0>	buff	207	Marks string of printer control codes (2 req'd).

* In hi-res displays the markers are inverse characters of the command key, except FLUSH RIGHT (inv. "f") and LINK (inv. "1").
** FLUSH RIGHT and LINK marker command keys alternate with AUXILIARY A and B command keys with <CLEAR>-<CLEAR>-<A>-<ENTER>

PROGRAMMABLE PRINTER CODE MARKERS

CODE	32 DISPLAY	HI-RES	DEC	HEX (of marker)	KEY(S)
CODE 1	GRAPHICS+1	INVERSE d+1	228	E4	<CLEAR>-<1>
CODE 2	GRAPHICS+2	INVERSE d+2	228	E4	<CLEAR>-<2>
CODE 3	GRAPHICS+3	INVERSE d+3	228	E4	<CLEAR>-<3>
CODE 4	GRAPHICS+4	INVERSE d+4	228	E4	<CLEAR>-<4>
CODE 5	GRAPHICS+5	INVERSE d+5	228	E4	<CLEAR>-<5>
CODE 6	GRAPHICS+6	INVERSE d+6	228	E4	<CLEAR>-<6>
CODE 7	GRAPHICS+7	INVERSE d+7	228	E4	<CLEAR>-<7>
CODE 8	GRAPHICS+8	INVERSE d+8	228	E4	<CLEAR>-<8>
CODE 9	GRAPHICS+9	INVERSE d+9	228	E4	<CLEAR>-<9>
CODE 10	GRAPHICS+:	INVERSE d+:	228	E4	<CLEAR>-<:>

APPENDIX E EXTENDED ASCII CHARACTER SET

The Super "Color" Writer II supports the full 256 character Extended ASCII Character Set. The following table covers the first 128 characters, the Standard ASCII Character Set (see Section 7-F). Characters in the second Extended ASCII Character Set (128-255) may be generated using PROGRAMMABLE CHARACTER CODES (see Section 7-I). PROGRAMMABLE CHARACTER CODES key sequences are initially defaulted to generate ten of the characters from 0 to 128 (see asterisked characters below). The display of the second set in the 32 display is the same as in BASIC. In the hi-res displays ONLY the display of characters 128-255 are the inverse display of the characters from 0-127.

ASCII	32X16	HI-RES	DEC	HEX	NOTES	KEY(S)
NULL	GRAPHICS	TRIPLE CROSS	0	0		PROG CHAR CODE ONLY
CTRL A	INVERSE !	±	1	1		<CLEAR>-<SHIFT><A>
CTRL B	INVERSE "	MOON	2	2		<CLEAR>-<SHIFT><B>
CTRL C	INVERSE #	LIGHTNING	3	3	<BREAK>	<CLEAR>-<SHIFT><C>
CTRL D	INVERSE $	PI	4	4		<CLEAR>-<SHIFT><D>
CTRL E	INVERSE %	SIGMA	5	5		<CLEAR>-<SHIFT><E>
CTRL F	INVERSE &	CHECK	6	6		<CLEAR>-<SHIFT><F>
CTRL G	INVERSE '	BELL	7	7		<CLEAR>-<SHIFT><G>
CTRL H	INVERSE (	←	8	8	<BACKSPC>	<CLEAR>-<SHIFT><H>
CTRL I	INVERSE)	→	9	9		<CLEAR>-<SHIFT><I>
CTRL J	INVERSE *	↓	10	A		<CLEAR>-<SHIFT><J>
CTRL K	INVERSE +	↑	11	B		<CLEAR>-<SHIFT><K>
CTRL L	INVERSE ,	↓↓	12	C	<FORMFD>@	<CLEAR>-<SHIFT><L>
CTRL M	<ENTER>	WEDGE	13	D	<CR>	<CLEAR>-<SHIFT><M>
CTRL N	INVERSE .	↑	14	E		<CLEAR>-<SHIFT><N>
CTRL O	INVERSE /	SECTION	15	F		<CLEAR>-<SHIFT><O>
CTRL P	INVERSE 0	ROOT	16	10		<CLEAR>-<SHIFT><P>
CTRL Q	INVERSE 1	REVERSE F	17	11		<CLEAR>-<SHIFT><Q>
CTRL R	INVERSE 2	EMPTY BOX	18	12		<CLEAR>-<SHIFT><R>
CTRL S	INVERSE 3	FULL BOX	19	13		<CLEAR>-<SHIFT><S>
CTRL T	INVERSE 4	LEFT T	20	14		<CLEAR>-<SHIFT><T>
CTRL U	INVERSE 5	RIGHT T	21	15		<CLEAR>-<SHIFT><U>
CTRL V	INVERSE 6	≠	22	16		<CLEAR>-<SHIFT><V>
CTRL W	INVERSE 7	SHIFT ←	23	17		<CLEAR>-<SHIFT><W>
CTRL X	INVERSE 8	SHIFT →	24	18		<CLEAR>-<SHIFT><X>
CTRL Y	INVERSE 9	SHIFT ↓	25	19		<CLEAR>-<SHIFT><Y>
CTRL Z	INVERSE :	SHIFT ↑	26	1A		<CLEAR>-<SHIFT><Z>
ESCAPE	INVERSE ;	∞	27	1B	<ESCAPE>	<CLEAR>-<@>
FS	INVERSE <	CENT	28	1C		<CLEAR>-<SHIFT><CLEAR>
GS	INVERSE =	REVERSE E	29	1D		*<CLEAR>-<SHIFT><1>
RS	INVERSE >	○	30	1E		*<CLEAR>-<SHIFT><2>
US	INVERSE ?	○	31	1F		*<CLEAR>-<SHIFT><3>
SPACE	SPACE	SPACE	32	20		<SPACE BAR>
!	!	!	33	21		<!>
"	"	"	34	22		<">
#	#	#	35	23		<#>

@ This is a firmware formfeed, usable like the FORMFEED described in Section 5-H, supported by most printers. It should ONLY be used by experienced persons.

APPENDIX B (continued)

ASCII	32X16	HI-RES	DEC	HEX	NOTES	KEY(S)
$	$	$	36	24		<$>
%	%	%	37	25		<%>
&	&	&	38	26		<&>
'	'	'	39	27		<'>
(	(	(	40	28		<(>
)	)	)	41	29		<)>
*	*	*	42	2A		<*>
+	+	+	43	2B		<+>
,	,	,	44	2C		<,>
-	-	-	45	2D		<->
.	.	.	46	2E		<.>
/	/	/	47	2F		</>
0	0	0	48	30		<0>
1	1	1	49	31		<1>
2	2	2	50	32		<2>
3	3	3	51	33		<3>
4	4	4	52	34		<4>
5	5	5	53	35		<5>
6	6	6	54	36		<6>
7	7	7	55	37		<7>
8	8	8	56	38		<8>
9	9	9	57	39		<9>
:	:	:	58	3A		<:>
;	;	;	59	3B		<;>
<	<	<	60	3C		"<"
=	=	=	61	3D		<=>
>	>	>	62	3E		">"
?	?	?	63	3F		<?>
@	@	@	64	40		<@>
A	A	A	65	41		<A>
B	B	B	66	42		<B>
C	C	C	67	43		<C>
D	D	D	68	44		<D>
E	E	E	69	45		<E>
F	F	F	70	46		<F>
G	G	G	71	47		<G>
H	H	H	72	48		<H>
I	I	I	73	49		<I>
J	J	J	74	4A		<J>
K	K	K	75	4B		<K>
L	L	L	76	4C		<L>
M	M	M	77	4D		<M>
N	N	N	78	4E		<N>
O	O	O	79	4F		<O>
P	P	P	80	50		<P>
Q	Q	Q	81	51		<Q>
R	R	R	82	52		<R>
S	S	S	83	53		<S>
T	T	T	84	54		<T>
U	U	U	85	55		<U>
V	V	V	86	56		<V>
W	W	W	87	57		<W>
X	X	X	88	58		<X>
Y	Y	Y	89	59		<Y>

ASCII	32X16	HI RES	DEC	HEX	NOTES	KEY(S)
Z	Z	Z	90	5A		<Z>
[	[	[	91	5B		<SHIFT><DOWN ARROW>
\	\	\	92	5C		<SHIFT><CLEAR>
]	]	]	93	5D		<SHIFT><RIGHT ARROW>
^	↑	^	94	5E		*<CLEAR>-<SHIFT><4>
_	←	_	95	5F		<SHIFT><UP ARROW>
`	INVERSE @	`	96	60		*<CLEAR>-<SHIFT><5>
a	a	a	97	61		<a>
b	b	b	98	62		<b>
c	c	c	99	63		<c>
d	d	d	100	64		<d>
e	e	e	101	65		<e>
f	f	f	102	66		<f>
g	g	g	103	67		<g>
h	h	h	104	68		<h>
i	i	i	105	69		<i>
j	j	j	106	6A		<j>
k	k	k	107	6B		<k>
l	l	l	108	6C		<l>
m	m	m	109	6D		<m>
n	n	n	110	6E		<n>
o	o	o	111	6F		<o>
p	p	p	112	70		<p>
q	q	q	113	71		<q>
r	r	r	114	72		<r>
s	s	s	115	73		<s>
t	t	t	116	74		<t>
u	u	u	117	75		<u>
v	v	v	118	76		<v>
w	w	w	119	77		<w>
x	x	x	120	78		<x>
y	y	y	121	79		<y>
z	z	z	122	7A		<z>
{	INVERSE [	{	123	7B		*<CLEAR>-<SHIFT><6>
|	INVERSE \	|	124	7C		*<CLEAR>-<SHIFT><7>
}	INVERSE]	}	125	7D		*<CLEAR>-<SHIFT><8>
~	INVERSE ↑	~	126	7E		*<CLEAR>-<SHIFT><9>
Rubout	INVERSE ←	▒	127	7F		*<CLEAR>-<SHIFT><:>

DMP-200 PRINTER DRIVER SYSTEM DEFAULTS

FUNCTION	DECIMAL	HEX	KEY(S)
Bold on	27 31	1B 1F	<CLEAR>-<1>
Bold off	27 32	1B 20	<CLEAR>-<2>
Underline on	15	0F	<CLEAR>-<3>
Underline off	14	0E	<CLEAR>-<4>
Half rev linefeed	27 30	1B 1E	<CLEAR>-<5>
Half for linefeed	27 28	1B 1C	<CLEAR>-<6>
Select condensed	27 20	1B 14	<CLEAR>-<7>
Select standard	27 19	1B 13	<CLEAR>-<8>
Elongation on	27 14	1B 0E	<CLEAR>-<9>
Elongation off	27 15	1B 0F	<CLEAR>-<:>

APPENDIX F

64K AND THE SUPER "COLOR" LIBRARY - WITHOUT USING FLEX

All Super "Color" Library programs have MEMORY SENSE to automatically sense the amount of memory that is available with your system and adjust to allow you to make the fullest use of that memory.

You may obtain 64K with your Color Computer either by modifying your computer, or by purchasing the 64K version of the Color Computer in the first place. The modifications necessary to obtain 64K have been ably discussed in the March, 1983 issue of Rainbow and in the "64K Corner" columns in the January 1983 and February 1982 (reprinted in the January 1983 issue) of Color Computer News. The modifications suggested require some technical knowledge and experience. We recommend that you refer to these technical articles and the references therein for modification information. The authors of those articles have obvious expertise about such modifications and should be happy to answer any questions you have.

APPENDIX G

WHAT TO DO IF YOUR TV DISPLAY IS HARD TO READ

After loading your Super "Color" Library program and using it for a while you may be dissatisfied with the screen display when you are using the 51, 64, or 85 displays. You may notice that when you are using color you have a rainbow of colors on the screen instead of the background color of green or white that you wanted, making it nearly impossible to read your text. This inability to have a sharp, clear, crisp display is NOT a problem with the program, it is a problem of the Color Computer and your particular TV.

Unfortunately, color TVs were not made to be used with computers. Color pictures are made from a composite of blue, green and red. Although they may be easily controlled by the scanning techniques used to generate TV shows, that scanning technique is not adequate to control the screen the way the computer does by controlling individual spots (or pixels) on the screen. Thus, when the background color is not a pure color, red, green or blue, the color cannot be controlled to be pure. Shadows and blooms of other colors mix in. This causes the black letters to sometimes be blotched or to have shadows so that they are difficult to read. The smaller the letters the greater the problem. Thus, the 85 display is sometimes very hard to read.

Of course the 85 display, and to some extent the 64 display, were not designed for entering and editing text unless you use a monitor. They were designed for formatting purposes. If you can use them to input text, so much the better.

There are some things that can be done to help make the displays more readable. First, you can use the green background to avoid the problems associated with color mixing. You could also get a color monitor for better control of your display colors. Alternatively, you can use a black and white TV instead of a color TV. If all else fails, the 32 by 16 display will definitely work with your color TV.

Although many find the 32 by 16 display too small, others like it because it is easier to read, and since the lines are short like those of a newspaper, editing is faster. Of course, the 32 by 16 retains the reverse video display used by the Color Computer for displaying lowercase characters, and you may find this unsatisfactory. There is, however, a fix for this. The LCA-47 Lowercase Adapter sold by Micro Technical Products, Inc., Mesa, Arizona will give you true lowercase with the 32 by 16 mode.

The final option is to buy a monitor, a special kind of video machine, for displaying your text. Since the Color Computer does not allow a direct connection to a monitor (only color TVs), you will have to purchase and install an adaptor. Although we cannot tell you how to do this, we can recommend a company that can. World Electronics, 177-27th Street, Brooklyn, New York 11232, sells a monitor adaptor called the "TV Buff" which will do the trick.

APPENDIX B

HOW TO USE OTHER SUPER "COLOR" LIBRARY PROGRAMS

Each of the programs in the Super "Color" Library, with the exception of Super "Color" Speller and Super "Color" Disk-ZAP, were specifically designed to create files compatible with other programs in the Library. With the Library you can perform the essential home business tasks and combine the results for many purposes.

The Super "Color" Writer II is one of the central programs in the Library. It contains the most sophisticated editing and printing features, and it is to be used to create all reports combining files created on other applicable Library programs. Around it are grouped two other dependent programs: Super "Color" Speller and Super "Color" Mailer. The Speller can be used to correct typos and misspellings in Super "Color" Writer II files. The Super "Color" Mailer uses Super "Color" Writer II files as a base for sending out mailings and merging documents.

Super "Color" Calc is used to create financial or mathematical reports. It contains sophisticated print functions for independent printing of such reports. You may create files usable by the Super "Color" Writer II for reports to be combined with other text, and you may create templates with the Super "Color" Writer II for use in Super "Color" Calc.

Super "Color" Terminal is a communications program capable of transmitting and receiving any ASCII file, including Super "Color" Library files. ASCII files can be transferred to the Super "Color" Writer II for further editing. The Terminal program also allows you to transfer files to work, clubs or friends. You can also print files received from others.

Super "Color" Database, similar to Super "Color" Calc has its own sophisticated print functions for independent printing of database files. You can also create files for use with the Super "Color" Writer II to create combined text and database files.

Super "Color" Disk-ZAP is a disk repair utility designed to repair any kind of file created using the Color Computer disk operating system. Of course, it therefore will also work on other Library files.

GLOSSARY OF TERMS

ASCII: An acronym for American Standard Code for Information Interchange. It is a standard set of characters and their numeric equivalents used by most systems for representing, storing and transmitting text.

AUXILIARY LINE: A line of text that can be placed anywhere on the printed page and which may be set to appear on every page, just even-numbered pages or just odd-numbered pages.

BAUD RATE: The speed at which data is transferred from the computer to other peripherals such as printers and modems (see your printer manual).

BINARY: The base 2 number system used to signify the states of computer memory circuits.

BINARY FILE: A file in binary machine language code. The Super "Color" Writer II will not load binary files.

BLOCK: A segment or string of text of any size.

BOTTOM MARGIN: The number of lines from the top of the page to the last line of printed text.

BUFFER: Area set aside in computer memory for text storage.

BYTE: A single unit of computer memory which is synonomous with a single character.

CARRIAGE RETURN (CR): End of line terminator marker created by pressing <ENTER>.

CASE: A term which refers to the type of character generated (uppercase or lowercase).

CHARACTER: Single unit of text, e.g., number, letter, punctuation, or space.

CHARACTER CODES: Key sequences programmed to produce up to ten characters each for display on the screen and input into the buffer for printing.

CLEAR: The process of erasing the buffer contents.

COMMAND MODE: The mode from which specific commands are executed, entered by pressing <CLEAR>-<CLEAR>.

COMPRESSED FONT: A print font allowed by dot matrix printers which allows 16.7 characters to be printed per inch, versus the 10 characters per inch in the normal mode.

CONTROL CODE: See Printer Control Code.

CONTROL KEY: The <CLEAR> key used to access commands in the TEXT mode.

C.P.I.: An abbreviation for Characters Per Inch.

CURSOR: The blinking block indicating your current position.

CURSOR ORIENTED: A term which refers to the system characteristic which requires that all functions be implemented from the position of the cursor.

DECIMAL: The numeric value of an ASCII symbol, expressed in the base 10 number system, ranging from 0 to 255.

DEFAULT VALUE: A value automatically assigned to a parameter, printer code or function at the start of the program.

DELIMITER: A character which serves as a signal to the Super "Color" Writer II to treat that which precedes the character as one unit and to begin to look for another unit (i.e., commas in format lines).

DOUBLE STRIKE: A printer function which causes the printer to emphasize a string of text by printing the same string over itself a second time.

DOUBLE WIDE: A print font used in dot matrix printers to print characters at 5 c.p.i., double the normal font width.

DRIVER CODES: Key sequences programmed in the printer driver to represent up to ten printer control codes.

EDIT WINDOW: A term for the screen display of the textfile when in the TEXT mode. The system allows you to edit line lengths longer than those which can fit on your screen, so the screen is like a window into your textfile.

ELONGATED: The same as "double wide".

ESCAPE: A printer control code, generated by pressing <CLEAR>-<@>, which some printers require to precede certain control codes to tell the printer that the next code is a control code.

FLUSH RIGHT: The process of pushing a string of printed text to the right margin.

FOOTER: A line of text which may be set to appear on each printed page, or on odd or even pages, usually at the bottom of the page.

FORM FEED: A character which causes the printer to feed the paper to the top of the next page.

FORMAT WINDOW: The mode used to view the textfile finally formatted for printing, checking for format errors and setting values for the character and printer driver codes.

FORMATTING: The process of arranging text to achieve a desired result when printed.

GLOBAL: A term which refers to the ability to perform a function on every occurrence of a string anywhere in the textfile, such as global LOCATE.

GRANULE (GRAN): A unit of disk storage equal to 2304 bytes.

HANDSHAKING: A term referring to the interchange of communication signals between printer and computer to control the flow of information. Terminals often do not exchange signals with the computer to control data transfer, and thus do not use handshaking.

HEADER: A line of text which may be set to appear on each printed page or on odd or even pages, usually at the top of the page.

HEXADECIMAL EQUIVALENT: The numeric equivalent of an ASCII symbol expressed in the base 16 number system.

HI-RES: A term referring to the software-generated lowercase displays generically, including the 51, 64 and 85 displays.

HOME: The upper left-hand corner of the video screen.

INSERT: The process of opening up text at the cursor position and adding a character(s).

INVERSE: A term which describes the display of a character on the screen as the opposite of the prevailing display, i.e., light character on a dark background where the rest of the screen has dark characters on a light background.

INVISIBLE PRINT: A printing and formatting function which causes the system to process the textfile without sending it to the printer from where this feature is turned on to where it is turned off. Printing will be resumed where this feature is turned off, with the appropriate page number, etc.

JUSTIFICATION: The process of adding spaces to the printed text to create an even right hand margin.

LEFT MARGIN: The number of spaces from the left lock of the printer to the first character of the line.

LINE: When referring to text to be printed, it means a line of text as printed according to your print margins. When referring to text displayed on the screen it means the text on one line on the screen, whether or not terminated by a carriage return.

LINE DELAY: A time delay required by some printers to allow the carriage time to return to the beginning of the line (see your printer manual).

LINEFEEDS: Characters required by some printers to advance paper one line. (See your printer manual.)

LINE LENGTH: The total difference between the left and right margins.

LINE SPACING: The number of blank lines between each line of text.

LINKING LIST: A list, created by the user and resident in the buffer when linking, which is comprised of the names of all disk files to be linked.

MARKER: A character which sets aside a string of text or a character for a particular function.

MASK: A function of the system which allows the Super "Color" Writer II to disregard case during LOCATE.

MEMORY SENSE: A feature which allows the system to automatically sense and use the maximum amount of memory available with different versions of the Color Computer.

MONITOR: A cathode ray tube device designed specifically for optimum display of high resolution graphics.

NONSPACE: A marker which binds text together and which is treated like a space by the printer.

NO-PRINT PRINT: A formatting feature which allows the system to print text blocked off from printing by NO-PRINT markers.

NULLS: Empty characters required by SOME PRINTERS to be added after lines to give the printer time to return the carriage (see your printer manual).

NUMERIC EQUIVALENT: The number assigned to a specific ASCII symbol, expressed either in decimal or hexadecimal, which is generated by the key sequences set out in APPENDIX E (see your printer manual).

ORPHAN LINE: A line which has been separated during printing from other lines of which it is an integral part, such as when one line of a paragraph is printed on one page and the rest on the next page.

P CODE MARKER: A marker used in pairs to set aside control codes to be sent to the printer.

PAGE LENGTH: The total number of lines from the top of the page to the bottom of the page.

PAGING: The function of moving up or down in the textfile one full screen at a time.

PARAMETER: A variable element of page formatting (e.g., page length, margins, line spacing, etc.), each of which controls a specific aspect of formatting and which may be altered through the COMMAND mode or a FORMAT line.

PAUSE PRINT: A function which allows non-buffered printers to be stopped to change thimbles, etc.

PERPETUAL INSERT: The system characteristic of allowing characters or lines, including carriage returns, to be inserted until the INSERT feature is turned off when no longer desired.

PITCH: The number of characters printed per inch by your printer, e.g., a 10 pitch printer prints 10 characters per inch.

PIXEL: A distinct, definable graphics unit used by the system in the 32 display to represent function markers, the cursor, the carriage return, etc.

PRINT COMMENTS: A format parameter which allows normally non-printable comments to be printed along with the rest of the textfile.

PRINT FONT: The style of typeface used by the printer, e.g., elongated, compressed, or italics (see your printer manual).

PRINT MODE: A term referring to special features available with some printers which allow such things as alteration of line spacing, dot matrix size, etc.

PRINTER CONTROL CODES: Codes which control the printer's print fonts and print modes which codes are expressed in ASCII equivalents and which are usually defined differently by different printer manufacturers (see your printer manual and APPENDIX E).

GLOSSARY OF TERMS

PRINTER DRIVER: A customized reusable textfile of printer control code equivalents for the commands from <CLEAR>-<1> through <CLEAR>-<9> and <CLEAR>-<:> created by the user to match his or her printer and special needs.

PRINTER HANDSHAKING: See HANDSHAKING.

PROGRAMMABLE CHARACTER CODES: User definable character or printer control codes used to control features or special characters available with printers (see your printer manual).

RIGHT MARGIN: The number of spaces from the left lock of the printer to the last character on the line.

SCROLL: The process of moving in the textfile up, down, left or right a specified number of lines or columns at one time.

SINGLE SHEET PAUSE: A function which allows printing to be stopped at the end of a page so that a new sheet of paper may be added.

STATUS: The current setting of format parameters.

STRING: Any number of characters in a row, usually terminated by <ENTER>.

STRING INSERT: A programmable command with which a string of text, such as a name, can be inserted at one time.

SYSTEM: A term which refers to the Super "Color" Writer II program itself.

SYSTEM LINE COUNTER: A line counter internal to the system which keeps count of the number of lines on a page and the number that have been printed. This line counter is reset to zero whenever the <BREAK> key is used to halt printing.

TAB: Preset spacing for indentation.

TEXT BUFFER: The work area for the textfile.

TOGGLE: To switch a mode or status on or off.

TOP MARGIN: The number of lines from the top of the page to the first line of printed text.

TYPE-AHEAD: A feature which allows characters input to be stored for display, used when the system is engaged in another task or when the display is slowed down by other tasks such as word wrap around or line insert.

GLOSSARY OF TERMS

TYPE OVER: The ability to retype text over previously entered text.

UNDO: To reverse the effects of a delete, line insert or type over action.

VERIFY: A feature used during the LOCATE AND CHANGE or DELETE command to allow sequential decisions about whether to change or delete a string of text.

WILD CARD: The term for a character which represents a surrogate or "don't care" character during LOCATE.

WINDOWING: The process of using the FORMAT WINDOW to view text; with PROGRAMMABLE CHARACTER CODES, to go into the FORMAT WINDOW to set the values of the newly programmed codes (in a FORMAT line) to allow their implementation in the textfile.

WORD WRAP AROUND: The ability of the system to move the entire word to the next line instead of splitting it up at the end of a line on the screen.

I N D E X

-A-

-B-

-C-

-R-

-S-

-T-