

# small computer support group

---

Indiana University Bloomington, 1125 Atwater, 337-2893

---

-----

NEWSLETTER. NO. 4. JULY 1981.

-----

Editor: J.V. Lombardi, Bryan 205, IUB, 7-8669

Manager SCSG: J.W. Smith, 112 Atwater, 7-2983

## CONTENTS

In Search of the Perfect Apple II Word Processor

PIE + FORMAT

SUPER TEXT II

The Exidy Sorcerer Computer: Part I

Applesoft Compiler Speed Comparisons

Update on The Brain Surgeon

Bloomington Apple Users Group

Data-Base Management Systems

The Buffered Modem

Local Bulletin Boards

\* \* \* \* \*



## IN SEARCH OF THE PERFECT APPLE II WORD PROCESSOR

In the search for the perfect word processor, many programs are called but few can be chosen. In the effort to make the Apple II do what it was not designed to do, that is word processing, compromises are necessary, and these compromises are what keeps the many excellent word processors on the market from being declared the perfect word processor.

For those unfamiliar with word processing and the Apple II environment, a few comments about the features of the perfect program may prove helpful. In general, there are two kinds of word processors. One, frequently called "what you see is what you get," formats the text on the screen exactly as it will be sent to the printer. The second variety permits you to enter the text in free form on the Apple screen. The job of organizing the text to look the way you want it is done by another program that formats the text in accord with commands you define. This second type of program is called a text editor with a text formatter.

Normally, the Apple II has only forty columns of text on its screen. It can only display upper case letters, and some useful symbols can not be shown on the screen. Each program that attempts to do word processing must invent ways to cure these ills. Some programs require hardware changes or additions such as a \$300 accessory card that puts 80 columns on the Apple screen and uses a reasonably high quality monitor that costs \$200. This is an expensive solution to the problem.

Most Apple word processors can use less expensive adapters that put 40 columns of upper and lower case characters on a TV screen. This solution is in the range of \$60 and is quite practical. Many programs, in addition, will work without any modification, displaying upper case in inverse video (black on white) and lower case in regular (white on black). This also works ok, although it is somewhat more difficult to proofread accurately on the screen. Finally, most of the word processors that use an upper and lower case adapter (either 80 or 40 columns) allow a simple one-wire modification of the Apple that will make the shift keys work as they do on a typewriter.

As if these variations were not enough, some word processing programs enter text a line at a time and require that each line be treated as a unit for editing, although the formatting of the final document will ignore these lines. This is called a line oriented text editor (or word processor). Others use a character at a time method in



which the basic unit for editing and text entry is a single character. This is a character oriented word processor (or text editor).

What you see on the screen of a word processor is usually called a window, and each word processing program has its own special features related to that window. In solving the 40 column dilemma, some word processors move the window along an 80 column line to give you a 40 column picture of the text being entered. Others do not try to represent the 80 columns on the screen. Some have a preview mode that allows you to move the window around a formatted page to check on where the line, paragraph, table, and other items of the text are placed. Of course, the 80 column versions do not have to worry about this because the entire 80 characters are on the screen. Additionally, the programs must do something when the text being entered on one line runs into the right-hand margin. Most, but not all, will move the incomplete word down to the next line, breaking at the last space. This makes for an easy-to-read display. A few of the simpler, and cheaper, programs will break words in the middle if they run into the right edge of the screen.

All word processors have facilities to move the cursor (the blinking square or other character) that marks the point where text can be entered. The cursor can be put at the beginning, the end, or anywhere else. But not all programs do this operation easily or quickly. Special control characters must be pressed to accomplish these moves, and part of the trouble of learning to use one of these programs is memorizing the special key sequences that move the cursor around.

It is always necessary to delete or insert characters and lines, move blocks of lines, delete blocks of lines, and manipulate the text in a variety of ways. The more sophisticated the editor, the more variations and special features for moving and managing text are available. Every program needs to be able to search for a word or string of characters and then move the cursor to those characters. The better programs will also do a search and replace that finds one set of characters and replaces them with another set.

Word processors should produce multiple copies and form letters. Most will, but they do not all do them with equal flexibility. These tasks often require the programs to read information from one file and insert it into the text prepared on another file, thus permitting customized form letters.

The appearance of the output offers an almost infinite number of possibilities. Some documents require page



numbers at the top, at the bottom, at the left or middle or right side. Some require wide margins, some narrow margins. Some want block paragraphs, some want indented paragraphs. The document may need double and single spacing within the same page, hanging indents to handle quoted material, underlining, centered titles, and the rest. It is in this area that the word processors differ most. While all will put the text on the page neatly, with the correct number of lines per page and a page number, the rest of the variations in output format depend on the sophistication of the program. Often, in the more complicated programs, it will take a user quite a lot of trial and error to discover all the possibilities.

Word processors will justify the text, that is line up the right-hand margin. The fancy programs will permit either left or right or both margins justified. However, many programs will not justify text that has special characters imbedded in it. For example, much foreign language text requires strike-over accents, umlauts, or other special marks. If the printer will do these things it usually does them by backspacing and then striking over the original letter. That means a backspace character has to be inserted in the text after the letter and before the accent mark. This confuses word processing programs that count the backspace as a character and the accent as a character. When the line is justified it doesn't look right because the program thinks it is two spaces longer than it is. This line is an example of the problem with the Spanish name Gómez. As you can see it doesn't come out right. The same will happen with other control characters that give superscripts, subscripts, and the like. However, since the vogue these days appears to be for unjustified text, this problem may not be significant.

Finally, the choice of any word processing program is a very personal thing. What works well for one individual will be an aggravation for another. It is essential to borrow a word processor and play with it for a while before making a decision. Once a word processing program is purchased and learned, it is not easy to change over to another one. This is so because the conventions and rules for the use of these programs are very different. It is also important to consider what kind of work the program will be asked to do. What is ideal for a business, sending out short form letters to a large mailing list, may not be suitable for the book author who is composing four hundred page manuscripts. If the program will be used by someone interested in computing who likes to play with the machine, a complex program may be appropriate. But for someone who wants to learn as little as possible about computing to do word processing, a simpler or at least foolproof program may be better.



There are, however, some technical details that may be of interest to everyone. These programs have two characteristics that may make a difference to a user. These are first, whether or not the program can be copied, and second, how the text is stored and recalled.

Because many programmers and the companies that sell the programs have become somewhat hysterical about the widespread pirating of computer programs, many of them have gone to extraordinary lengths to make their products difficult to duplicate. While this may be good for business, it can be bad for the user. The disks and the programs themselves are fragile and can be destroyed relatively easily. Some companies provide back-up disks with the program, some will send copies when you return the damaged original, and some have provisions for partial copies or a limited number of copies. A few do not worry about it and provide completely copyable programs. All include elaborate statements about the illegality of pirated programs, and some want the user to sign licensing agreements that limit the use of the program to one person, one machine, or some similar arrangement. All things being equal, users should prefer programs that can be copied. Unfortunately, all things are never equal with word processing programs.

The text that is entered into a word processor must be saved to a diskette so that it can be recalled later for editing, additions, or printing. Some programs use special processes to put the text on diskette and then restore it to the machine later on. Others use standard Apple disk operations to achieve this. In general, it is preferable to use standard disk operations, for then the word processor can read files that have been created using other programs. If the program uses standard files, it should be able to use either binary files or text files, and most do this. If a user will only need word processing and will never use any other programs to create text or other material, then the form of the files may not be a major consideration.

( Note: The following reviews were composed and printed with the word processors they review.)

John V. Lombardi  
Bryan Hall 205  
337-8669

\*\*\*\*\*



## THE WORD PROCESSING SYSTEM PIE+FORMAT

Now for a description of one of the more interesting word processing packages for the Apple II, the Word Processing System from Programma Software Products (Programma International, Inc., 3400 Wilshire Blvd., Los Angeles, CA 90010. Available from Data Domain in Bloomington for \$129.)

This word processing package belongs to the text editor, text formatting program variety. It consists of the editor, Programma Improved Editor or PIE, and the text formatter, FORMAT. All of the programs, PIE and FORMAT and a variety of utilities to make them work with different systems, are copyable with normal copy programs. The manual wisely recommends making two copies of the original disk before any work at all is attempted. This is good advice.

The package comes in special versions for Apples that have 80 column cards or 40 column displays. This review refers entirely to the 40 column version with the Dan Paymar adapter installed for upper-lower case display on the screen, and with the shift key modification that makes the Apple shift key work like a typewriter shift key.

### PIE+FORMAT: A Summary

The PIE+FORMAT word processing system is clearly one of the more sophisticated and flexible Apple word processors available. It has a number of interesting features that may make a user love it or hate it. It is not a program for those who like simple, easy to operate machinery. But if complicated, sophisticated processes intrigue you, then this may be the best of all the programs available.

First, the manual. For many users, it is the manual that makes or breaks the program. Programma has provided this program with a manual designed for the dedicated computer nut. It is very nicely presented, reasonably well written, and mostly complete. But someone decided that because this word processing system is so good it deserves a special language to describe its features. Hence, the manual frequently invents new words or abbreviations to describe ordinary Apple keys. Keys are renamed to such things as APP (ctrl-b) or ARG (esc) or CINS (ctrl-p) or GOB (ctrl-j). Now, these clever names are perfectly logical, CINS means "character-insert" for example. But the user reading the manual has to remember first that CINS means "character insert" and then that CINS also means "ctrl-p." The package comes with some nice little labels that can be stuck on the keys to help with this identification, but of course for the touch typist that is only of marginal help.



This peculiarity of the manual is significant only if you are in a hurry to explore all of the program's features. With time and experience, the conventions of PIE can be mastered and become automatic. But it does take some doing.

The second feature of the manual is that it has no index. It has lots of tables and lists that help, but no index. It has a good quick reference card, but the card and the similar lists refer back to the manual only by section number, and there are some typos on the card. You need to be prepared to read the manual many times and be ready to leaf through it often to find what you remember you read somewhere.

These minor inconveniences aside, the manual is comprehensive and clear. Things mostly work just like the manual says they do with practically no surprises. This is a major accomplishment.

#### The Programma Improved Editor: PIE

The PIE editor is a fine text editor, the equal of many large computer text editors. It has a host of options that make it possible to edit anything that can be put in the Apple's memory. It reads binary files, text files, and Basic programs captured in text files. It can take files with line lengths that are too long, cut them up for editing, and paste them back together for printing. It can move blocks of text around and put them anyplace else in the file. It can delete, insert, and move text. Like any first-class text editor, it will search and find any string of characters, and if you want, it will replace that string with any other string. It does these search and replace missions with great speed. Control characters can be inserted into the file, and they can be seen as flashing, inverted letters. Pages can be scrolled up and down in a variety of ways. The cursor can be jumped instantaneously from anyplace in the file to anyplace else in the file. All of this is accomplished through sequences of control characters whose functions and options are fully described in the manual and summarized on the reference card.

In addition to these editing functions, PIE also offers a variety of text entry modes. You can enter text as if it were one long string of characters. In this mode, the text is broken at the end of each line at a space and the next word is wrapped around to start on the next line. This is all done automatically, so fast that no characters are lost. A second mode expands the text line from 38 characters to 64 characters. As you type along and get to character 38 the screen shifts to the right and gives you the last 12 characters plus another 26 on the second



screen. When you get to the end of the 64 characters, the program beeps and won't let you enter any more text until you press return to start a new line. When you press return, the screen shifts left so that you are once again at the left-hand margin. This sounds complicated, but in practice it works well. Of course, text entered in this fashion is somewhat difficult to scan, but it can be managed with practice.

In either mode, the program allows any number of tab stops. It also lets you put a bell at any column to signal the impending end of line in the 64 character/line mode. There is also an automatic indent mode that is useful in making outlines or tables. When in this mode, a return will put the cursor right under the first character of the last line typed. This makes it easy to line up table or outline entries.

This summary of editing functions is somewhat incomplete because each of the basic functions of inserting, deleting, moving, and the like have options that extend their usefulness for special editing tasks. The manual explains them well.

PIE also has a large number of file handling features. You can load or save binary or text files. You can do this using any names that seem appropriate. You can ask for the length of the file in words or characters. You can save or load selected line numbers from or to a file. You can output a file with line numbers printed alongside the lines. You can append data in memory to data on a file. You can append data on a file to data in memory. It is possible to load a file with lines that are too long and slice them up into shorter lines for editing. These lines can be sliced to be rejoined later, or broken permanently. You can also load and save to cassette tape. Moreover, all of these things can be done either to the disk file or to any other Apple output port. That means you can send the file to a printer, a modem, or perhaps an intelligent plotter. Or you can send the file from memory to the screen.

Given the complexity and variety of these possibilities, PIE has a procedure to put a sequence of commands into a sort of mini-program that can then be executed. This would allow someone to set up a complicated sequence of file manipulations to be run automatically every time the mini-program is called. Finally, the program allows disk commands without leaving PIE. You can delete, verify, rename, lock, unlock, and exec files, as well as catalog the disk from within the program. While the program alerts you when you are approaching the limits of the computer's memory, it does not prevent you from entering text beyond the memory limits. If you are absent minded and



type more than a page or two after the warning, your file may not save properly and will not load properly.

As is evident from this description, the editor is very powerful. There are a number of other exotic possibilities also available that require some knowledge of the internal workings of the Apple. It is possible to redefine the entire keyboard so that the keys produce different things than they normally do. It is possible to invent machine language routines and call them from the editor. But it does not seem to be possible to create macros or groups of characters that can be entered into the file by typing one key. This feature would be useful for people who use foreign language accents and other symbols that require an overstrike. The sequence of backspace+accent requires four keys in PIE. A macro facility would allow a one character entry of this sequence.

Such is the PIE editor in general terms. Powerful, complicated, and flexible, it is a superior product.

#### The Text Formatter: FORMAT

If the Programma Word Processing System has an impressive editor, its text formatting program is also sophisticated. There is a separate manual for this program that follows much the same style as the PIE manual, although there is no confusing special language to cope with.

FORMAT takes the text prepared with PIE and rearranges it to look good on the page. The options for this process are many. Some are obvious such as the specification of the page length, the spacing between lines, the length of the line, the start of a paragraph, the ability to underline and center, and the choice of justified or un-justified text. But because this is a sophisticated program, it can also do some other interesting things. For example, the program will guarantee any number of lines will be on the next page. This avoids having a new page start only to print one line. Text can be printed by asking FORMAT to fill out the lines from words in the input file, or by asking FORMAT to print the lines as they appear in the input file. In addition, the lines can be printed with a justified right margin. While the program does normal paragraphing, it also will allow variable indentations, guarantee a number of lines of a new paragraph on a new page, and the like.

This formatter handles centering and underlining with considerable flexibility. But because it does the underlining by printing backspaces, you must have a printer capable of backspacing to use the underline feature. Some formatters, but not this one, will underline by first



printing the text line and then printing the underline line without advancing the paper. This allows underlining with less complex printers.

In addition to these features, the formatter will handle a variety of headers and footers. The program allows three fields for headers and three for footers. At the top or bottom of the page the program will print a left justified, a centered, a right justified, or any combination of header message. The page number can be imbedded into any of these headers or footers. Moreover, the program will allow a variable number of blank lines before and after the header or footer.

Because FORMAT is designed to be used in form letter applications, it has a series of commands for input file switching. This allows the printing of essentially the same text with the insertion of material from another file such as names and addresses. It is also possible to insert material into the file as it is being printed. The program will stop at the appropriate spot, the operator enters text from the keyboard, and then this is printed where it belongs as the program continues on printing the document. It is also possible to stop the printing at any point to adjust the printer. You can even request that everything be printed in upper case.

The only serious drawback to this formatting program is its inability to handle strikeouts or other special control character printer commands at the same time it justifies lines. An example of this problem follows for an accent overstrike and a subscript number done with a negative half-line feed.

This is an example of some text that uses the justify feature and the control characters necessary for superscripts and subscripts or strikeouts. First the example of Juan Vicente Gómez. Second, I will try to get some text in here that will show the superscripts and subscripts such as  $H_2$  or some similar complication.

All of these options can be started or stopped any number of times in the same document. The options are selected and defined through the insertion of codes when the file is created. Each option is chosen by a command that is placed by itself on a line starting with a period and followed by a two-letter command. Some of the commands also have a number that tells how many lines, how many spaces, and the like. The commands work quite well as should be evident from this review which uses a variety of them.



Documents longer than the memory space available in the Apple (about 20,000 characters or about 10 typewritten, double-spaced pages) can be printed and formatted in sections. At the end of the first part, a command is included that chains immediately to the second part and so forth for any number of files. Of course the documents involved must be on the disk in the drive. But by combining some of these commands it is possible to have the program stop and ask you to change a disk and then chain to another set of files on another disk. The program does not, however, permit footnotes nor does it do any arithmetic. It will issue messages to the screen requesting special type or different paper.

This formatter is quite fast, although I don't know whether it will drive a high speed printer to capacity. It has no difficulty making my Diablo 630 run at maximum, but that is a relatively slow printer.

FORMAT, like its companion program PIE, takes some getting used to. Although this manual is easier to work with, it too lacks an index. Because it is a very flexible program, users need to experiment a lot to be sure they understand all of its potential.

#### PIE + FORMAT: Some General Comments

The Programma Word Processing System is a fine product. It is well designed, flexible, powerful, and easy to use once the initial barrier of the manual's language is surmounted. Because the programs are designed to be used by people with a wide variety of needs, it is an open program that allows many modifications and changes. This has the advantage of being useful in non-standard text processing or program editing tasks, but it also means that there is ample opportunity to damage your text. Because it is easy to destroy files on the disk in the drive, you should be certain all files that you do not want to erase are locked. The powerful commands for moving files in and out of the system and changing them in subtle and not so subtle ways also has the potential for turning your files into junk.

This program is not at all hostile to the user, but it is so flexible that it will let you do things that you might discover you didn't want to do.

If you are willing to learn the syntax of the editor and are willing to understand the operation of the program, this is one of the best of the Apple word processing packages. The ability to use any Apple file, the freedom to back up files and programs without complications or tricky copy routines, and the power of the programs make this package hard to surpass. But if your interest is in a package that creates a special word processing environment rather than leaving the Apple computer environment



intact, then this is not the program for you.

In any event, the best way to decide is to experiment with this and any other word processing program. This kind of trial should find out if the program will accomplish your special tasks. All do a good job of standard text editing and formatting. It is at the fringes of operation that these programs differ. If you need superscripts, subscripts, and strikeovers, if you need form letters, if you need multiple file access, if you need to edit programs as well as text, if you need to do arithmetic as you print, if you need to see exactly what you will get, or if you will be doing mostly tables, these requirements will help you determine which of the many good word processing programs for the Apple is the best compromise between features and price for you.

John V. Lombardi  
Bryan Hall 205  
337-8669

\*\*\*\*\*

#### FILE + FORMAT: Some General Comments

The program's word processing system is a fine product. It is well designed, flexible, powerful, and easy to use once the initial barrier of the manual is surmounted. Because the programs are designed to be used by people with a wide variety of needs, it is an open program that allows many modifications and changes. This has the advantage of being useful in non-standard text processing or program editing tasks, but it also means that there is ample opportunity to damage your text. Because it is easy to destroy files on the disk in the drive, you should be certain all files that you do not want to erase are locked. The powerful commands for moving files in and out of the system and changing them in subtle and not so subtle ways also has the potential for turning your files into junk.

This program is not at all hostile to the user, but it is so flexible that it will let you do things that you might discover you didn't want to do.

If you are willing to learn the syntax of the editor and are willing to understand the operation of the program, this is one of the best of the Apple word processing packages. The ability to use any Apple file, the freedom to back up files and programs without complications or tricky copy routines, and the power of the program make this package hard to surpass. But if your interest is in a package that creates a special word processing environment rather than leaving the Apple computer environment



## SUPER TEXT II

This excellent Apple word processor is of the integrated program variety, although the editor and the formatter are actually separate modes within the program. Super Text II (Muse Software, 330 N. Charles Street, Baltimore, MD 21201) is a modular, cursor based, word processor. Practically all operations are defined by the cursor. The package is structured around four basic modes. The Cursor mode is the main interaction point for the rest of the package. The search, search and replace, block moves and block deletes, and transfer to other modes occurs from this Cursor mode. The Add mode does the text entry, and a Change mode deletes and replaces characters, although it does not insert them. The Print mode runs the formatting and printing of the text, and a Math mode permits arithmetic and automatic column totalling while entering or editing a document.

At first glance, this arrangement of functions and modes appears clumsy, but if your writing and editing style can conform to the logic of this package, it works rather easily. In some word processing programs the most difficult barrier to efficient use is the sequence of special keys that accomplish the various editing tasks. This is also the case in Super Text, and in addition, you must adapt to the program's clearly defined logic.

In general terms this package assumes that writing takes place in two distinct operations. The first is text entry when you string words together until the piece is completed. Then, the second operation, the editing can take place. Here you are primarily concerned with moving, deleting, and rearranging the text to achieve the best result. If while editing you want to insert a line, a phrase, a letter, or other new material, you must switch into the Add mode to add text, and then switch back into the Cursor mode to continue editing. Similarly, if you want to change text by writing over the letters already entered into the file, you must switch into the Change mode, do your strikeouts (but no insertions), and then return to the Cursor mode to continue editing. Should you be in the Add mode and want to overwrite a word, you must do a double switch into the Cursor mode and then into the Change mode. To return to the Add you must reverse the double switch. Each of these modes permits movement around the text file and deletion of characters.

All of this jumping around from mode to mode takes some



practice. Nevertheless, the program does permit a very wide range of functions. To follow the organization of the package, it is best to review these functions mode by mode.

### The Cursor Mode

Because the Cursor mode is the heart of the package, it has the most elaborate set of commands. From this mode it is possible to inspect any portion of the file by moving the cursor to that section. Commands will position the cursor to the end of the line, the beginning of the line, the beginning or center or end of the current page. It will move the line the cursor is on to the center of the screen. In addition, you can move the text one line or one page at a time in either direction. You can jump to the beginning or the end of the file, or you can jump to the place where the last change was made in the text.

The Super Text window for editing is 40 columns wide, and the screen displays either all upper case if the Apple does not have a lower case adapter or upper/lower case if it does. With an adapter, the program supports the use of the shift key and provides a wire and instructions on how to enable the shift key. Because this is a cursor based system, the appearance of the cursor indicates which mode is currently in progress, and in addition, the program displays a status line on the bottom of the screen that can contain a variety of information about the mode and any other special features being implemented. Super Text does not support an 80 column display board.

The Cursor mode also will delete text either a character at a time, a word at a time, or from the cursor to the end of the line or to the end of the file. You can also delete the entire file, but in this instance the program asks for verification before performing the operation. All of these functions happen reasonably fast.

Super Text's Cursor mode also supports block operations. These are defined by inserting markers at the beginning and end of the block and then performing the various block functions. You can copy the block into another part of the file, save the block on disk as a separate file, delete the block from memory, and delete the block markers.

One of the unique features of Super Text is its ability to present a split screen. This divides the screen into two windows. You can then put different parts of your file in each window, and operate on both of them. This feature is very useful for documents that have repetitive, standard formats that recur in widely spaced parts of the text. You put the first instance in the top window, for example a special format for chapter or section titles, and then enter



the next occurrence in the bottom window. The split screen can be turned on and off and the size of the two windows can be adjusted. This is a specialized feature, but if you can use it, it is most valuable.

Unlike PIE+FORMAT, Super Text II can not be used to edit regular Apple text files. The program only reads those files it creates. It is, as a result, a closed program. Moreover, Super Text II is copy protected; the program disk can not be duplicated using standard copy routines, and the data disks can be copied only through Super Text's dual disk copy program. The package comes with two copies of the program disk. If you do not have two disk drives, the data on the disk can only be copied one file at a time by reading a file into Super Text, changing the data disk in the drive, and then saving the file onto the new disk. Further, new data disks must be formatted by Super Text prior to being used. The file manipulations possible through Super Text include most standard Apple disk commands including Rename, Delete, Load, and Save, but the program has a safety check built in before allowing files to be deleted or overwritten. The disk directory allows loads and saves by file number as well as by file name, a convenience feature. Most of these operations work just as they are described in the manual, although the procedure for resaving a locked file does not follow the manual's description.

According to the manual, the following sequence of actions ought to work. Enter a file, call the file system and save the file, lock the file, revise the file in memory, call the file system and save the file again either by the number in the directory or the name in the directory. If the file is locked, the program should ask if you want to unlock it, save the new version, and relock it. If you say yes, that operation takes place. However, if you have used the file name and not the file number, this sequence results in a program crash and a return to Applesoft. Fortunately, Super Text has excellent error recovery ability. It is possible to reset, follow the error recovery procedure, and return to Super Text with the file intact.

#### The Add Mode

This mode is identified by a flashing underline character for the cursor and an "A" in the lower right-hand corner of the screen. While in this mode you can enter text at the cursor. It makes no difference where the cursor is in the file. If it is in the middle, characters are simply inserted at the cursor and the rest of the text is adjusted to make everything fit. No carriage returns are required except to force a new line or add blank lines. Commands for new paragraphs and the commands for the printing or the tabbing options, or other printing features, are inserted through the use of control characters. Upper case lock is



possible.

Within the Add mode the backspace key is a destructive backspace. Each backspace erases the character it passes over. This makes it easy to correct errors as they are made. But it is not possible to move elsewhere in the file without switching to the Cursor mode. The Add mode has a number of special conventions to help with columns, outlines, and tabs, and these work quite well. You can also copy the word or section of the line directly above the cursor, a useful device for some applications. The program has word wrap, of course, which breaks each screen line at a space and puts the entire next word on the next line.

Unfortunately, Super Text, like most Apple word processors, has its one serious defect that prevents it from being regarded as the perfect program. In this case the difficulty is with the method used to put characters on the screen and to scroll the screen upwards as new lines are required at the bottom of the screen. Super Text scrolls the screen upward by moving each line up one row, a line at a time starting with the top line. This generates a reverse waterfall effect which is fun to watch but which takes a fraction of a second longer than it should. The result is that the word at the end of the last line that has to be moved to the next line, gets moved after the scrolling takes place. If you type very fast, this operation takes too long, and you lose a letter in the first or second word of the new line. The problem is most severe if your typing style is irregular with bursts of letters followed by pauses. Whether or not this is a fatal defect depends on individual preferences, and potential users should definitely try the program out very extensively before making any decision.

As if in compensation for the slowness of the screen display, Super Text allows a one-key entry of the combination "the". Wherever "the" occurs in a word or by itself you can hit a colon ":" and the magic three letters will be inserted into the text. When this feature is turned on, the regular colon can be accessed by a ctrl-K. For me this feature is not much help, but for other, more organized typists, it may be useful.

Super Text also treats blanks differently from other word processors. Blanks entered into the file on the screen will appear in the printed output. The formatting program does not delete blanks. This makes the entry of some kinds of text easier, but you must be careful when editing not to insert unwanted blanks.

An especially valuable feature of the text entry system is the ability to redefine twelve keys to mean something else. There are two types of redefinitions. The first is



simply a one to one replacement. For example, if you need a special symbol that your printer produces but that the Apple keyboard will not type into the file, you can redefine a key such as the ctrl-Y to produce that special character when the text is printed. Of course, the redefinition does not affect the screen display; the ctrl-Y shows on the screen as an inverse capital Y, but when the file is printed your special character will appear there.

The second type of redefinition involves what Super Text calls non-printing characters. This is something of a misnomer. By non-printing, Super Text means characters which do not affect the number of characters on the final printed line of text. Examples are special printer instructions that will generate a half-line feed to give footnote numbers in the text, give subscripts for chemical formulas, shift the type face or ribbon color, or do other operations available on various printers. These non-printing characters have the feature that they do not change the position of the printer on the line. It is not whether they print or not that defines them as non-printing, but whether the printer action ends up changing the horizontal print position on the line.

A specific example may make this clearer. These non-printing character redefinitions make it possible for one keyboard character to generate several output characters. Because none of these characters result in the horizontal print position being changed, the number defined is not important. For example, to achieve an overstrike character it is necessary to have the following sequence of events take place. Type the character, backspace, type the overstrike. At the end of this sequence the printer will be positioned at the same place it would have been without the backspace, overstrike sequence; hence the backspace+overstrike is, in Super Text terms, a non-printing character sequence even though the overstrike is actually printed. This makes possible the addition of accents, other diacritical marks, and the overstrikes sometimes necessary in scientific work. Text with these overstrikes will be justified correctly, such as José María Gómez Rodríguez. This is the operation that could not be accomplished with PIE+FORMAT, and it constitutes a major advantage for this word processor.

The procedure for creating these character substitutions is a bit complex and requires entering hexadecimal values into machine memory and then saving a special file to the Super Text disk. The manual is quite clear about how this is done, but a novice might have some difficulty. Any dealer, however, can easily help a user set up this facility to achieve the desired result.

Some other conventions that may be of interest in the



Add mode relate to the specification of some of the print formatting instructions. For example, underlining is accomplished by inserting a control character at the end of each word to be underlined. A word is defined as the characters between spaces. Centering of text is defined by a control character at the beginning of the line to be centered. This centered line is defined by the control character that begins it and the carriage return that ends it. The line must, of course, be shorter than the line length for the printed document, otherwise it can not be centered. Tabs and margins are specified in the text through command lines. These begin with a control character and have numbers that specify the margins or tab stops as necessary. All of these can be changed at any time in the document.

Super Text's Add mode will accept about 20,000 characters into memory, about 10 double-spaced typed pages. When the memory is full the program will accept no more characters at all and you must either delete a few lines or save the file to disk and start over with a new file. There is no danger of overflowing the input and damaging the program or any of the text.

Because many documents run over 10 pages, Super Text has an elaborate system of file linkage that not only links files for printing but also for editing. In the Cursor mode where the search and replace operation takes place, it is possible to do searches and replaces on a number of files. The program will read a file, search, if it finds it will do the replace, it will then search again, at the end of file 1 it will save file 1 if there were changes, it will load file 2, and then repeat the process for the files specified. The program also supports dual disk drives. This ability to link files is quite powerful and may be an essential feature for some applications. Super Text will not, however, link with other files for the pupose of form letters or other merging operations. The Super Text company has a special Form Letter program that can be purchased for this application.

### The Math Mode

Among its other unique features, Super Text has a Math mode that allows addition, subtraction, multiplication, and division plus column totals during the text editing process. Called from the Cursor mode, the Math mode will behave like a calculator or it will permit you to operate on numbers scattered throughout the text by moving the cursor to each number, or it will total the column of numbers above the cursor. It will also transfer the results of any of these calculations to the file. The Math mode supports the full range of cursor movement commands to help with the math options. For some documents, this is clearly a major



advantage of the Super Text program.

Here is an example of the Math mode. First I set up the columns of numbers, and then I use the Math mode to total them and line up the total. I insert the underline, presto, a table that is correctly formatted and totalled.

10	20	30	40
10	20	30	40
10	20	30	40
10.1	20.2	30.3	40.5
----	----	----	----
40.1	80.2	120.3	160.5

### The Print Mode

It is the Print mode that actually formats the document according to the specifications provided by the program or those entered into the text. It does all the usual things such as justification, top, bottom, left, and right margins, page numbering, paragraph indentation, tab stops, underlines, backspaces, overprinting to produce boldface type, centering, printer on and off, and whatever printer controls are specified by the user definable characters.

Super Text has a most useful print preview feature. This allows you to see on the Apple screen what the printed document will look like. Because the Apple screen only shows 40 characters on a line, the preview only shows a window into any 40 characters you want to see. So you can look at the first 40, the middle 40, the end 40, or any other 40 you like. This makes it possible to check for centered titles, indentations for outlines, location of page breaks, and other characteristics of the printed document. The feature is not useable for proofreading the text, but it saves many test printings if you have complex formatting instructions for outlines, bibliographies, lists, and the like.

The Print mode will print any portion of a file, and it starts printing at the cursor. But it also starts formatting at the cursor. So to print only page 6, for example, you need to insert the printer-off character at the beginning of the file and the printer-on character near where page six will start. Then the program formats, but does not print, the material from pages 1 to 5+, it then hits the printer-on character, turns the printer on, and formats and prints the material for the rest of page 5 and all of page 6, then it hits the printer-off command, turns the printer off and continues formatting. You can then abort the print from the keyboard since you are only interested in page 6. This is a bit clumsy, but it does work.



It is possible to change the page numbering in the middle of a document or to begin numbering from any page number. Page numbers can be placed anywhere in the top or bottom margin, but it is not possible to put a header or a footer into the document or to surround the page number with decorative characters such as "-3-". In compensation, the program will allow chapter-relative page numbering, thus a document can be numbered 1-1, 1-2, 2-1, 2-2, etc. When printing, the printer can be stopped, started, or the print terminated from the keyboard at the end of the current line. Any number of copies can be printed also.

### Super Text II Summary

This program, then, is a very sophisticated, integrated word processing package. It has many special features which may make it the program of choice. It has few disabilities with the notable exception of the lost characters at the beginning of the line for rapid typists. The program sells for \$150 and is available at Data Domain in Bloomington.

John V. Lombardi  
Bryan Hall 205  
337-8669

\*\*\*\*\*



## THE EXIDY SORCERER: PART I

Exidy Incorporated, a leading manufacturer of arcade type video games, took advantage of its experience in the field and developed the Sorcerer computer in 1978. Exidy combined the most popular features of the two popular computers at that time, namely the Commodore PET and the TRS-80, and added several unique features to create the Sorcerer computer.

### Hardware Features

The Sorcerer is a single board personal computer with a Z80 MPU operating at 1.8 mhz. A video monitor (or TV) and cassette player are required but not included. The Sorcerer 1 features memory expansion to 32K on-board. The Sorcerer 2, a later version, features up to 48K. An eight bit parallel port and an RS-232C port are included also. Two audio cassette drives may be used including motor control. Exidy offers an expansion box using the S-100 buss that can be plugged into the edge-card connector at the rear of the computer. A variety of S-100 products are now being used with the Sorcerer. A Centronics compatible parallel port allows a variety of printers to be used.

The Sorcerer keyboard is one of the most powerful in the industry. Each key may have up to five functions. Control characters are generated by pressing the control key and the proper letter. Regular upper and lower case ASCII characters are generated in normal typewriter fashion. Two full sets of graphics characters are generated by pressing the GRAPHIC key for the first set and SHIFT GRAPHIC for the second set. There are 78 keys including CLEAR, LINEFEED, CONTROL, RUNSTOP, ESCAPE, REPEAT, Cursor Control, and HOME. A sixteen key numeric pad is included to the right of the main keyboard.

The Sorcerer video displays 30 lines of 64 characters each. Each character is defined in an eight by eight matrix of dots giving an effective resolution of 512 by 240 pixels. Up to 128 graphics characters can be programmed in addition to the full ASCII character set. The Sorcerer has no high resolution graphics mode. The graphics are created by defining characters and using these graphics characters to create objects and shapes. This is often a serious drawback to those of us that wish to create smooth graphics animation.

Disk drives may be added with an S-100 compatible system or with the Exidy video/disk unit. The Exidy unit



features a built-in 12 inch video monitor and two quad density minidisk drives. The ROM-PAK can be removed and memory expanded in the S-100 box giving memory capacity up to 60K. CP/M is the disk operating system used by Exidy and other manufacturers with equipment hooked to the Sorcerer.

### Software Features

The Sorcerer features a plug-in ROM-PAK which can hold 16K of ROM or EPROM. Exidy offers four ROM-PAKs, an 8K Microsoft Floating Point BASIC, a Z80 Assembler Development System, a blank EPROM PAK, and a CP/M compatible Word Processor. These ROM-PAKs consist of a small PC card and are housed in an eight track tape cartridge that plugs into the side of the Sorcerer. BASIC is included with the purchase, all other ROM-PAKs are priced separately.

The Sorcerer features a monitor program in 4K ROM that may be used to route input or output, modify or examine registers, set baud rates, load or save machine language tapes, and other functions too numerous to mention. Unlike the TRS-80 and some other micros, the Sorcerer Monitor is accessible to the user with simple commands or with USR calls from BASIC. Drivers are provided for the RS-232 port, cassette I-O, Centronics parallel, keyboard, and video.

The BASIC used in the Sorcerer was created by Microsoft and is similar to PET BASIC. It features floating point arithmetic, one character BASIC keywords, and most other functions found in other versions of Microsoft BASIC. It is significantly less powerful than TRS-80 Level 2 or APPLESOFT as it lacks graphics commands and a text editor. Several machine language overlays are now available from independent vendors which provide the functions lacking in Exidy BASIC.

### Conclusions

The Sorcerer is a powerful and complete computer suffering largely from the lack of software support. The Sorcerer makes a magnificent smart terminal, has the only word processor in ROM anywhere, runs CP/M, and offers S-100 expansion. On the other hand, the graphics are hard to use, the BASIC is limited, and few non S-100 peripherals are available. Most of the drawbacks are slowly being resolved by independent vendors who believe in the virtues of the Sorcerer.

A variety of new system software has become available including an extended basic, graphics animation, powerful editors, and terminal communications. New peripherals include a real time clock, voice input-output, a tone generator, an EPROM programmer, and hard disk drives. Pascal is now being used on the Sorcerer in Europe and is soon to be introduced here.



Exidy is currently reorganizing and plans to support the Sorcerer as a system with the probable emphasis on CP/M applications as opposed to graphics and other specialized features.

For more information on the Sorcerer hardware and software feel free to call.

John A. Dodrill  
Real Estate Department  
School of Business  
333-8159

\* \* \* \* \*

#### APPLESOFT COMPILER SPEED COMPARISONS

The popularity of the Apple II microcomputer with its graphics commands integrated into a BASIC interpreter has made it my choice for distributing biomedical teaching simulations. However, simulations generally involve integrating nonlinear differential equations which is very time-consuming. As one example, simulation of a cardiac action potential, which requires the evaluation of 20 exponentials for each iteration, takes a total of over 7 minutes to compute and plot 250 points on the CRT screen when using Applesoft. Recent availability of Applesoft compilers, combined with the AM-9511 Arithmetic Processor Unit, speeds up this particular simulation by a factor of 10 times, but still retains all the conveniences of using the Applesoft interpreter for initial debugging.

During the summer of 1981 two Applesoft compilers were released. I have not used the Expediter II by On-Line Systems but have had experience with the Hayden Book Company Applesoft Compiler by Johathan Eiten (50 Essex Street, Rochelle Park, NJ 07662, \$200). These support essentially all of the standard commands of the Applesoft interpreter but produce a binary file which may be run as a program or as a subroutine. In the Hayden version the only precaution required was to locate the binary code so that it did not overlap the graphics memory area. For the cardiac potential program, with its many exponentials, the binary version ran at a rather disappointing 1.8 times as fast as Applesoft. Programs involving multidimensional arrays will show much greater speed advantages. The binary file was about twice as large as that for the Applesoft.

California Computer Systems has a card (CCS-7811) which interfaces the AM-9511 Arithmetic Processor Unit to the Apple II. Supplied with recent versions of that card is a



CCSOFT ROM which can be used to replace the software floating point routines in ROM versions of Applesoft. This provides the speed advantages of the numerical processor and the convenience of the BASIC interpreter. However, this processor truncates Applesoft's 5-byte real variables to only 4, so that a 0.1 entered may print as a 0.0999992. For the mentioned cardiac action potential this hardware addition provides a modest speed gain of 2.25 times as compared to the Applesoft interpreter.

The Hayden Compiler uses the standard Applesoft interpreter floating point software routines. When these are replaced with the CCSOFT ROM which supports the AM-9511 on the CCS-7811 Arithmetic Processor board, there is a dramatic increase of speed which amounts to 10 times for my particular application. The interesting thing is that this factor of 10 is greater than the product of the speeds gained by the two methods taken one at a time. The compiler and the numerical processor in combination are synergistic, to use a biomedical term.

The table below shows the speeds of the four possible combinations for this particular benchmark: the cardiac action combinations for this particular benchmark follow the cardiac action potential model of Beeler and Reuter. To implement this, a California Computer Systems 7811C Arithmetic Processor card was placed in slot 1. When used, the CCSOFT ROM replaced Applesoft chip E8 (card and rom from California Computer Systems, 250 Caribbean Drive, 24Sunnyvale, CA 94086, \$395). A coded statement was added to the first line of the Applesoft source program which forced the string space to be above the HGR2 locations in memory. This was then compiled and executed by a BRUN command. The speeds are all normalized to that for normal Applesoft which took 450 seconds for the 250 points computed and plotted.

		AM-9511 CCS-7811 <u>CCSOFT ROM</u>
<u>Applesoft</u>	<u>Software</u>	
Interpreter	1.00X	2.25X
Compiler	1.80X	10.00X

James E. Randall  
Medical Sciences  
Myers Hall

\*\*\*\*\*



### BLOOMINGTON APPLE USERS GROUP

After a considerable delay, the Bloomington Apple Users Group, called IndianApplUs, has organized. The first meeting brought out a small number of the paid-up members, but they quickly and unanimously elected John Prather of RCA and Data Domain as club president. Prather then distributed a variety of public domain software, the most interesting of which was the complete Fig-Forth package for the Apple.

Meetings will be on the third Wednesday of every month, 7:30 pm at the Data Domain in Bloomington. Anyone interested in the Apple computer can join.

Call John Prather at Data Domain for more information.

\*\*\*\*\*

### UPDATE ON THE BRAIN SURGEON

The Apple diagnostic package reviewed in a previous issue of the NEWSLETTER, the Brain Surgeon by Nikrom, has been reissued at a slightly higher price with a few more features. This new version works well as did the previous one. Added features are tests for monitor or television alignment, checks of the color adjustments for low and high resolution graphics, speaker function tests, and various tests for buttons and paddles. There are also general instructions about how to clean and maintain your Apple computer. The new package sells for \$49 at Data Domain in Bloomington.

\*\*\*\*\*

### DATA-BASE MANAGEMENT SYSTEMS

The Small Computer Support Group NEWSLETTER is interested in developing a series on microcomputer data-base systems. Next to word processing packages, these programs are the most expensive and most difficult to evaluate. If you have experience with a package that runs on any of the major micros --CP/M machines, TRS-80, Apple, or others-- please let us know. If you would be willing to help with the evaluation of this type of software, please send the NEWSLETTER a note.



## THE BUFFERED MODEM

One of the advantages of having a microcomputer is the ability to use a larger, mainframe computer. With the expansion of enthusiasm for microcomputers, many companies have set up time-sharing systems specifically for micro users. The Source is one example. Similarly, users with access to university or college computers or business computers often find it very helpful to make their micros act like terminals. It is then possible to prepare files off line for transmission to the mainframe for specialized processing, or transfer subsections of large files from the mainframe to the micro for manipulation off line. Many Apple users also communicate with other Apple users through free bulletin board services available for the price of a phone call. All of these uses require a communications device such as a modem and a software package to make the modem work easily. The most popular communications hardware is probably the D.C. Hayes Micromodem II, which also includes considerable built-in software to make the communications easy. But even so, there is a need for a good communications package to help the Apple work with other computers, hence the appearance of ASCII Express II (reviewed in an earlier issue of the Small Computer Support Group NEWSLETTER), Data Capture 4.0, and the package reviewed here, The Buffered Modem. All of these packages attempt to provide the ultimate software assistance for Apple data communications. All of them will do a fine job of simply dialing up the other computer and permitting you to work with it in terminal mode. But so too, of course, will the built-in Micromodem II terminal program. These packages also provide assistance in copying files from the other computer to the Apple or from the Apple to the other computer, and they have facilities for using a printer in conjunction with the modem system.

The Buffered Modem is a useful and cleanly designed package of programs that makes communication with another computer from the Apple quite easy. The program is menu driven, and the choices usually result in the loading of a subprogram. The menus are very thoroughly protected from bad entries, and they require you to verify everything. This is good protection, but it can become an annoyance with frequent use.

The program maintains two special areas in memory for the use of the program features. The first is the Capture Buffer. This is an area of up to 23,000 characters that can receive and save in memory characters sent from the other computer. It can also record what you send out to the other computer as well as what is received. Or it can save only what you receive or what you send. The Capture Buffer can be reviewed very rapidly to see what is there before it is saved or erased. But the material in the buffer can not be edited before it is sent out or saved. The program will allow the capture buffer to overrun the space allocated.



When you get near full the program buzzes to indicate that the data entering the buffer is about to overrun the buffer. If you let it overrun, the first lines in the file are pushed out of memory to make room for the new lines coming into memory.

In addition to the capture buffer, the program has a printing buffer. The purpose of this buffer is to hold the characters coming from the other computer until they can be printed by a printer that may be slower than the transfer speed of the modem. At 110 to 300 baud this is not really a major advantage, since most printers can keep up reasonably well and most other computers can be adjusted to send nulls or slow down the transmission speed if necessary. But the Buffered Modem program expects to be able to handle 1200 baud transmissions in the future. The space for this buffer is shared with the capture buffer. You can set the size of each, and each can vary from 0 to 23000 as long as the total buffer size (print and capture) does not exceed 23000. When the print buffer is overrun it is reset to zero and what is in the buffer is lost. A status line can be called that tells how much space is allocated and used for each buffer.

The Buffered Modem does not support the Paymar lower case adapter in its screen display except when listing the contents of the capture buffer. Information cannot apparently be entered in upper and lower case from a Paymar equipped Apple. The manual says that most 80 column cards are or will be supported. Currently the program requires a 48K Apple II or II+, Applesoft in ROM, Modem card or serial card with acoustic coupler. It supports parallel or serial printers with appropriate cards and M&R 80 column board and some other boards. I tested only the set up for a serial printer, CCS Serial Card, Micromodem II, Paymar adapter, all on an Apple II with an Applesoft card. The setup procedure is quite well designed and flexible. It was no problem to set up the program to work with my hardware. The program is supplied on a DOS 3.2 disk which can be Muffined up to DOS 3.3.

In sum, this is a most competent program that does just what it says it will do. The program is not as versatile nor as powerful as the ASCII Express II communications package, but it holds out the promise of many new additions including 1200 baud transmission. The Buffered Modem is available from Data Domain in Bloomington for about \$70.

John V. Lombardi  
Bryan Hall 205  
337-8669

\* \* \* \* \*



## LOCAL BULLETIN BOARDS

Two local bulletin boards are available in the Bloomington-Indianapolis area. These provide useful information, announcements, advertisements, and general news about micro computers. These two are mostly Apple oriented. The Data Domain, Bloomington, Indiana, has a bulletin board available from 8 pm to 8 am Monday through Saturday and all day on Sunday at 334-2522. The Indianapolis users group, the Apple Pickers, has a bulletin board accessible 24 hours a day by dialing (317) 862-6191. All that is required to use these services is an Apple or other micro computer equipped with a modem. You simply dial the number and then follow the instructions once connected. These are free for the price of a phone call.

\* \* \* \* \*

If you received this NEWSLETTER you are on the mailing list. If you or a friend would like to be put on the mailing list, please send the form below to the editor.

Name \_\_\_\_\_

Address \_\_\_\_\_

IU Department and Phone \_\_\_\_\_

Special Interests \_\_\_\_\_

\* \* \* \* \*

Small Computer Support Group

NEWSLETTER

Bryan Hall 205 IUB

\* \* \* \* \*