COMMODORE COMPUTER CLUB NEWS

VOLUME 1, NUMBER 9 — JUNE, 1984 — VANCOUVER, B.C. • FREE • 24-HOUR CLUB ANSWER PHONE: PET-3311 (738-3311)

WHAT'S A DATA BASE?

By GERRY DONALDSON
(from Calgary Users' Newsletter)

Data is computerese for information. A *data base* is simply a collection of information. We are acquainted with many data bases: a telephone book, department store catalog, supermarket advertising flyer, a checkbook register, recipe box, notes on a collection of rock and minerals (or stamps, or cars, or flags, or clothes), etc. One thing all people and institutions do is manage information -- on paper, or in their head or (with a computer) ... electronically!

A program that enables the user to manage a "base" of information is called a *DBMS*: Data Base Management System. Data base management has been one of the major applications of computers since their inception. A *DBMS* is simply an electronic method for storing and ordering information so it is easy to find and manipulate. The present state of the art on microcomputers in a DBMS is a program called *dBASE II* which, as it happens, was written entirely in machine language for the Z-80 microprocessor. Commodore users must purchase the CP/M operating system to run this particular program.

FILES, RECORDS AND FIELDS

One must understand what each of these units of information are and how they relate to other units before one can easily deal with a DBMS.

The *DATA FILE* is the largest unit of information. A hobbyist might, for example, keep all of his information stored on disk in his "stamp file." The two most frequently seen types of data files are the *SEQUENTIAL* and *RANDOM ACCESS FILE*. These terms indicate the difference between these two types of data files in terms of how you store or retrieve a record of information. (Programmers would use the terms "write to" or "read from" the file.)

Let's suppose our stamp collector divides his data file into 100 records -- a different record for each country. When he later wishes to retrieve data (i.e. "read from a particular record"), he will do so either "sequentially" or by "random access." Let us suppose that he wishes to retrieve information from the 7th record.

Sequential Files: Data in a sequential file must, like a cassette music tape, be stored or retrieved "in sequence." For example, to retrieve information stored in the 7th record, the program must first go to the first record, determine if that is the desired record (it is not), then go to the second record, the third, etc. Finally, after sequentially searching and determining that records #1 through #6 do not contain the desired record, the search arrives at the 7th record. Here it determines that this is indeed the desired record and thus will retrieve the information in that record. The process is much the same as playing the 7th song on a cassette tape after first searching through all prior songs before arriving at the 7th song in order to "access" it. The process can be very time consuming, whether you're accessing a song or a data record of information.

Random Access Files: Data in a random access file may, like a phonograph record, be stored or retrieved "randomly." For example, to retrieve information stored in the 7th record, the program may go *directly* to the 7th record. It need not go to the 1st record, go to jail, or go by GO. This process, by comparison with sequential access, is much more rapid. It is much the same as playing the 7th song on a phonograph record by moving the stylus directly to the first groove of the song. In a disk drive the read/write arm is moved directly to the first byte of the desired recorrd.

Note, however, that sequential files generally use far less space than random access files. This advantage, however, is far outweighed by the speed of access of random access files.

Fields of Information refer to specific pieces of information contained in each record. Our stamp collector, for example, might store fields of information such as year of issue, denomination, condition, theme, value and color.

Programs generally use both types of files in various ways to maximize flexibility in managing information. For example, data records may be stored in random access files to get rapid access, but a *key* correlating the actual location of records with a *sorted* sequence will be stored in a sequential file to save space on the disk.

FUNCTIONS OF DATABASE MANAGEMENT

There are certain things that a database program should somehow do. The effectiveness and efficiency with which it does it will determine the quality of that program, but it should fulfill the following functions.

1. Create files.
2. Add records to an estab-

(Continued on page two)

COMPUTER PROGRAMMING LAWS

* Any given program, when running, is obsolete.

* If a program is useless, it will have to be documented.

* If a program is useful, it will have to be changed.

* Any program will expand to fill any available memory.

* The value of a program is directly proportional to the weight of its output.

* Program complexity grows until it exceeds the capability of the programmer to maintain it.

* Make it possible for programmers to write in English, and you will find out that programmers cannot write in English.

* If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.

* Inside every large program is a small program struggling to get out.

* If a test installation functions perfectly, all subsequent systems will malfunction.

* Not until a program has been in production for at least six months will the most harmful error be discovered.

* A carelessly planned project takes three times longer to complete than expected; a carefully planned project will only take twice as long.

* If the input editor has been designed to reject all bad input, an ingenious idiot will discover a method to get bad data past it.

* Machines work, people should think.

-- From WINePEG newsletter

INSIDE THE 1541: PART 3

By LARRY PHILLIPS

In part 2 we NEWed a diskette. We are now ready to put something on it. Since the first item most people put on a diskette is a program, we will now see what happens when we tell the computer to SAVE.

Assume that we have a program written in BASIC. If we type in the command SAVE "TEST",8 we will see that the red LED on the disk unit will turn on, the motor will spin, some 'ticking' sounds will be heard, the LED will extinguish, and your cursor will reappear on the screen. Seems simple, but let's look at what has happened.

The computer analyzed your command, and sent signals to the disk controller (device #8) telling it to open a 'program' file with the name "TEST". The controller acknowledged the command by telling the computer that it was powered on and ready to receive data. The computer then sent the address of the start of the program, followed by the program itself. It then told the controller that it had finished sending data, and returned to whatever it was doing, usually flashing the cursor and waiting for your next command.

Well, that sounded a bit more complex, but during this time the controller was extremely busy. Consider the problem from the controller's point of view. It had to make sure that there was a disk in the drive, start the motor, and tell the drive to move the head to the directory (track 18). All this before doing any reading or writing. Then it had to read the directory in order to find an empty place to write the program name. It had to check to see if it needed to use a new sector in the directory, and if so, to rewrite the last sector used in order to put in a track and sector link pointing to the newly written sector. It had to make sure that there was at least one block (sector) free to use for the program data. Then it looked at the BAM (block availability map) to find the physically nearest sector to start writing the program.

This accomplished, it was ready to receive data from the computer. As it did this, it put it all into a 'buffer' (a chunk of memory used to store data temporarily). It then had to convert this data from 8 bit bytes into a special code called GCR, or 'group code recording'. This is a code that has very strict criteria as to how many 1's or 0's may be written in a row, and serves to keep the reading circuitry in sync when reading back the data. GCR uses 10 bits for every 8 bit byte written, but does not require the use of non-informational 'clock' bits. Once converted to GCR, the controller causes the head to seek to the track to be written, and waits for the appropriate sector to show up under the head. It recognizes the right sector by reading everything passing under the head and locating the sector by its header (written during the NEW). After detecting the header, and getting into the data area of the sector, it was then able to write the data.

The first thing written was a 'link' to the next track and sector to be used for the same program. It knows when this is necessary because the computer did not tell it that it was finished sending data. This passing of information about the data, by the way, is commonly called 'handshaking'. As for the program itself, just as the computer did not concern itself with what type of device was on the serial bus, the controller does not concern itself with the actual data written. In other words, the first two bytes are treated as data to be written like all the rest, even though they are the load address and not part of the program. If the track still has free sectors on it, the controller will use them at this time. Otherwise, it will tell the drive to seek to the next track with a free block. This process goes on until the computer tells the controller that it has no more data to send. Now the controller has a partial bufferful of data. It will write the data in the appropriate sector, but instead of writing a link in the first two positions, it will write a 0 in the first position, and a count in the second. The count will specify the position of the last valid byte written in the sector.

The program is saved! But wait! The controller isn't finished yet. It must write the directory entry. Until it was finished, it didn't know how long

(Continued on page five)

WHAT'S A DATA BASE *... from page one*

lished file: the user should not have to re-create a file from scratch.

3. Update records in a file: Addresses of friends in a mailing list file change, as do values of stamps in a stamp file. The user should be able to change such individual "fields" of information without having to re-write an entire record.

4. Delete records in a file: people die and stamps get sold. The user should be able to "remove" a particular record from a file.

5. Redesign layout: the user may wish to later add or delete fields of information to all records of a particular file.

6. Sort file: users want files in alphabetical, numerical, ascending, descending order, etc. A good DBMS should permit subsorts as well where, for example, all stamps -- after being sorted by value, should be sorted by denomination with each value category.

7. Search records in a file: this is one of the most valuable functions of all. Suppose, for example, that you find a stray piece of paper on your desk with a phone number on it, but can't recall whose it was. A good DBMS should allow you to enter the phone number and then find the record with that number in it. Or perhaps you want all recipes with (or without) spinach. Simply specify this and your DBMS should find them for you -- and do so rapidly!

8. Arithmetic operations: this simply means that you can add, subtract, multiply or divide various fields of numeric information in various ways. For example, a good DBMS will add all outstanding balances of an Accounts Receivable file or find the average of all players on a basketball team using a statistical file. Teachers, of course, appreciate the ability of a DBMS in calculating grades each term.

9. Design reports: this permits the user to design output in the form of reports that need not duplicate an entire record. Only some information from some records may be desired, or perhaps only a statistical summary of other fields. Users should ensure that this separate utility is present in a DBMS.

10. Print reports: of course you want the choice of a hard copy (i.e.: paper copy) of output from time to time.

11. Interface with other utilities: this feature increases the power of a DBMS immensely. It should be possible, for example, to compose a form letter on your word processor and have output of that letter merge with various information in your data base, such as name, address, and outstanding balance of customers in your Accounts Receivable file. Alternatively you may want to merge information from a BASIC program into records of a DBMS file.

This function is generally accomplished by ensuring that all utilities store data in a common format such as sequential or random access files on disk.

12. Security levels: there should be some feature that would frustrate would-be snoops. The requiring of one or more passwords before permitting access to a disk, file or certain records is conventional.

WORD PROCESSING

by LOUIS BERNHARDT

Although I'm fortunate enough to make my living working in the computer division of a large corporation, with its attendant access to huge IBM mainframes (with many megabytes of main storage) and a number of IBM PC's and clones, I'm constantly impressed by the performance of my own humble C-64. I'm floored by the fact that it runs Multiplan and Flight Simulator II. I'm amazed that computer users -- the same ones that complain about how difficult it is to interact with the mainframe user interface -- have bought and use the C-64 in droves, especially when one considers the inherent unfriendliness of Commodore's disk operating system, the fact that one practically has to be a Master Console Operator just to load a directory.

I have not had too much experience with word processors. On the mainframe, one uses "text editors" which were originally designed for entry of program source code, and the closest thing to a word processor is the "text formatter", which is used in conjunction with the text editor. I have used the rather primitive word processor which runs on North Star Advantage micros, as well as the rather sophisticated MultiMate which comes with the Corona (an IBM PC clone) and contains some very serious bugs as well as an extremely slow and annoying keyboard buffer, so when it came time to acquire a WP for the C-64, I did have a fairly good idea of what I wanted.

There were probably two reasons I had for not buying Paper Clip. 1) I don't like anything from Toronto, 2) Not having word wrap is very bothersome. (Word wrap is where incomplete words on one line are automatically placed intact on the next one.) After looking at several WP's and listening to comments from users of several others, I bought the Mirage Concepts Word Processor because 1) it seemed sophisticated, 2) it looked easy to use, 3) it creates standard sequential files, and 4) nobody knew enough about it to tell me what was wrong with it. I am going to tell you what's wrong with it.

To be absolutely fair, the version I bought came with a notice stating that the price had recently been reduced and a new version with 30,000-word spelling dictionary and other enhancements was available as soon as I sent in the warranty card. I sent in the card with order for the enhancements to the company's head office in Fresno, CA about a month ago. I'm still waiting. Anyway, I don't know if any of the enhancements address the problems I found. I'm one of those people confident enough in my spelling not to be too concerned with having a spelling checker, so I'm not too concerned about that part of it.

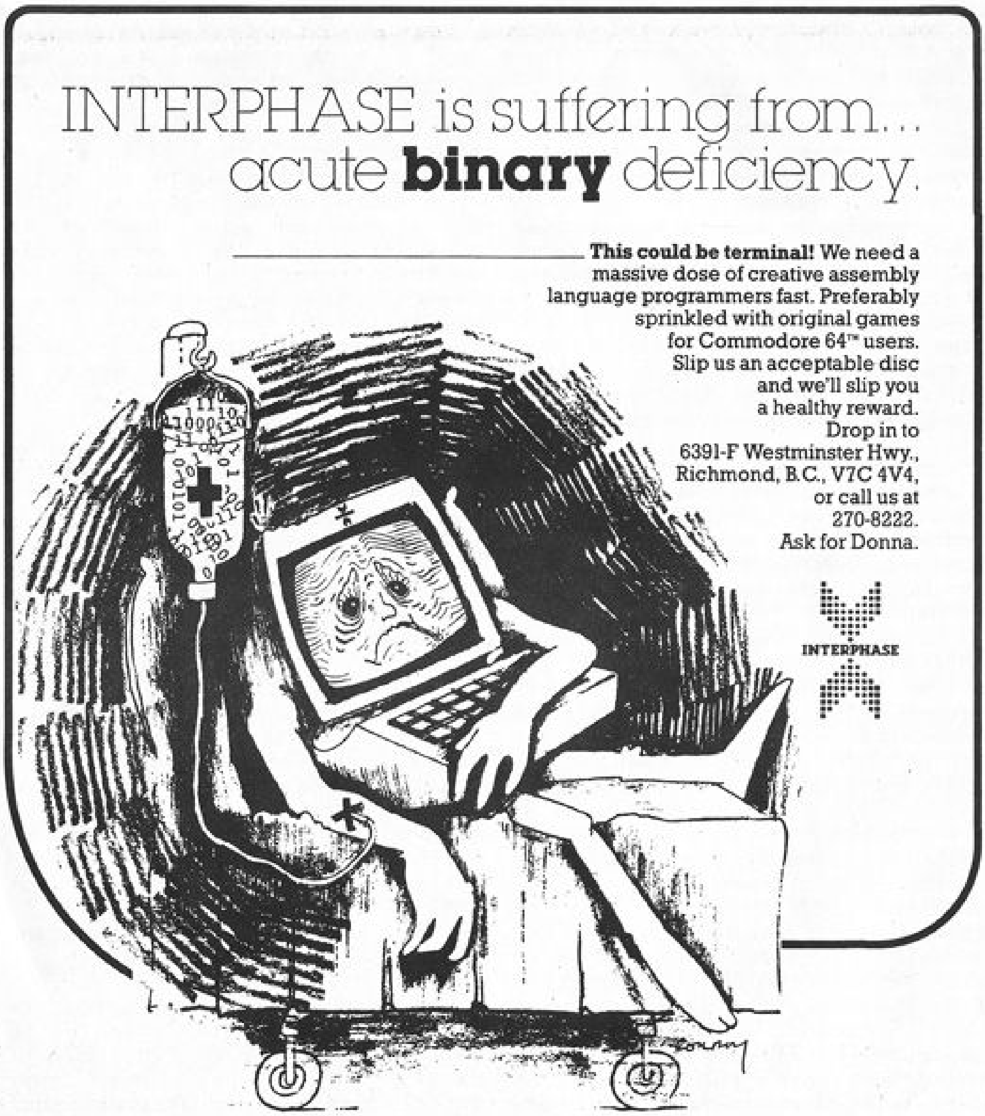
The Mirage Word Processor (MWP) comes professionally packaged in a slick plastic 3-ring binder which looks exactly like the binder used for their database management package. Be sure you don't buy the wrong package at the computer store. Inside the binder is a nicely-typeset 200-page manual with index, appendices, reference section, and tutorial; a doublesided quick-reference card; one diskette; and all the warranty papers. The index is inadequate -- not enough useful entries. The diskette is copy-protected, and a backup is available for US$7.00 when you send in the warranty card. The product is warranted for 90 days; after this period, a replacement diskette is available for US$10 plus the defective diskette.

A LOAD"*",8 brings up the initial screen of the MWP, where you get to select background, text, and highlight colors. This boot program is written in BASIC, but you can't change it (unless you save it to another disk) because of the distribution diskette's copy protection. This is unfortunate, since it prevents you from configuring the system in your own default colors. After you hit RETURN, the real MWP is brought in. The manual says load time is about 45 seconds. I've timed it at 1:03.

The next menu instructs you to remove the distribution diskette, insert your own data diskette, and hit RETURN. Your data diskette doesn't have to be formatted; MWP will do it for you, as you will find from the options provided on the next (master) menu. You can edit, save, load, merge, delete, or format a file, view a directory (nondestructively!), print, advanced print, or quit. It's helpful to follow along in the manual's tutorial for the initial phase of getting to know the MWP, but after a very short time, the quick-reference card becomes adequate. There are the odd occasions which require looking at the manual. In such cases, don't bother looking in the index; what you want probably isn't there.

When you want to edit a document, you key in "1" on the master menu. This brings up an edit menu asking for lines per page, characters per line, and

(Continued on page six)



A COMPLEAT GUIDE TO MACHINE LANGUAGE PROGRAMMING ON THE PET

By HAROLD BROCHMANN

USING THE CLEAR SCREEN ROUTINE FOR SOME FUN (6-3)

Enter MICROMON and assemble at $033A:

```
 .A 033A A2 00    LDX #$00
```

Now we will make use of another one of MICROMONS neat tricks ... the TRANSFER COMMAND.

```
 .T  E246  E256 033C  (for BASIC 3) ...
```

and

```
 .T E05A E06A 033C  (for BASIC 4)
```

Now:

```
 .D033A
```

Gives you:

```
 .:  033A A2 00    LDX #$00
 .:  033C A9 20    LDA #$20
 .:  033E 9D 00 80 STA $8000,X
 .:  0341 9D 00 81 STA $8100,X
 .:  0344 9D 00 82 STA $8200,X
 .:  0347 9D 00 83 STA $8300,X
 .:  034A CA       DEX
 .:  034B D0 F1    BNE $033E
 .:  .....
```

MICROMON has TRANSFERRED a part of the clear screen routine to $033A - .

All we need to do now is to add a RETURN at the end.

```
 .A034DRTS
```

Disassemble from $033A again just to make certain that all is in order.

Note that the number which gets "poked" into all the screen locations is contained in $033D. This corrresponds to decimal 829. Exit MICROMON, NEW, and enter this little BASIC program:

```
 10 FOR X=0 TO 255
 20 POKE 829,X
 30 SYS 826
 40 NEXT
```

Of course this dazzling display is much slower than it should be because we jump back and forth to BASIC. Let's do the whole thing in ML. Leaving the previous program intact where we placed it betweem $033A and $034D, assemble this one at $034E:

```
 .:  034E A9 00    LDA #$00
 .:  0350 8D 3D 03 STA $033D
 .:  0353 20 3A 03 JSR $033A
 .:  0356 EE 3D 03 INC $033D
 .:  0359 D0 F8    BNE $0353
 .:  035B 60       RTS
```

In order to run this program which is the ML equivalent to the BASIC program we wrote a paragraph or two ago, we need to know the decimal value for $034E. This is easily accomplished with MICROMON. Simply eneter:

```
 .$034E
```

... and we get our answer, 846. Exit Micromon and SYS 846.

This version, being all ML, is even faster. Let's see just how fast:

```
 10 TI$="000000"
 20 SYS 846
 30 PRINT TI
```

My PET gave the result 109 for this experiment. This means that this ML routine will fill the screen 256 times in 109/60 = 1.8 secs. That's about 7 milliseconds per screenfull ... more or less. This is 700 times as fast as the following BASIC program:

```
 10 FOR X=32768 TO 33767
 20 POKE X,32
 30 NEXT
```

....and 300 times as fast as this one:

```
 10 PRINT CHR$(147);
 20 FOR X=1 TO 1000
 30 PRINT "A";
 40 NEXT
```

THE COMPUTING CYNIC

By MARK JACQUES

What is computer software (and hardware) really worth? This question is a source of constant puzzlement to me.

People who received a copy of the recent catalogue from Protecto Enterprizes, perhaps the largest dealer in C-64/VIC merchandise in the States, must have been just as surprised as me at some of the prices it contained on software.

C-64 game programs with a U.S. list price of up to $34.95 were reduced to $9.95, VIC-20 cartridges which retail around the $50 range in Vancouver cost only $7.95. Some VIC game programs on tape were 3 for $10!

Even with an added expense of about 50% over Protecto's price thanks to exchange, duty, postage, and so forth, there are obviously considerable savings to be had.

Locally, there is some interesting activity with the price of the C-64. The most recent "deal" is a 64 plus something else such as Easy Script or Magic Desk. The software "bonus" is allegedly valued in the neighborhood of $100, making the 64 worth $300.

This is the kind of marketing procedure followed some time ago with the VIC. The VIC used to cost $400 plus $120 for the cassette recorder. Then the price was $400 including the cassette. Then came the VIC plus cassette plus bonus software and/or joystick. Finally, the VIC itself ended up being sold for $127 or so, which is supposedly $2 less than the wholesale cost to the dealer.

I was quite interested by the most recent 64 financial shenanigans, since a couple of months ago I was in one of our local stores which had dropped the price of the 64 to around $375. This store was having a "no reasonable offer refused" sale, and I inquired about the 64 in this situation. The salesman told me that "There's no way we can reduce this any further, since we're already losing money." He then mentioned that the price of Commodore hardware was soon to go up! Two months later, I can only say "Sure it is..."

***

Machine language is something which fascinates most people who get involved with computers, including myself. Unfortunately, trying to learn something about it is another matter.

There are numerous books about ML, most of which cost in the range of $20-$25. Most of these are of a dismally bad quality. As a review in a recent issue of TPUG magazine points, the Compute-published *Machine Language for Beginners* is the best of a bad lot, but even it has numerous faults, not the least of which is the fact that it is written for a variety of 6502-based computers.

One thing I can't understand is why people who write about ML can't illustrate the various instructions and so forth in terms of concrete examples to which normal people can relate. Surely this is the most intelligent way of doing things, but the Compute book, among others, loses me in a sea of mumbo-jumbo and bad logic.

At $20 a crack, learning about ML becomes an expensive and, in the final analysis, frustrating exercise which does little to demystify the mysterious world of computers.

***

Gripe of the month: those "utilities" that appear in magazines which all utilize the same memory areas, either up in the top of RAM or the cassette buffer (SYS 828)...

Published by The Commodore Computer Club, P.O. Box 91164, West Vancouver, B.C. V7V 3N6. Editor: Mike Quigley.

Copyright 1984 by the Commodore Computer Club. Material in this paper may not be reprinted for profit without written permission. Opinions expressed are those of the individual authors, and not necessarily those of The Commodore Computer Club. The name "Commodore" is used with the permission of Commodore Business Machines of Canada Ltd.

Club meetings are normally held: *Workshop*: first Tuesday of the month, 7:00 p.m., Thompson Secondary School cafeteria, 1755 E. 55th Ave. (near Victoria Drive); *Business*: third Tuesday of the month -- 7:00 p.m., King Edward Campus, 1155 East Broadway, 2nd floor auditorium. These dates and locations are subject to change. For up-to-date information on any changes, please call the club's 24-hour answer phone:

PET-3311 (738-3311)

Club Executive: President -- Jim Bauerle; Vice-President -- Sigmund Steiner; Secretary -- Marvin Steinway; Treasurer -- Hu Reijne; Directors -- Robert de Boer, Guenter Hake, Jim Jorgenson, Terry Juuti, Murray Kopit, Mike Quigley, Elmer Roy, Philip Seligman, Nick Shevchenko, Gerri Sinclair, Tony Smith, Arthur Tamer, Al Townsend, Jim Wilcox.

—DATA BASE MANAGER—

By PHILIP D. SELIGMAN

In my search for a database manager that was both powerful and flexible, I came upon Database Manager by Mirage Concepts. The programme would appear to combine into one package most of the features one could want in a database management system, with few of the restrictions on use often found with such a programme.

The programme 'boots up' with a main menu on the top one-third of the screen setting out the primary options contained in Database Manager. This split-screen type of display is present during all functions of the programme, guiding the user on his or her options and required inputs.

Creating a form on Database Manager is a pleasure. Merely prepare your form on the screen as you would on a piece of paper, using the F1 function key to place an underline where your fields will be placed. The form may be as complicated as you wish, with the only 'restrictions' (if you can call it that when compared to other database management programmes) being a maximum permitted field size of 250 characters, a maximum of 200 fields per record (form), and no more than a total of 2000 characters (2500 characters if you include titles) permitted on each record. When creating the form, you have free use of all cursor keys to fly over the form, making additions and deletions before you are satisfied with your creation.

One especially helpful feature to many who will use Database Manager for financial record-keeping is the power to use 'calculated fields'. For example, when creating your form, you may wish to keep a record of your weekly gasoline purchases, together with a running account of your purchases to date, and your average weekly purchase. Database Manager will allow you to tell the computer to keep track of the amounts placed in each of your 'weekly purchase' fields, total them and place the running total in the appropriate field set up for that purpose, and then calculate the average of the amounts entered, and place that figure in its appropriate field. A more complicated form of the 'calculated field' ability is what makes a spreadsheet such a powerful tool.

Once you are ready to save your form, merely press the indicated function key and follow the prompted instructions. One small complaint I have with the programme is that, should the form fail to be 'SAVED' for any reason (for example, if the disk was full or unformatted), the form you have created is lost, and you must start again. Therefore, always make sure you have a formatted disk available before creating a form (the programme will format a disk for you with a mere keystroke, if you remember to ask it in time).

Once your form has been created and saved, you are ready to input your data using the 'Append' function. The form you have created is displayed, and the flashing cursor placed in the subject field. Input is fast and easy, and you can continuously refer to the screen for help and to keep you advised of the current status of your input. As each record is completed and entered, a new blank form is presented for your data input. At any time, when using any of the options available, pressing the F7 function key will return you to the prior menu, and eventually will lead you back to the main menu.

Okay, so you have your database ... now what? Perhaps you wish to review your records? Or perhaps view only reords that satisfy certain conditions? With Database Manager, not only can you request it search your database for a particular name or condition ('all persons whose last name is 'Smith''), but you can search on very exacting specifications. For example, on my database consisting of my friends' and family's home address, telephone number, birthdate, occupation, work address, work telephone, alternate address, alternate telephone, and any comments I may have, I can instruct the programme to search for 'all my friends over the age of 25, living outside of the City of Vancouver, but inside the Province of British Columbia, who are either lawyers or students'. And it works!

One thing I especially appreciated with Database Manager was the ability to edit individual records while in the REVIEW function, then allowing the programme to continue with its search. Some other database management programmes require you to exit from the search function and enter a special edit funtion for this purpose.

Although the programme generates a RELATIVE file to allow a faster retrieval and search time that that possible with a SEQUENTIAL file, many of the REVIEW functions still will require the entire database be searched. Thus, you should try to keep your form as small as possible, as a longer form will take longer to search. As well, you will not be able to store as many individual records on a disk with longer forms. To help speed your access time to your records, Database Manager contains two different options: SORT and CREATE SUBFILE. SORT will allow you to sort your records by any field in alphabetical (or numeric) order, and will also allow multiple sorts on as many fields as you have in your record. Using the above example of my database, I could sort my records by city/age/last name, as a multiple sort. A record of the sort is kept by the programme and may be saved on the disk for future reference. Use of the SORT function can speed up your access time on a search to almost instantaneous access.

Creating a SUBFILE is useful when you do not require the entire database for a particular task. For example, if I was a teacher with information on all my students in each grade, I may wish to create a subfile of students in only one particular grade for certain purposes. Obviously, a search on a small, specialized database would be much faster that using the entire main database.

One function I found especially welcome was the MERGE FILES option. We've all done it -- created our database, typed in all our data, and then discovered

(Continued on page six)

INSIDE THE 1541: PART 3

from page two

the program was. It needs to know if the program size required more blocks than were available. If while writing the program, it found it had no more sectors available, it would have flashed the LED, stored an error message in its memory, and stopped. It also read back each block as it was written, just to make sure that everything went right. If any errors were detected, the directory entry would be written with a special identifier byte to show that the file was not properly completed. This will cause the dreaded asterisk (*) to appear in a directory listing. However, if all went well, the directory entry will be written with a byte that signifies a program file, and the name, length, and starting track and sector of the program.

Whew! As the old expression goes, "Busier than a one armed paper hanger".

Some of the points brought out in this installment make for interesting possibilities. Since the controller does not care what data is sent, it becomes relatively simple to open a program file just as if you were opening a data file. You may then use PRINT# to send a starting address, followed by data or a program. Thus, you may use sequential file methods to create either a LOADable program, or LOADable data, and believe me, LOAD is much faster than GET#. More on this in later installments.

WORD PROCESSING

from page three

paragraph indent. Defaults are provided, and it's usually best to work with them (55, 65, and 5, respectively) since if they're changed you'll have to reset them every time you go into edit, whereas if you let them default, you just have to hit RETURN to begin editing. You are then presented with a screen showing what page, line, and character your cursor is currently on, the status of insert, block mode, and column mode (all default to OFF), and your text area in the colors you specified.

There is a highlighted line towards the bottom of the text area. This is where all text is entered. As you type along, the text scrolls upward. This alleviates a major complaint of many word processors which throw the cursor to the top of the screen whenever a page is filled. With the text entry area near the bottom, always at the same place, you can review what you've recently typed at any time with a minimum of scrolling.

Funny things MWP does: If you move the cursor backwards at any time (using DEL or CRSR), page, line, and char number immediately zero themselves. You can get them back by hitting F1 or by shifting into insert mode and shifting back out (SHIFT INST) twice.

If you've just loaded a large document and gone into edit, you must wait a while for the document to go through some formatting. MWP displays a message and sort of countdown (probably of blocks left to format) while you wait.

Sometimes, as you're typing away, you will end a line with a period in column 39 and a space in column 40. Since it's common practice to leave two spaces after a period, you end up placing the second blank into the first column on the next line. When you print this, however, you will sometimes get a blank in column 1. This is inexcusable! It should only happen to leading blanks which follow carriage returns!

Centering is awkward. You type your text, hit RETURN, move the cursor back to the last letter in the line to be centered, then hit CTRL :. You should be able to center text as you type.

As I mentioned, the disk directory it displays is nondestructive (i.e., you can view the directory without erasing the active document). However, the directory only displays file names (not types, such as PRG or SEQ), and it only displays these names as a separate option. The result is that you sometimes forget whether a certain file might be sequential or PRG (Paperclip can create PRG files -- MWP doesn't like them at all) and if you try to load a PRG file, MWP responds with a cryptic "64" (file type mismatch). Ideally, the file names would be displayed on the screen with their types, and all the names would automatically display on the LOAD selection screen as well as on its own option.

MWP's Big Bug: You finish editing your massive (4-page -- 16K max) document. You hit COM Q to return to the main menu in order to save it. You accidentally hit 3 (LOAD) instead of 2 (SAVE). Well, no problem, you hit COM Q to get out of LOAD. Just to make sure nothing went wrong, you go back into EDIT (1). Voila! Your active document has been deleted! Goodbye, 16K of text! I'm seriously hoping that Mirage's enhancement diskette has the fix to this STUPENDOUS blunder.

Just plain bugs: Advanced print doesn't work. Period. You need it to print with doublespacing or to produce customized form letters. You get to the final menu after loading the advanced print modules from the distribution diskette and setting up all the parameters, and then the whole system just dies with a cryptic error message and you're back in BASIC. Not only that, even if it did work, it seems to leave you in BASIC anyway after it's finished, rather than reloading MWP.

Good features: 80-column preview without hardware. However, I would have settled for them leaving this out if it meant I had more RAM for the document. MWP is written entirely in machine code. The global search and replace is quite nice, as is the database insert idea (due to advanced print problems, I was not able to verify that it worked, so it's still just an idea) which allows use of inserted data from external sequential files in any order.

I must admit, though, that I like the basis of MWP. It rivals the capability of more expensive word processors running on much more expensive machines; and so what's a few bugs on a word processor that's cost you less than $1,000 complete?

DATA BASE MANAGER

... from page five

that we omitted a field, or mispelled some crucial section With the MERGE FILES option, your problems are solved! Merely create a new form, and command the programme to 'merge' individual fields in your old database to specific fields in your new form. In addition, you may use this function to merge one database with another, or a number of databases into one. This has proved to be a wonderful timesaver, especially as I have expanded a number of my forms on numerous occasions, and the thought of retyping all my data would probably have given me more than second thoughts...

Two other functions shall be briefly mentioned: CREATE SEQ (Sequential File) and REPLACE FIELDS. Many word processors (such as Word Processor by Mirage Concepts) have the ability to place variables in a text, then retrieve the values for those variables when printing a document (a form letter to go to a number of people, for example). The majority of these word processors require a standard sequential file for this purpose and the CREATE SEQ function generates this file from the relative file used by Database Manager.

Unfortunately, I have not had a great deal of success with the REPLACE FIELDS function of Database Manager. The intent is to allow replacement in a field with a set value (for example, replace all occurences of the word 'Solicitor' in a field with 'Lawyer', or replace all occurences of 'Grade 11' in a field with 'Grade 12' if someone's average grade was above C-). However, my programme seems to take great delight in destroying my database if I attempt to use this function -- something I am sure will be remedied in future versions of the programme, as this would be a very useful function if it worked! Which reminds me: ALWAYS BACK UP YOUR DATABASE DISKS!!!

Actually, I am pretty sure the defect in REPLACE FIELDS will be remedied shortly. An earlier version of Database Manager could only display approximately the first 38 fields of any form, although the data was still present, and would refuse to display any field character in the second position from the left side of the screen. These problems do not occur in the current version of Database Manager.

As you can tell, I am quite impressed with Database Manager and find it a pleasure to use. I have not even mentioned the various and powerful PRINT functions, which should be suffiient for almost every purpose (if not, there is an ADVANCED REPORT GENERATOR available from Mirage Concepts), nor the superb manual that comes with the programme, nor the excellent application of the use of the function keys. As far as I am concerned, Database Manager is one of the best database management programmes available for the Commodore 64.

USER FRIENDLIES

By LARRY PHILLIPS

Everyone has an opinion as to what constitutes user friendliness in a program. Some will refer to ease of use, clear instructions, etc. True, a program may be said to be user friendly if it includes these features, but is there more to it than that? I think so, and I'd like to speak about another aspect of friendliness in a program.

Take, for example, the various messages that tell the user how he is doing. I personally dislike a game program that insults me when I do poorly, then only says 'YOU WIN' or 'GOOD' when I do well. This is especially poor practice when a program is designed to be educational as well as fun. The rewards for doing well should always be stronger than the admonishments for poor performance. I'm not saying that you shouldn't let the user know how he stands, but why not include some positive encouragement in the message?

The next type of program I'd like to address is the one that has only two responses. A user becomes a little bored with seeing the same two responses all the time. If you have ever used a program written by someone with a good sense of humour you will know exactly what I mean. The author with a sense of humour will not be satisfied with only two responses, no matter how funny, clever or cute. He will include a variety of responses for each level of accomplishment. Implementing this variety is not at all difficult. Let's look at one technique.

Early in the program, you can 'READ' strings into two string arrays. Let's call these R$(X) and W$(x), for RIGHT and WRONG, respectively. We'll do it like this:

```
10 FOR X = 0 TO 10: READ R$(X):
   NEXT X
500 DATA "YOU'RE GOING TO WHOMP
   ME!"
510 DATA "YOU'VE PLAYED THIS
   BEFORE."
520 DATA "I THOUGHT I HAD YOU
   THAT TIME"
```

etc... for a total of 10 responses.

Followed by:

```
600 DATA "CLOSE... BUT NO
   CIGAR!"
610 DATA "YOU'VE GOT TO BE
   KIDDING!"
620 DATA "WHOOPS! BETTER TRY
   AGAIN."
```

etc... for a total of 10 responses.

Now, in the main part of the program, if the answer is correct, just:

```
PRINT R$(RND 1 * 11)
```

or, if the answer is incorrect:

```
PRINT W$(RND 1 * 11)
```

This technique may be applied to games, educational or otherwise, as well as many other programs. There is one caution, however, and that is that you should never use varied responses when asking the user to enter options. A program becomes very UNfriendly when you have a different display each time you go through the 'setup' procedures, and causes the user to be more prone to mistakes that could be disastrous in a utility or business program. Besides, we all know that businessmen go out of their way to avoid any kind of fun while computing. Just kidding! Really!

GIVE US A BREAK!

This will be the last issue of the paper until September, when there will be a (hopefully) larger edition for the Pacific Coast Computer Fair. Having put out seven issues since last November (yes -- believe it!) has proven to be quite a task for our fearless publications staff who have (should that read "has"?) received many excellent contributions from club members.

If need requires it, there might be a one-page instant printing-style handout at one of the July meetings to keep you abreast of club activities. In the absence of that, remember to call the club's 24-hour answer phone -- PET-3311 (738-3311) for up-to-date club information.

-- 64 LIBRARY NEWS --

The 64 library now contains a program which will copy an entire disk using a single drive in about 4 minutes. In order to use this program, you must unplug all printers and similar peripherals from your disk drive. You should also poke the screen so it is black with POKE 53281,0. When the program runs, the red LED on the front of the drive does not go out when you are inserting or removing the source and "target" disks. While the actual copying is going on, the LED keeps flashing very quickly. When the program ends, the only way to get out of it is by turning off both computer and disk drive (after removing your disk, of course).

```
"CCC LIBRARY        " K1 2A
12  "LIST-ME FIRST "    PRG    7   "MORT SCHED.C"      PRG
27  "INTEREST.C"        PRG    6   "MORT CALC.C"       PRG
21  "BOOKKEEPING.C"     PRG    12  "MORTGAGE.Z"        PRG
30  "LUMP SUM.C"        PRG    23  "MEMORANDA.C"       PRG
23  "BOND YIELD.C"      PRG    13  "INVOICER.C"        PRG
4   "FICA TAX.C"        PRG    7   "DATES.C"           PRG
9   "STOCK OPTION.C"    PRG    5   "MILEAGE.C"         PRG
11  "STOCK LIST.C"      PRG    8   "MARKS.C"           PRG
13  "LOAN.C"            PRG    10  "GROWTH RATE.C"     PRG
28  "IRES CASH FLOW.C"  PRG    4   "DAY OF WEEK.C"     PRG
12  "INVESTMENT.C"      PRG    6   "CALENDER.C"        PRG
11  "INVENTORY.C"       PRG    3   "TYPING TEST.C"     PRG
4   "GROWTH CALC.C"     PRG    12  "BUSPRCASHROI.C"    PRG
14  "DECISION MAKER.C"  PRG    24  "APARTMENT.C"       PRG
10  "FINANCIAL CALC.C"  PRG    19  "COPS BASE TAPE.C"  PRG
12  "PORTFOLIO.C"       PRG    20  "COPS BASE DISK.C"  PRG
7   "MORTGAGE.C"        PRG
230BLOCKS FREE.
```

```
"CCC LIBRARY        " K2 2A
13  "MPRM SCREENS"      SEQ    19  "CATLOG.IMG"        PRG
87  "MPRM.DOC"          PRG    6   "CMBOOT.IMG"        PRG
67  "MPRMEXAMPLE"       PRG    7   "GRFUTL.IMG"        SEQ
15  "MPRMSEQ"           PRG    1   "HO.IMG"            PRG
74  "MPRMUTIL"          PRG    11  "PRTERR.IMG"        PRG
30  "MPRM"              SEQ    7   "SNDUTL.IMG"        SEQ
15  "DOC/64.LP"         PRG    3   "SEQ LISTER"        PRG
24  "BINGO"             PRG    3   "ACCEPT.INF"        SEQ
6   "SPRITE GEN"        PRG    7   "ACCEPT.IMG"        PRG
8   "SUPER DIRECTORY"   PRG    3   "FORMAT.INF"        SEQ
11  "DISK EDITOR"       PRG    17  "FORMAT.IMG"        PRG
9   "DIR ORGANIZER"     PRG    12  "CLUE1"             PRG
33  "DISK MASTER"       PRG    38  "CLUE TWO"          PRG
1   "COMAL---STUFF---"  SEQ
150BLOCKS FREE.
```

VIC LIBRARY ADDITIONS

Following is a list of the most recent VIC-20 tapes and disks. TAPE 21 will not be available at the June workshop meeting.

The VIC library is in need of someone to take over the making of tapes and disks. Requirements, aside from both a disk drive and cassette are patience, time and organizational abilities.

```
"CCC VIC TAPE 17 " 17 2A
12  "DIRECTORY       U" PRG    11  "BASE CONVERTER A" PRG
7   "PRG CLASS VIC   U" PRG    2   "BINARY QUIZ    E" PRG
6   "MAIL LIST       B" PRG    1   "BINARY TABLE   E" PRG
3   "BUDGET          B" PRG    8   "VIC FINANCE    B" PRG
6   "FUNCTION KEYS   U" PRG    4   "CHAR MAGNIFIER P" PRG
2   "SQUARED         E" PRG    7   "LINE PRO       B" PRG
7   "FILE CLERK      B" PRG    11  "DISASSEMBLER   A" PRG
2   "ALPHABETIZER    U" PRG    8   "JOTTO          G" PRG
4   "MEMRY SNAPSHOT  A" PRG    35  "COMPUTE FILE   U" PRG
4   "MULTIPLY        E" PRG    36  "HAUNT HOUSE 8K G" PRG
7   "LONG DIVISION   E" PRG    12  "FUNKY MONKEY   G" PRG
3   "FLASH PROMPT    P" PRG    10  "BAR BATL JS/KB G" PRG
13  "SPEED TYPE      E" PRG    10  "BAR BATL 2JS   G" PRG
7   "8 X 8 CREATE    U" PRG    8   "MONITOR        A" PRG
7   "PROG CHARS      U" PRG    10  "VIC DT"           PRG
2   "VISIBLE VIC     P" PRG
399BLOCKS FREE.
```

```
CCC VIC TAPE 18 " 18 2A
12  "DIRECTORY       U" PRG    19  "L2"               PRG
7   "PRG CLASS VIC   U" PRG    12  "CAVE-IN JS     G" PRG
10  "RAND. ANALYSIS  D" PRG    8   "SOUND STORY    P" PRG
12  "UNSCRATCHER     U" PRG    6   "ROADER         G" PRG
7   "ALPH-DIRECTORY  U" PRG    14  "TYPING DERBY   E" PRG
8   "TINY ALPH-DIR   U" PRG    7   "DUNGEON JS     G" PRG
10  "VARIABLE LISTR  U" PRG    13  "D"                PRG
2   "WORM 8K JS      G" PRG    12  "TREE TUTOR     E" PRG
23  "NM"                PRG    13  "SNERTLE        E" PRG
11  "MATHQUIZ        E" PRG    12  "PENTONIMOS 8K  P" PRG
14  "QUATRAIN 8K JS  G" PRG    9   "FISHERMAN JS   G" PRG
11  "FAST ADD        E" PRG    11  "FRANTIC2        " PRG
11  "MATCHEM         E" PRG    10  "VIC DT"           PRG
1   "LADDERS 8K JS   G" PRG
379BLOCKS FREE.
```

```
"CCC VIC TAPE 19 "  19 2A
11  "DIRECTORY"         PRG    3   "FR. ACC. PRACT."   PRG
13  "SNERTLE"           PRG    5   "BUG RACE"          PRG
5   "FIRST MATH"        PRG    6   "ALPHA-BLAST"       PRG
12  "BIG MATH"          PRG    14  "CONNECT THE DOTS"  PRG
13  "MUNCHMATH"         PRG    12  "HEARTS"            PRG
13  "MATHMAN TIMES"     PRG    10  "AARDVARK ATTACK"   PRG
13  "MATHMAN MINUS"     PRG    11  "ALPHA-SHOOT"       PRG
11  "ELECTRO-TIMES"     PRG    12  "TREE TUTOR"        PRG
11  "ELECTRO-PLUS"      PRG    8   "KEYBOARD"          PRG
11  "FAST ADD"          PRG    10  "CHOOSE THE NOTE"   PRG
7   "LONG DIVISION"     PRG    4   "VIC PIANO"         PRG
11  "MATHQUIZ"          PRG    9   "HANGMAN"           PRG
10  "FRENCH"            PRG    12  "WORD MATCH"        PRG
12  "F"                 PRG    10  "VIC DT"            PRG
385BLOCKS FREE.
```

```
"CCC VIC TAPE 20 "  20 2A
12  "DIRECTORY"         PRG    14  "MICROS-5"          PRG
7   "PRG CLASS VIC   U" PRG    14  "MICROS-6"          PRG
8   "1-TOUCH KEYWDS  U" PRG    2   "CHECK STUB     B" PRG
3   "CASSETTE-BEEP   U" PRG    13  "ASTRO SHOWER   G" PRG
22  "THERAPY 8K      G" PRG    10  "COPYCAT        G" PRG
18  "SHAPE MATCH 8K  E" PRG    7   "SPACE DRIVE    G" PRG
13  "CYON REVNG JS   G" PRG    11  "ROCKS          G" PRG
14  "MICROS-1        E" PRG    9   "BRICK OUT      G" PRG
13  "MICROS-2"          PRG    41  "THIS & THAT 8K G" PRG
14  "MICROS-3"          PRG    10  "VIC DT"           PRG
14  "MICROS-4"          PRG
393BLOCKS FREE.
```

```
"CCC VIC TAPE 21 "  21 2A
12  "DIRECTORY"         PRG    37  "FANTASYLND 16K G" PRG
7   "PRG CLASS VIC   U" PRG    20  "CRAPS 8K       G" PRG
6   "ASTROGLADIATOR  G" PRG    18  "LIFE SPAN 8K"  G  PRG
8   "AST2"              PRG    12  "MORSE MAKER    E" PRG
12  "PACK MAN        G" PRG    16  "MAKE DECIS. 8K B" PRG
11  "ALIEN           G" PRG    16  "DOMINOES 8K    G" PRG
12  "MAZEMAN         G" PRG    12  "TRON           G" PRG
10  "FIREBALL        G" PRG    5   "MICROMON INST."   PRG
10  "VIC TERMINAL    C" PRG    17  "MICROMON@@3000@K" PRG
7   "TERM.VIC"          PRG    10  "VIC DT"           PRG
8   "SPACE DODGE     G" PRG
398BLOCKS FREE.
```

VIC-20 DISKS

```
"CCC VIC AK      " MK 2A
4   "VIC WEDGE       U" PRG    12  "FUNKY MONKEY   G" PRG
7   "PRG CLASS VIC   U" PRG    10  "BAR BATL JS/KB G" PRG
4   "DIRECTORY,      U" PRG    10  "BAR BATL 2JS   G" PRG
8   "MONITOR         A" PRG    14  "QUATRAIN 8K JS G" PRG
6   "MAIL LIST       B" PRG    11  "MATHQUIZ       E" PRG
3   "BUDGET          B" PRG    10  "RAND. ANALYSIS D" PRG
6   "FUNCTION KEYS   U" PRG    12  "UNSCRATCHER    U" PRG
2   "SQUARED         E" PRG    7   "ALPH-DIRECTORY U" PRG
7   "FILE CLERK      B" PRG    8   "TINY ALPH-DIR  U" PRG
2   "ALPHABETIZER    U" PRG    10  "VARIABLE LISTR U" PRG
4   "MEMRY SNAPSHOT  A" PRG    1   "WORM 8K JS     G" PRG
4   "MULTIPLY        E" PRG    23  "NM"               PRG
7   "LONG DIVISION   E" PRG    12  "PENTONIMOS 8K" P  PRG
3   "FLASH PROMPT    P" PRG    9   "FISHERMAN JS   G" PRG
13  "SPEED TYPE      E" PRG    11  "FRANTIC2"         PRG
7   "8 X 8 CREATE    U" PRG    11  "FAST ADD       E" PRG
7   "PROG CHARS      U" PRG    11  "MATCHEM        E" PRG
2   "VISIBLE VIC     P" PRG    1   "LADDERS 8K JS  G" PRG
11  "BASE CONVERTER  A" PRG    19  "L2"               PRG
2   "BINARY QUIZ     E" PRG    12  "CAVE-IN JS     G" PRG
1   "BINARY TABLE    E" PRG    8   "SOUND STORY    P" PRG
8   "VIC FINANCE     B" PRG    6   "ROADER         G" PRG
4   "CHAR MAGNIFIER  P" PRG    14  "TYPING DERBY   E" PRG
7   "LINE PRO        B" PRG    7   "DUNGEON JS     G" PRG
11  "DISASSEMBLER    A" PRG    13  "D"                PRG
8   "JOTTO           G" PRG    12  "TREE TUTOR     E" PRG
35  "COMPUTE FILE    U" PRG    13  "SNERTLE        E" PRG
36  "HAUNT HOUSE 8K  G" PRG    15  "SINGLE DRV COPY"  PRG
143BLOCKS FREE.
```

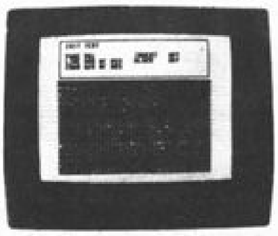
```
"CCC VIC AL      " ML 2A
4   "VIC WEDGE       U" PRG    9   "BRICK OUT      G" PRG
7   "PRG CLASS VIC   U" PRG    6   "ASTROGLADIATOR G" PRG
4   "DIRECTORY       U" PRG    8   "AST2"             PRG
8   "1-TOUCH KEYWDS  U" PRG    12  "TRON           G" PRG
3   "CASSETTE-BEEP   U" PRG    12  "PACK MAN       G" PRG
22  "THERAPY 8K      G" PRG    11  "ALIEN          G" PRG
18  "SHAPE MATCH 8K  E" PRG    12  "MAZEMAN        G" PRG
13  "CYON REVNG JS   G" PRG    10  "FIREBALL       G" PRG
14  "MICROS-1        E" PRG    7   "TERM.VIC"         PRG
13  "MICROS-2"          PRG    10  "VIC TERMINAL   C" PRG
14  "MICROS-3"          PRG    8   "SPACE DODGE    G" PRG
14  "MICROS-4"          PRG    37  "FANTASYLD 16K  G" PRG
14  "MICROS-5"          PRG    20  "CRAPS 8K       G" PRG
14  "MICROS-6"          PRG    18  "LIFE SPAN 8K   G" PRG
2   "CHECK STUB      B" PRG    30  "MAKE DECIS. 8K B" PRG
13  "ASTRO SHOWER    G" PRG    16  "DOMINOES 8K    G" PRG
41  "THIS & THAT 8K  G" PRG    17  "MICROMON@@3000@K" PRG
10  "COPYCAT         G" PRG    12  "MORSE MAKER    E" PRG
7   "SPACE DRIVE     G" PRG    15  "SINGLE DRV COPY"  PRG
11  "ROCKS           G" PRG
140BLOCKS FREE.
```

```
"CCC VIC AM (ED1)" MM 2A
4   "VIC WEDGE"         PRG    5   "BUG RACE"          PRG
4   "DIRECTORY"         PRG    6   "ALPHA-BLAST"       PRG
13  "SNERTLE"           PRG    14  "CONNECT THE DOTS"  PRG
5   "FIRST MATH"        PRG    12  "HEARTS"            PRG
12  "BIG MATH"          PRG    10  "AARDVARK ATTACK"   PRG
13  "MUNCHMATH"         PRG    11  "ALPHA-SHOOT"       PRG
13  "MATHMAN TIMES"     PRG    12  "TREE TUTOR"        PRG
13  "MATHMAN MINUS"     PRG    8   "KEYBOARD"          PRG
11  "ELECTRO-TIMES"     PRG    10  "CHOOSE THE NOTE"   PRG
11  "ELECTRO-PLUS"      PRG    4   "VIC PIANO"         PRG
11  "FAST ADD"          PRG    9   "HANGMAN"           PRG
7   "LONG DIVISION"     PRG    12  "WORD MATCH"        PRG
11  "MATHQUIZ"          PRG    10  "VIC DT"            PRG
10  "FRENCH"            PRG    15  "SINGLE DRV COPY"   PRG
12  "F"                 PRG    5   "VIC AM"            PRG
3   "FR. ACC. PRACT."   PRG
368BLOCKS FREE.
```

EXPAND YOUR HORIZONS

MIRAGE CONCEPTS, INC.

Personal Word Processor



The Word Processor — Personal Version makes writing easier than ever before. It will enable you to put your ideas into printed form faster and better than you ever thought possible. For new owners, and those with basic needs, this program is the perfect solution! Upgradeable to Prof. Version.

Outstanding Features
- 100% machine language
- True word wrap
- Printed page/line/character counters
- Accomodates all printer functions
- Merge with Mirage Database Manager

Specific Features

Cursor Movement
- Up/down: character, line, screen
- Left/right: character, word, file

Delete
- Up/down: line
- Left/right: character, word, file
- Entire: file

Insert
- Character

Block Operations
- Block: move, delete, print

Screen Control
- Menu driven • On screen formatting
- Automatic and manual text reformatting
- Indent paragraph

File Operations
- Finished editing (save, no save)
- Optional back-up • Disk directory
- Delete file

Line Formatting
- Justify, center, multiple line spacing
- * Half line spacing

Printing (Full Macro)
- * Underscore * Double underscore
- * Bold face * Super/Sub script
- * Overstrike * Strikeout
- * Ribbon shift * Proportional print
- * Set character/lines per inch
- Left, right, top, bottom margins
- Print with or without saving
- Print file from disk
- Pause for Paper • Chain files

Page Control
- Print selected page, range of pages
- * Force new page

Editor
- Change colors (screen, text, edit line)
- Uses highlighting, function keys
- Audible key feedback (on/off)

*IF PRINTER IS CAPABLE

Suggested List Price: $124.95

Professional Word Processor

UNITED STATES COMMODORE USERS GROUP APPROVED

For computer owners with more than just basic needs in the area of word processing, Mirage offers the Word Processor—Professional Version. This package adds scores of features to our less expensive program, making it the most versatile word processor currently available for the Commodore 64.

The 'Professional' Word Processor has all the functions of the 'Personal' version plus these additional, outstanding features:

- 80 columns w/o additional hardware
- 30,000 word SPELLING CHECKER
- Global search & replace
- Extended block functions
- Column movement

Specific Features

Cursor Movement
- Left/right: edit line, sentence, paragraph, tab, block

Delete
- Up/down: screen
- Left/right: edit line, sentence, paragraph,
- Entire: edit line

Insert
- Spaces, line/multiples, from external file

Search and Replace
- Find string • Ignore case
- Find/Replace (all occurances, prompted)
- Match (any character-wild, whole words)

Block/Column Operations
- Block: copy, indent, save, read
- Column: move, erase

Screen Control
- Tabs (set, clear, clear all)
- Tab movement (forward, backward)

File Operations
- Merge files

Printing (Full Macro)
- Print more than one copy
- Save print format

Page Control
- Page length
- Headings/footings (single or multiple)
- Page numbering (regular, odd/even)

Suggested List Price: $124.95

Data Base Manager

UNITED STATES COMMODORE USERS GROUP ENDORSED

Suggested List Price: $124.95

The Database Manager is a comprehensive electronic filing system that makes it easy to organize, maintain, and use all the information you must handle in your home, school, or business. It will store, search, sort, retrieve, display, calculate, and print reports, lists, and even mailing lists.

Outstanding Features
- 100% machine language
- Free form design and input
- Sort on any field/any level
- Calculated fields
- Merge with Mirage Word Processors

Specific Features

Overview
- Single load • Screen oriented
- Menu driven • On screen prompting

Structure (Maximums)
- Records/file: limited only by disk size
- Characters/record: 2,000 • Fields/record: 200
- Characters/field: 250
- Characters/form: 2,500 (60 screen lines)

Input/Edit Functions:
- Free form design • Free form data input
- Full use of cursor and function keys
- Review record: previous, next
- Search access time (with index): 1 second
- Search does not require whole field
- Conditional statements
- Edit during review (forward/backward)
- Global search and replace
- Calculated fields (10/file) cont. update
- Right justify (auto on numeric fields)

Sort/Index Functions
- Sort: any field, any level
- Indexes can be saved to disk

File Manipulation
- Append from keyboard or another file
- Append select fields or all fields
- Change file structure w/o loss of data
- Create subfiles using conditionals
- More than one file on single disk
- Only one disk needed per database file
- Create new data disk
- Disk directory • Delete file
- Data accessible from BASIC file

Print/display functions
- Display/print file (original form or list)
- Reports: column headings, page numbers, page breaks, centered titles
- Sorted order option, decimal digit option
- Further reports: Advanced Report Generator

Mirage Concepts' Power Packed Software

Advanced Report Generator

The Advanced Report Generator is an extremely flexible accessory program which uses the data files created by Mirage's Database Manager to produce the "special application" forms and reports so common to home, school, and business environments. It must be used in association with the Database Manager.

Outstanding Features
- 80 column form reports
- Summary reports
- Additional calculation & sorting ability
- Printer commands & saved report formats
- File subtotals & totals

Specific Features

Overview
- Single load • Menu driven
- Screen Oriented • On-screen prompting

Report Style
- Columnar reports: titles, field headings page numbering, subtotals, totals
- Form reports: specify row/column, extra text (anywhere), variable page size, totals

Sorting, Conditionals, & Field Matching
- Sorts: ascending, descending
- Print selected records
- Wild card match

Calculated Fields
- Create new calculated fields

Field Formatting
- Fixed decimal • Floating dollar sign
- Text field • Accounting format

Totals & Subtotals
- Subtotal on sort field
- Summary reports

Text Fields (Form Format)
- Print text anywhere on page

Special Printer Considerations
- Issue printer commands

Suggested List Price: $69.95

IRISH OAK SOFTWARE & PERIPHERAL'S DISTRIBUTORS LTD

#1-2475 MANITOBA ST.
VANCOUVER B.C.
V5Y 3A4
Tel.(604)875-6238

Your Representative:

OTTO KELLER

DEALER LISTING:

ACE SOFTWAREHOUSE
E. Broadway, Kingsway, Bby.
BROADWAY COMPUTER SUPPORT
Vancouver
COMPUSOFTWARE
Broadway, Vancouver
FUTURE SHOP
Broadway, Vancouver; Coquitlam, Richmond
KELLY'S
Lansdowne, Guildford, Park Royal, 810 Granville, Vancouver
SOFTSTUFF
Canada Way, Burnaby
SUPER SOFTWARE
No. 3 Rd., Richmond
WOODWARD'S
Hastings, Oakridge, Lansdowne, New West., Guildford, Coquitlam, Park Royal