Two Exciting Games:
Missile Math For Apple,
 IBM PC And PCjr, Atari,
 Commodore 64, TI-99/4A,
 And VIC-20
Lightsaver For
 Commodore 64, IBM PC
 And PCjr, VIC-20,
 And Atari

Lightning And
Bulldozer Sort:
For Apple, Commodore 64,
VIC-20, IBM PC And PCjr,
And Atari
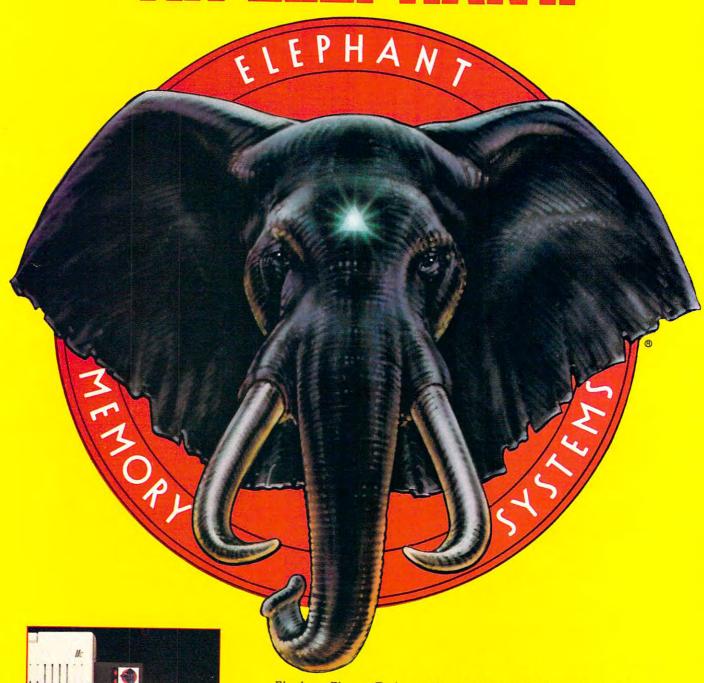
Autoboot For
Commodore 64
And VIC-20: Load Programs
And They RUN Themselves

Atari Speed-Reading:
Improve Your Speed
And Comprehension

09

0

71486 02193

TEST DRIVE AN ELEPHANT.™
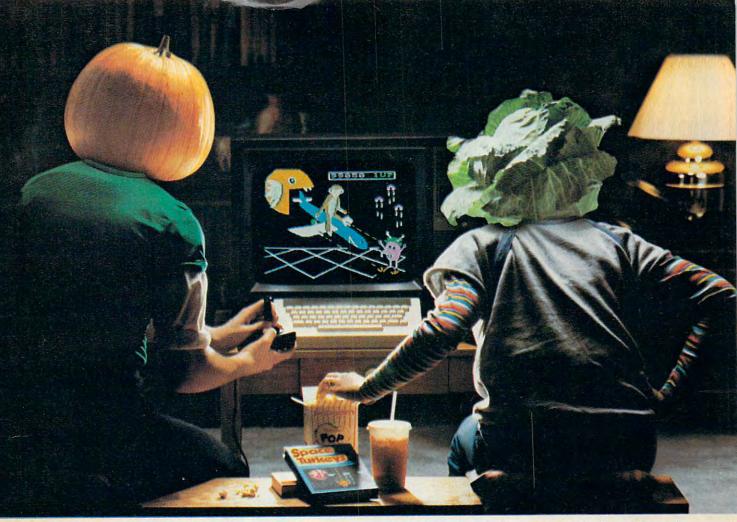
ELEPHANT

MEMORY SYSTEMS

Elephant Floppy Disks are the perfect vehicle for storing and pro-
tecting data. Because Elephant never forgets. You'll get high performance
that's 100% guaranteed for a lifetime of heavy use. So take them for a test
drive. They're available now at your local computer showroom. And
there's no waiting for delivery. For the Elephant dealer nearest you, call
1-800-343-8413. In Massachusetts, call collect (617) 769-8150.

Dennison **ELEPHANT™NEVER FORGETS.**

You bought a computer to cultivate your kids' minds. Make sure it's bearing fruit, not growing vegetables.

Introducing a whole crop of Learning Adventure games from Spinnaker.

When it comes to cultivating adventurous young minds, the computer's potential is endless.

Unfortunately, the search for software that makes the most of that potential has been endless, too.

That is, until Spinnaker created the Learning Adventure Series. A unique collection of games that reward curiosity with hours of adventure and learning. So the time kids spend with our games will help them develop valuable skills. Instead of just tired thumbs.

But what really makes our Learning Adventure games unique—educational value aside—is how much fun they are. Which isn't too surprising when you consider you can do things like bargain with aliens, search a haunted house, or build your own railroad empire.

In fact, our games are so much fun, kids will really enjoy developing some very important skills. Deductive reasoning, note taking, and problem solving, for instance.

So, if you're in the market for software that will truly cultivate young minds, pick the Spinnaker Learning Adventure Series.

It's the best way to be sure your search will be fruitful.

Spinnaker Learning Adventure games are available for Apple,® Atari,® IBM® and Commodore 64™ home computers.
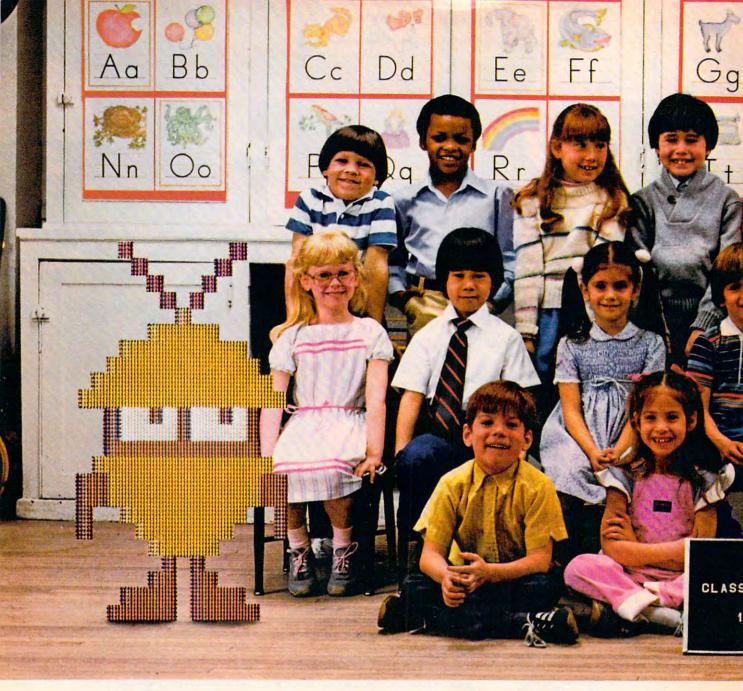
It's New! TRAINS.™
You're in charge of an old-time railroad—and whether it turns into a bonanza or a bust depends on how well you run it. But either way you'll find that working on this railroad is a challenge—and a lot of fun! **Ages 10-Adult.**

It's New! ADVENTURE CREATOR.™
Design a challenging adventure game that you or a friend can tackle—or let the computer design one for you. It's complex, exciting—utterly addictive! **Ages 12-Adult.**

IN SEARCH OF THE MOST AMAZING THING.™
It isn't easy to find—even in your B-liner. But you'll have help from your Uncle Smoke Bailey as you search the universe to find the Most Amazing Thing. **Ages 10-Adult.**

SPINNAKER™
We make learning fun.

Disks for: Apple, Atari, IBM, and Commodore 64.
Cartridges for: Atari and Commodore 64 — (ADVENTURE CREATOR only).

www.commodore.ca

Modern times have create

We've cleverly disguised them as funny creatures from a computer game. But underneath the funny exterior is one of the most serious approaches to home education you've ever heard of.

INTRODUCING SPROUT™ SOFTWARE. GAMES THAT TEACH.

Your kid's new teachers are called Tink and Tonk. They come from Sprout. Software for kids 4 to 8.

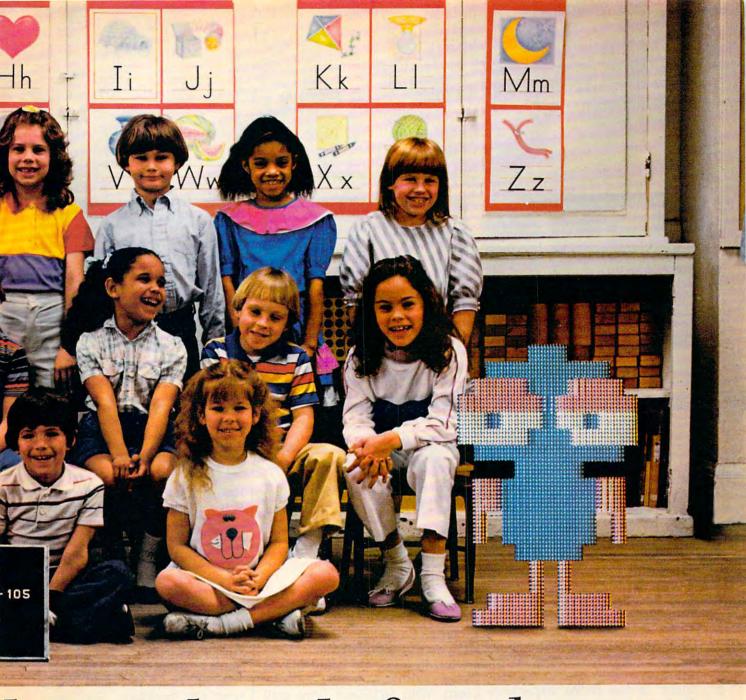The beauty of Sprout software is how entertainment is balanced with a healthy dose of education.

While kids are having fun at home, they're actually reinforcing what they've learned at school. Things like the alphabet, spelling, vocabulary, counting, adding, and pattern recognition.

You'll also like how Sprout prevents boredom. Our games grow up, instead of wear out. As kids get older, the game adjusts and gets harder. Because there are many

d a new breed of teachers.

variations and many decisions to make.

Knowing how to do all this isn't something Sprout learned overnight.
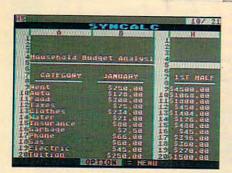
You see, we've got a hundred years of experience to lean on. (Our parent company is SFN, the country's #1 textbook publisher for elementary and high schools.)

We've also got the expe-

rience of Mercer Mayer, who has written or illustrated 80 children's books. He dazzles kids with ideas and pictures that keep them coming back for more.

With TINK!TONK!™ software, kids see that learning can be more fun than destroying space creatures.

sprout

Games that grow up. Instead of wear out.

Compatible with Atari®, Commodore®, Apple®, and IBM.®

NOTE: See page 69 before typing in programs.

TOLL FREE Subscription Order Line
800-334-0868 (In NC 919-275-9809)

www.commodore.ca

EDITOR'S NOTES

We recently received a letter from a long-time subscriber that thoughtfully raises some concerns about the present composition of COMPUTE!. It is excerpted and addressed here.

"Dear Mr. Lock,
I felt compelled to write to you to express my feelings about the disappointing changes which have been coming over my once favorite magazine, COMPUTE!. I have been reading COMPUTE! since January 1981 and have a subscription which is paid through August 1985.

In recent months, since January 1984, you have been 'spread too thin' by attempting to cover so many types of computers that I feel none have been done adequately. The technical content of the magazine, for the Commodore computers, and computer use in the home, is what had interested me the most, and the quantity and the quality have dwindled so much . . .".

Our reader goes on to discuss concerns in specific about the breadth of our articles, and the size of our columns. We thought this presented a good opportunity to address these concerns and hopefully explain some of what we see happening from here, and where it will take us.

The Case Of The Dwindling Magazine

In December of 1983, COMPUTE! had 392 pages. This issue, September 1984, has 160. The December issue had 215 pages of advertising; this issue has 60.5. During the course of 1984, we've seen a massive shakeout, not unexpected of course, in the industry. Where there once was an advertiser base of thousands of companies, there is now an advertiser base measured in the hundreds. All well and good, you may say, but is it really a problem or concern for you, the readers? In the sense that we must attend to some economic realities in planning and publishing our magazines, the answer is yes. Do we subjugate our concerns to formula? No, absolutely not.

A typical rule of thumb for the publishing business is a 50/50 advertising mix. As size increases, this ratio gives over to an increased percentage of advertising. In our case, the December 1983 issue of COMPUTE! was over 55 percent advertising. The September issue, by rule of thumb, should be at most 128 pages. Given the overhead we carry in fixed page content (i.e., everything from columns to MLX), this was not acceptable. The result is that our editorial percentage in this issue approaches 65 percent.

The Future

Certainly all of this makes sense, but it still doesn't solve the problem/question at hand: more content. What else can we do? We're working on it. In our "fixed overhead" areas, we're whittling down column sizes. We're trying to expand the scope of some columns so they're more useful to more of you. And we're evaluating all of our columns with an eye toward further reductions.

One of your complaints, in essence, is that some of the "meat" of our content is diminished. Upon reflection, I think that's an offshoot of our attempts to provide continued breadth. We probably have a tendency to run shorter articles to enhance variety. The unfortunate by-product of this is that some of the more technical, lengthy articles are bypassed. This we can address immediately, and we will begin to do so with our October issue. We would welcome some additional input from readers. One suggestion here has been that we start to run some of our "fixed" material on an every-other-issue basis. This would mean that "Beginner's Guide To Typing In Programs," for example, might appear on alternate months. We are open to your thoughts. Your comments?

Robert Lock

Editor In Chief

Coming In Future Issues

The New Wave Of Computer Games

Chess: A Sophisticated Program That Pits You Against Your Computer

Two Exciting Games: Space Caverns And Horse Racing

All About The Status Register

Apple Disk Checker

Commodore Unicopy

Multiscreen Atari Animation

TI Disassembler

COMPUTE! Publications, Inc. publishes:

COMPUTE!
COMPUTE!'s GAZETTE
COMPUTE! Books
COMPUTE!'s GAZETTE DISK

Corporate Office:
324 West Wendover Ave., Suite 200
Greensboro, NC 27408 USA

Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA
Telephone: 919-275-9809

Subscription Orders

COMPUTE! Circulation Dept.
P.O. Box 914
Farmingdale, NY 11737

TOLL FREE Subscription Order Line
800-334-0868   In NC 919-275-9809

COMPUTE! Subscription Rates
(12 Issue Year):

US			Air	
	(one yr.)	$24	Europe, Australia	$42
	(two yrs.)	$45	Middle East, Central	
	(three yrs.)	$65	America and North	
Canada and Foreign			Africa	$52
Surface Mail		$30	South America, South	
			Africa, Far East	$72

Advertising Sales

1. New England
Al Dalton
David Fay
617-451-0822

2. Mid Atlantic
Sharon Brooks
Joe Porter
Kathy Hicks
215-646-5700
212-567-6717 (NY)

3. Southeast & Foreign
Harry Blair
919-275-9809

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/Mountain/Texas
Phoebe Thompson
408-354-5553
Jerry Thompson
415-348-8222

6. Southwest
Ed Winchell
213-378-8361
JoAnn Sullivan
619-941-2313

Director of Advertising Sales
Ken Woodard

COMPUTE! Home Office 919-275-9809.

Address all advertising materials to:
Patti Williams
Advertising Production Coordinator
COMPUTE! Magazine
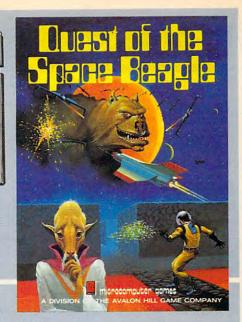324 West Wendover Ave., Suite 200
Greensboro, NC 27408

www.commodore.ca

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

Automatic Conversions?

Many of the programs you publish in COMPUTE! are for several different microcomputers. Do you have utilities that automatically convert a program to another type of computer? Are they available for sale?

J.B. Allen

We call the process of converting a computer program to another computer homogenization. *Unfortunately, we do not have, nor have we seen, any utilities that will automatically homogenize a program.*

We have a staff of talented programmers that manually goes through programs and homogenizes them. The process can be a laborious one, particularly with machine language programs which often require a byte-by-byte translation.

Missing Commodore Ribbons

I am the owner of a Commodore MPS-801 printer. Until recently, I have had trouble finding a replacement ribbon. It seems that the dealers selling the printers are not able to get the ribbons.

I have some good news for you to pass on to your readers. You can find a replacement ribbon at your local Radio Shack store. It is the ribbon for the DMP-110 printer, part number 26-1283. This ribbon is identical to the one for the MPS-801 printer.

Frank W. Fife

Many readers have mentioned difficulty locating replacement ribbons for the 801. Thanks for the tip.

Atari Monitor Hookup

I have ordered a BMC monitor and an Atari 800XL. Very recently, I was told that the Atari is designed for use with a standard TV set. How can I make the Atari work with the BMC monitor. Will the monitor be totally useless?

Benedict V. Sulit

Although optimized for use with a television set, the Atari 800XL looks even better on a composite color monitor. You can also hook up a black-and-white (monochrome) monitor. You'll need a special cable that plugs into the round, five-pin DIN plug on the back of your 800XL. For a color monitor, get a cable that feeds the chroma (composite video) signal into Video In on your monitor. For a monochrome screen, you'll get best results using the LUMA (luminance) signal. Some cables bring out all five pins to RCA phono jacks. With trial and error you can easily find which wire controls which signal. You can get a video cable at your local computer store or possibly a TV/video store.

Musical Apples

I am an Apple IIe user and would like to know how to program my computer to make music. Can you do it in Applesoft BASIC or do you need another language?

Denny Hays

It's a fairly simple matter to make elementary sounds on the Apple in BASIC. You can tweak the speaker by accessing location −16336 (i.e., POKE −16336,0) or ring the bell by PRINTing CHR$(7). But to do much more than this, you need a short machine language (ML) routine. Fortunately, this ML routine can be POKEd in from BASIC.

You can find such an ML routine in an article by Blaine Mathieu (COMPUTE!, October and November 1983). The tutorial explains both fundamental and complex Apple sound generation.

More Open Commodore Files

The "64 Explorer" by Larry Isaacs in the March 1984 issue concerning the maximum number of files which can be open at one time on the 1541 disk drive caught my interest. I have discovered some further information.

The number of sequential files opened for writing can be increased to three if the filename is prefixed with a 0:. When this is not the case, the drive acts as Isaacs described.

Since the Disk Operating System was originally used on a dual disk drive, I always add the prefix 0: to any disk operation to insure that it will work properly.

Mitchell S. Comstock

Autobooting Atari

Is there any way you can make a program run right after it has been loaded into your computer from tape or disk? This would prevent anyone from seeing the special codes in the program.

Mark Zobel

We assume you are using a password or are disabling the break key and system reset. If the program didn't run automatically, anyone could LIST the program and remove the protection. Although you can't completely bar an experienced, enthusiastic invader, you can make a program run automatically from disk when you boot up your system. See "Automate Your Atari" in COMPUTE's Third Book of Atari.

Although it is possible to create a boot tape that does the same thing, we don't know of a specific program. There are techniques for making BASIC programs RUN automatically when LOADed on Commodore machines (for example, having the program load in memory low enough to change some vectors), but the technique is too involved to cover here. See "Commodore Autoboot" elsewhere in this issue.

TI Memory Maps

I write this in reply to the letter from Davin A. Trulsen in the May issue of COMPUTE!. If he or any other TI user wants a comprehensive memory map for the TI, they should write to Miller Graphics, 1475 West Cypress Avenue, San Dimas, CA 91773. Miller Graphics offers a monthly newsletter, called *The Smart Programmer*, for $12.50 a year. They have published four so far and I have found them quite useful.

Bill Grant

Incomplete NEXTs

I have a VIC-20 and I have a question about the NEXT statement. I've seen some programs that had a NEXT statement with nothing after the next. For example:

```
FOR A=1TO10:NEXT
```

Why doesn't it include the variable after the NEXT as in:

```
FOR A=1TO10:NEXT A
```

Kevin Biebor

The NEXT statement increments or completes a loop that was started by a FOR statement. If a variable is placed after the NEXT, that loop is incremented. In the following example, the B loop will be incremented (and completed) ten times each time the A loop is incremented.

```
FOR A=1TO10
FOR B=1TO10
NEXT B
NEXT A
```

If the NEXT statement is not followed by a variable name, the loop completed will be the one most recently started. In the following example, the NEXT will complete the B loop even though the A loop was the first one started.

```
FOR A=1TO10
FOR B=1TO10
NEXT
```

Nested loops (loops within other loops) should be written with care. If they're programmed incorrectly, one or more of the loops may not be completed. For instance, the B loop in the following example will never be completed.

```
FOR A=1TO10
FOR B=1TO10
NEXTA
NEXTB
```

Don't Blame The Hardware

I am having a problem reading arrays on the Atari. This simple program is an example:

```
10 DIM X(5)
20 FOR I=1 TO 5
30 READ X(I)
40 PRINT X(I)
50 NEXT I
60 DATA 3,5,-2,7,4
```

I always get an error in line 30. I wonder if my computer memory has gone bad.

Lloyd R. Holmes

When faced with a particularly stubborn bug, most programmers start to suspect the hardware—but it's almost never the culprit.

As a general programming rule, never assume the hardware is bad except as a last resort. Some program bugs are so obscure, so hard to track down, that it becomes pretty tempting to blame the hardware. Nevertheless, true hardware errors almost always exhibit outrageous behavior, such as lockups when you turn the machine on, screens that suddenly go haywire, an eerie, inexplicable hum when you use SOUND, etc. Hardware errors are obvious, except when just a few bytes of RAM go bad. You can buy memory test programs, but it would probably be an unnecessary investment.

The specific problem you're having here is syntactic. On the Atari, you cannot READ a data item directly into an array. READ X(I) is just as illegal as INPUT X(I). It's a lamentable eccentricity, but it is easy to get around. Just read the item into a temporary variable, then assign that variable to X(I). To wit: READ T:X(I)=T.

Cwww.commodore.ca

And remember The Programmer's Debugging Rule: Hardware problems are as obvious as they are rare.

Commodore Plus/4

Since you published the article on the new Commodore computers (COMPUTE!, April 1984), I have been giving serious thought to purchasing one. I will be enrolling in college, and feel that the new Commodore 264 would be a good choice for my major: electronics. However, I have a number of questions about the 264.

1. I know that the 1541 disk drive works with the Commodore 64 and the VIC-20, but will it also be compatible with the 264?

2. How far can I expand the RAM or ROM memory in the 264?

3. Can I buy an Eprom burner for the 264?

4. Is there any software available for the new 264?

5. Is it possible to increase the baud rate of the 1541 disk drive?

Don Maxwell

The new Commodore Plus/4 (renamed from the 264) has not been released for sale as of this writing. It is, however, expected to hit the retail shelves sometime this fall.

About the same time, Commodore is planning to market a new disk drive, the SFS-481. SF stands for super fast because it has an advertised baud rate of 1675. However, a 1541 disk drive will be compatible with the new Plus/4 as well.

Although the Plus/4 will support (memory) bank switching, Commodore has not announced any memory expansion for the Plus/4. However, third-party companies might eventually offer something.

Regarding EPROM burners, again Commodore has no current market plans, but undoubtedly third-party manufacturers will offer this peripheral. Also, there will be software available for the Plus/4. There already exists a considerable amount of both application and entertainment software.

As to the baud rate change on the 1541 disk drive, a Commodore representative said that the baud rate (the speed that information is transmitted to or from the disk drive) on the 1541 cannot be changed on the drive as is. The current baud rate is dictated by the serial port interface on the computer. In other words, no matter how you modify the disk drive, the computer will only be able to receive data at its preprogrammed rate.

The 1541's rate can be increased by converting it to a parallel interface. But this is a major engineering project, and might prove to be both impractical and costly.

Buying The Right Modem

I am thinking of buying a modem. Do I need a rotary-dial phone, or will modems work with a Touch-Tone system?

Steve Milewski

Most modems will work with both the Touch-Tone and the older rotary phones. Some modems, in fact, work on Touch-Tone phones by emulating the rotary system. The Commodore 1650 auto-dial modem is one.

Before buying, however, it's always best to ask your dealer, or call the modem manufacturer for further information.

Commodore Service

I am an electronics repair technician, and own a Commodore 64. I cannot find anyone in my area who can repair my computer.

Where can I obtain the technical information to service my Commodore computer, and all of its related peripheral equipment?

Gordon Bates

The addresses and phone numbers of the Commodore service centers are as follows:

1200 Wilson Drive
West Chester, PA 19380
(215) 431-9105

2246 North Palmer Drive
Schaumburg, Il 60195
(312) 397-0075

4350 Beltwood Parkway South
Dallas, TX 75234
(214) 458-1000

3330 Scott Blvd.
Santa Clara, CA 95051
(408) 727-1130

Technical information on some of Commodore's products is also available. They offer packages which include schematics, technical information, troubleshooting instructions, etc. For price and availability information, contact the parts department at the West Chester, PA, address listed above.

Cassette Filenames And Built-In BASIC

I own an Atari 400 computer. Can you specify a filename for the 410 program recorder? Also, how do you run a program that asks you to remove the BASIC cartridge on the 600XL? Since BASIC is built-in, it seems like it might be impossible.

Doug Stevens

The tape operating system could have been written to allow filenames, but since tape access is sequen-

tial and so slow compared to a disk drive, filenames are not especially useful. You can instead use the tape counter to find and position programs. Be sure to leave plenty of space between programs, since the tape counter is not precise, and you could inadvertently write over a program.

Some programs require 48K, 8K of which is not available with a cartridge plugged in. These programs check for a cartridge and ask you to remove it before the program will run. The 600XL has only 16K, though, so this shouldn't be a problem. In any case, you can effectively remove (disable) the built-in BASIC by holding down the OPTION key while you turn on the computer.

6502/6510 Differences

I am a relatively new computer owner and I want to learn machine language programming. As far as I can tell, my Commodore 64 contains a 6510 chip rather than the more common 6502.

A phone call to Commodore's corporate offices revealed that the machine language for the two chips is not compatible. The *Programmer's Reference Guide* directs all its instructions to the 6502 processor.

I am unable to find anything in print on machine language programming using the 6510. I would appreciate it if you could help me out. Are the two chips incompatible? If so, can you tell me how and where I might learn machine language for the 6510?

Dean Lind

Unfortunately, the representative you talked to at Commodore was misinformed. The 6502 and the 6510 microprocessors are compatible. Both use the same instruction set (LDA, STA, etc.) and addressing format (low byte, high byte). Books on programming the 6502 also apply to the 6510.

The only major differences between the two processors are bytes one and two of the 6510. The 6510 allows you to bank switch memory.

The Commodore 64 has 20K of ROM, including the BASIC interpreter, Kernal, and Input/Output control programs. There is also 20K of usable RAM memory "underneath" this ROM memory. You can switch out the ROM and switch in the RAM memory by bank selecting blocks of memory. If you wished, you could turn your 64 into a computer with 64K of usable RAM memory by switching out all of the ROMs. However, you would have to supply your own BASIC interpreter, operating system, and I/O control programs. Without these, the computer would simply freeze, and you wouldn't be able to write or run BASIC or machine language programs. For more information on bank selection, refer to your Programmer's Reference Guide.

The VIC-20, which is equipped with the 6502 microprocessor, does not support bank selection of ROM/RAM memory.

A book on 6502 machine language programming will apply to the 6510. You might also want to take a look at the "Machine Language for Beginners" column in COMPUTE!'s GAZETTE, *our sister publication.*

Printer Device Number Changes

Is it possible to change the device numbers of a printer? I want to buy a second printer for my computer, but am unsure if I can use both printers at the same time.

Jacques Poulet

Usually the device number cannot be altered. It depends on the brand of printer, but most printers have no significant internal "intelligence" (no computer assistance inside). You could change the device number of a disk drive because it can be programmed from the computer; it is an intelligent device.

However, some models of printers do allow you to change device numbers via a switch on the printer itself. The new Commodore MPS 801, for example, has a three-position switch that allows you to choose between device numbers 4, 5, and 6. And some few printers can be software assigned.

DIM On Commodore

What happens to the data when it enters a DIM statement (array) from an INPUT statement or a sequential file?

I teach computer programming part-time at Tulsa Junior College. This is my first semester with micros. I have a Commodore 64 and a VIC 1541 disk drive. In advanced BASIC, sequential files are common, and are usually used for search and sort routines. When the data is read from DATA statements into the arrays, there is no problem. The sort or search never stops, but when that same data is brought into the arrays from a sequential file, the sort or processing stops many times, making a 16 to 30 minute program run for hours.

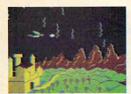I have written my own sequential file program, and later discovered and used the one off the demonstration disk that came with the VIC 1541 disk drive. Both have these stops. I have tried about everything. If you could give me a clue, I would appreciate it.

Darrel Henry

The pauses you see in the program are the result of a process called "garbage collection." It's caused by moving strings around.

Here's what happens: As new strings are created, the old ones are not thrown away; they lie dead in memory. Eventually, memory fills up and the computer has to stop and collect the strings that are still live. This takes time; the pauses are quite noticeable and can be time-consuming.

Strings that are completely defined within a program—from DATA statements or from an assignment statement such as X$="HELLO"—are used straight out of the program where they lie. These strings don't need to be collected; as you have noted, there's no garbage collection delay when you use these.

For your type of program—sorting and searching—there are two rules that will be very helpful in eliminating delays:

1. Don't move strings. Instead of sorting by moving them around from one part of the array to another—which creates garbage—use an "index" to keep track of where a string belongs within a certain sequence. (More on this in a moment.)

2. When you have finished with a string, set it to a null string, for example, A$(21)="". When you have disposed of almost all strings this way, and are ready to read in another set of strings from disk or tape, force a collection by using the FRE function, for example, code X=FRE(0). Garbage collection will run quickly if you have very few strings left. When you read in the next group of strings, they will come into the newly liberated memory space.

To illustrate point 1: Here's a program to sort an array of strings. It's a bubble sort, which is not very efficient. The point is this: After the strings are created, they are never moved. Only the index (A%) values move, and they are numbers, not strings, so there won't be any garbage.

```
90 REM BUBBLE SORT - INDEX DEMO
100 N=30:DIM A$(100)
200 REM CREATE RANDOM STRINGS
210 FOR J=1 TO N
220 A$(J)=CHR$(RND(1)*26+65)+CHR$
    (RND(1)*26+65)
230 NEXT J
300 REM: CREATE INDEX
310 DIM A%(N)
320 FOR J=1 TO N
330 A%(J)=J
340 NEXT J
400 REM: SORT INDEX
410 FOR J=N−1 TO 1 STEP −1
420 FOR K=1 TO J
430 REM: GET INDEX FOR K, K+1
440 X=A%(K) :Y=A%(K+1)
450 REM: FLIP IF OUT OF ORDER
460 IF A$(X)>A$(Y) THEN A%(K+1)=X:A%(K)=Y
470 NEXT K,J
500 REM: PRINT RESULTS
510 FOR J=1 TO N
520 PRINT A$(A%(J))
530 NEXT J
```

Study this program to see how the strings are sorted, but not moved.

There are other rules on how to handle garbage collection; the ones above will do the job for your application.

TI CALL KEY

I recently acquired a TI-99/4A and wondered if you would explain the use of the CALL KEY command?

David Stinchcomb

The CALL KEY statement has caused confusion for many TI users. The KEY subprogram, designed to return a single keystroke value, requires three parameters: a key unit, a return variable, and a status variable. The statement takes the format:

CALL KEY (n,K,ST)

where n is the key unit, K is the return variable, and ST is the status variable.

The key unit used in the CALL KEY statement determines the keyboard configuration assumed by the computer. Six key unit values (0–5), or keyboard configurations, are available on the TI-99/4A. The three key units generally used are 0, 1, and 2. A key unit of 0 refers to the console keyboard. Key units 1 and 2 map the console keyboard as split keyboards (a value of 1 to read the left side of the keyboard, a value of 2 to read the right), or read the fire buttons on joystick 1 and 2, respectively.

When a CALL KEY statement is executed with a key pressed, some value will be assigned to K (in our example above). The value given to K will depend on the key pressed and the key unit used in the CALL KEY statement. If you use a key unit of zero, K will correspond to the ASCII value of the key being pressed. For other keyboard configurations, the value of K will vary as noted in the TI User's Reference Guide (pp. II-87 to II-89). Eighteen in K signifies that the fire button was pressed.

The final parameter used in the CALL KEY statement is the status variable (ST). A nonzero value returned for ST indicates that a key was being pressed when the CALL KEY statement was executed.

CALL KEY can be used to get a desired response from the program user. If you want to test for any keystroke (with key unit 0), you would use the following two lines:

```
10 CALL KEY(0,K,ST)
20 IF ST=0 THEN 10
```

The program repeatedly loops back to line 10 until some key is pressed.

If you want the program to accept only a specific response from the user, such as Y for "yes," you could add these lines:

```
5 PRINT "TYPE THE Y KEY"
30 IF K<>89 THEN 10
```

Until the Y key is pressed, the program will loop back to 10.

THANKS TO COMPUSERVE'S CB SIMULATOR, "DIGITAL FOX" ACCESSED "DATA HARI" AND PROCEEDED TO AN "ALTARED" STATE.

The CB Simulator, where CompuServe Subscribers can Access Friends and Influence People on 72 Different Channels.

Just pick your handle and get on line. From math to matrimony, there's always someone out there who speaks your language. Friends from all over the U.S. and Canada are at it 24 hours a day. Talking tech or just having fun. And if you've got a secret, just use the CB Scrambler.

That'll fool the "lurkers," those CB "see it alls" who get their kicks by watching. Or you can always use the private talk mode for guaranteed one-to-one conversation.

The CB Simulator is just one of CompuServe's many electronic communications options that include a National Bulletin Board, Professional Forums and Electronic Mail. Plus, there's a world of on-line information and entertainment all for the price of a local phone call plus connect time.

You can access CompuServe with almost any computer and modem, terminal or communicating word processor.

To receive your illustrated guide to the CompuServe Information Service and learn how to subscribe, call or contact:

CompuServe

Consumer Information Service, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, OH 43220

800-848-8199

In Ohio call 614-457-0802

An H&R Block Company

Commodore Machine Language Clear

I have been trying to develop a monitor program for the VIC-20 and am having a difficult time creating a routine to clear the screen. Is there any way to clear the screen without getting the READY prompt?

Stan Payne

There are several ways to clear the screen from machine language. The easiest way is to assemble the following:

```
LDA  #$93  ;the equivalent of CHR$(147)
JSR  $FFD2 ;print it
```

This routine will work on both the VIC and 64. Incidentally, you can also use it to home the cursor. Just replace the $93 with $13. The hex number 93 is the same as the character that clears the screen in BASIC and $13 is the same as CHR$(19) which homes the cursor.

Another way to accomplish the same result is to assemble the following:

```
JSR  $E55F ;clear the screen and home the cursor
```

On the 64 the routine is located at a different location, thus the coding is different:

```
JSR  $E544 ; clear and home
```

If you just wanted to home the cursor on either machine you could JSR to $E581 on the VIC or $E566 on the 64. Although these are two quick, easy ways to clear the screen, there is yet another. You could write your own routine to clear the screen. The routine would need to store a $20 at every screen location. $20 is the number for a blank character.

Commodore 1541 Drive Grounding Error

In your May issue you printed a letter from Jay Elmore regarding read errors on the 1541 disk drive. After buying a drive for my Commodore 64, I discovered that it was giving me error 23 and 27. Both are read errors. I returned the drive to the dealer and he tested it on a 64 and discovered no problems with it. I took the drive back home and carefully read the manual that comes with it and discovered that errors 23 and 27 can be caused by grounding problems. I checked the electrical outlet where the drive was plugged in and discovered that the ground plug was not properly connected. After fixing this, I haven't had any problems with my drive.

Ron Restivo

www.commodore.ca

Computers And Society

David D. Thornburg, Associate Editor

Discovery-Based Learning And Teenagers

On the Saturday before the Summer Consumer Electronics Show in Chicago, I was speaking at a Logo workshop in northern California. Much of the emphasis in one of my talks was on the importance of discovery-based learning for all ages, not just for the child in elementary school. While I have been emphasizing this topic in my talks for several months, most people seem content to let Logo be a tool for the younger computer user.

Educational Software

As I drove to the airport to catch my plane to Chicago, I wondered if this reluctance to bring the power of educational environments like Logo to older students was also evident in other educational software. After all, the mention of educational software usually conjures up visions of activities for the younger child. Whether these activities are drills associated with a specific subject or are open-ended activities designed to teach problem-solving skills, the fact remains that only 2 percent of the current educational software appears to be directed to teenagers. Some people might argue that, once a child has entered the teenage years, educational software isn't needed. If a teenager wants to use a computer, why not just let him or her write programs.

In fact, there are several reasons why teenagers should have access to good educational software:

1. Teenagers are in the process of forming career decisions. Controlled exposure to computer environments can demonstrate the richness of this field in a way that transcends purely recreational applications of computers.

2. For those students who are already interested in computers, computer-based instruction in problem-solving methods and the development of programming style can help these students use computers more effectively in their jobs.

3. Students of all ages benefit from becoming better problem solvers. So much of our focus has been on problem-solving software for the young (including such excellent programs as *The Factory* by Sunburst) that we can easily lose sight of the fact that the acquisition of problem-solving skills is important for learners of all ages.

To see what *can* happen to educational software for teenagers, we should first look at the other two areas where these children use computers—at home and at the arcade.

Interactive TV

As I look at popular computer activities in homes and arcades, there seems to be a major distinction emerging between the two. Arcade software has continued its focus on coordination and skill games. As the technology has advanced, these games have become more sophisticated. For example, several popular games use computer-controlled video disks. Except for advances in technology, however, these games seem to be stuck in a niche that one might characterize as interactive television.

Popular home software has taken a different approach. The home user is not able to run out and buy new technology every three months or so, and the challenge has thus been to make the existing technology become ever more useful

David Thornburg is an author and speaker who has been heavily involved with the personal computer field since 1978. His main interest is in making computers responsive to people's needs. He is the inventor of the KoalaPad graphics tablet and is the author of nine books about programming including Computer Art and Animation: A User's Guide to Atari Logo, The KoalaPad Book, *and* Exploring Logo Without a Computer *(Addison-Wesley). His* 101 Ways to Use a Macintosh *will appear soon from Random House. He has been called "an enthusiastic advocate for a humanistic computer revolution," and his editorial opinions have appeared in* COMPUTE! *since its inception.*

www.commodore.ca

THE FIRST FAMILY

Introducing *the* home productivity software Commodore,™ IBM™ and Apple™ owners have been waiting for. Three easy to use, powerful programs from Creative Software.

Now you can run low cost, full-featured software that performs the most sought after functions.

Creative Writer.™ Creative Filer.™ And Creative Calc.™ Each developed by Creative Software and designed to give you the most for your software dollar.

HASSLE-FREE WORD PROCESSING.

Finally there's Creative Writer. A simple word processing program anyone can start using in just fifteen minutes. With Creative Writer, editing functions similar to the ones found in more expensive programs are at your fingertips.

And best of all, you won't have to relearn your computer's keyboard because all cursor control keys work similar to your computer's standard editing mode. You'll even be able to use the function keys to perform more advanced processing tasks—move blocks of type, search forward, even search and replace— all with just a few key strokes.

Plus "help" screens are always ready to prompt a desired function. And one more thing—every Creative Writer box comes with a free Verbatim minidisk. Creative Writer. The easiest way to replace your typewriter.
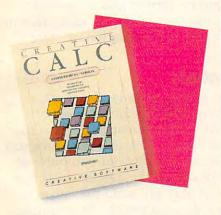
C R E A T I V E

with each new wave of products. While it is true that popular home titles continue to stress entertainment value, the most popular computer games appear to be those that let the player create his or her own levels or game fields. Anyone who doubts this need only look at the overwhelming popularity of Brøderbund's *Lode Runner*. Every *Lode Runner* enthusiast I know spends most of the game time creating new levels to play.

The Fourth Generation

The popularity of construction set software is not new, as readers of this column know. Programs like *Pinball Construction Set* (Electronic Arts), *Rocky's Boots* (The Learning Company [TLC]), and *Dancing Bear* (Koala Technologies) have been popular largely because they allow the user to make unique creations within the context of a predefined activity. The task of creating new games in *Lode Runner* or *Pinball Construction Set* is, quite simply, the task of creating a computer program. Construction sets are examples of fourth generation programming languages, and the fact that these programs are so popular in the home market is quite heartening.

These programs are helping their users develop problem-solving strategies and other higher-order thinking skills. Since they can (and are) developing these skills at home, they should probably also be able to develop them at school as well. Unfortunately, some teachers can't see past the game aspects of these programs, and have thus banned them from the classroom. This makes as much sense as banning humorous literature from the English classroom. Teachers should look for the deeper significance in the newer computer games.

With this background in mind, I walked through the massive computer exhibits at the CES looking for some sign of educational software that would appeal to the entertainment and cognitive skill levels of the teenage user. Not surprisingly, I found the answer at the home of *Rocky's Boots*—TLC. Long known for their support of the young computer user, TLC decided to create a product for the older child. The result of their effort is *Robot Odyssey I*, an entertaining adventure game that is also a sophisticated programming language. As the first product in TLC's "DigiWorld" series, it represents a significant contribution to open-ended educational software geared to the teenage (and older) market.

Players begin by falling into Robotropolis, a futuristic underground city inhabited by robots. The object is to escape this world and return to civilization. This is accomplished by constructing robots that help the players work their way upward through several layers of this underground city. Each layer has new obstacles and antagonists that require robots with different skills to help in the escape. As players get closer to the top, the challenge becomes more difficult.

Complex Circuitry
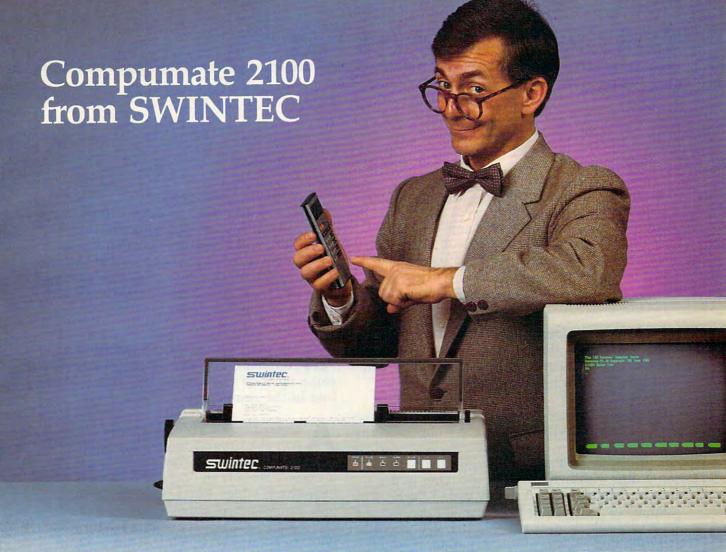
What distinguishes *Robot Odyssey I* from other multilevel adventure games is that the player must construct robots that are programmed to display certain behaviors needed to avoid or neutralize obstacles. The task of creating these robots involves learning how to "wire" the robots to perform certain tasks. A special environment called the Innovation Lab lets the player work on robot design. There are three robots at the player's disposal. Each robot resembles a spaceship and has four thrusters to move it in four directions. The robot also has a claw that can be used to pick up things, and bumpers to tell when the robot has bounced against a wall or other obstacle. By interconnecting the bumpers with the appropriate thrusters, the player can create a robot that solves simple mazes.

More elaborate programs can be created with the aid of integrated circuit chips built from standard logic elements (AND, OR, XOR, etc.). Once an array of these elements has been interconnected and brought to the desired pins on the chip, the resultant circuit is burned into a final chip that can be carried inside a robot to be connected with the rest of the circuitry. Amazingly, a finished chip can be carried inside another chip, and this recursive nesting can take place up to 40 times. This allows the creation of quite complex circuits.

Intelligent Robots

Each of the three robots becomes, in effect, a fully programmed entity. All three robots can carry out their tasks simultaneously. This is like having a word processor, a spreadsheet program, and a videogame running on your computer at the same time. But one of the neatest aspects of these robots is that they can send messages to one another using their antennae. For example, each robot could be programmed to look for fuel crystals. As soon as one robot finds a crystal, it can send a message to the other robots to stop looking for fuel and to find the first robot instead. This type of programming in which computational objects send messages to each other is reminiscent of the sorts of things one expects from Smalltalk, LISP, or Logo—not the sort of thing one expects from a videogame.

To properly explore *Robot Odyssey I* would take far more space than I have. Suffice it to say that if teenagers have only 2 percent of the educational software, this program shows that they won't need much more. ©

Questions Beginners Ask

Tom R. Halfhill, Staff Editor

Are you thinking about buying a computer for the first time, but you don't know much about computers? Or maybe you just purchased a computer and are still a bit baffled. Each month in this column, COMPUTE! will answer some questions often asked by beginners.

Q How long does a floppy disk last? If I have some disks that are more than a year old, should I copy them onto new disks? How can you tell if a disk is wearing out?

A The lifetime of a floppy disk largely depends on how heavily it is used and how carefully it is handled.

It's doubtful that you would actually wear out a disk with normal home use. One well-known manufacturer of floppy disks (3M-Scotch) guarantees its products for 3.5 million passes per track. 3M calculates that if you updated the disk (rewrote every track) once every hour, it would take more than 200 years to exceed 3.5 million passes. Not many home computer owners would use a disk that heavily. Some disk-intensive business programs which run 40 hours a week might approach that many passes in much less time than 200 years, in which case an alternative should be considered—perhaps a hard disk or a RAM disk. But the casual or moderate user has little to worry about. At COMPUTE! we have some disks that get fairly heavy use for storing word processing files, and after two years they're still going strong.

A floppy disk's lifetime can be drastically shortened, however, if it isn't handled and stored correctly. In careless hands a disk might not even last ten minutes. Disks are particularly vulnerable to magnetic fields from monitors and TV sets, spilled drinks, cigarette ashes, extremes of heat and cold, heavy weights, ballpoint pens used to mark on labels, and even chemical fumes.

Unfortunately, the first clue which indicates a disk is going bad probably will be a lost program or a lost file. One day you'll confidently slip the disk into the drive and discover that a certain file won't load. That's why you should always keep at least one backup copy of important files on another disk or tape.

There is one clue you can watch for, though—if you regularly clean your disk drive's read/write head and notice one day that it's much dirtier than usual, it could mean that the magnetic coating on one of your disks is starting to flake off. It could also mean that you bought some poor quality disks.

Q On my Commodore 64, when using the command PRINT 7↑2, the computer will come up with the answer 49.0000001. Is this a bug in the computer?

A Not really. It's a rounding error caused by the way the Commodore 64 calculates exponentiation. When you ask the computer to figure 7^2, it doesn't actually multiply 7*7. Instead, it uses logarithmic tables. Rather than engaging in a long discussion of higher mathematics, let's just say that these tables sometimes lead the computer to an answer which is slightly off. If 49.0000001 isn't accurate enough for your purposes, you can either PRINT 7*7 or PRINT INT(7↑2), which extracts the integer from the result.

The Commodore 64 isn't the only microcomputer which suffers from this problem by any means. Certain math operations will result in very small rounding errors on anything from an Atari to an IBM PC. On an Atari with the BASIC cartridge, for example, PRINT 2^2 (equivalent to PRINT 2↑2 on a Commodore) comes out to 3.99999996, while PRINT 2*2 yields the correct 4. The problem was fixed with the BASIC built into the Atari 600XL and 800XL.

The quirk which leads to this kind of problem is that computers perform all of their math in binary. When floating-point (moving decimal point) numbers are converted to binary and then back to decimal, small errors can accumulate.

Q I recently found several old 60-minute cassette tapes. Are the 60- and 30-minute cassette tapes acceptable for program recording? Is there any command or program which can be used to erase these tapes? They contain music and talk.

A Theoretically, longer tapes are thinner and thus more susceptible to stretching and print-through (signals leaking through the tape and imprinting on adjacent layers of tape). But we've never had any problems saving programs on 30- or 60-minute cassettes. If they are good-quality tapes, they should work as well as the 10- or 20-minute cassettes generally sold for data recording. However, the longer tapes could be slightly less convenient if you store many programs on the same cassette. You'd have to wind the tape further to locate the program you want to load. But even this is a negligible problem if you jot down the tape counter numbers.

It isn't really necessary to erase the old material on the tape before recording your programs. The recorder's erase head will wipe it out as you record. However, your letter indicates you have an Atari 400 computer. The Atari Program Recorder has a *stereo* read/write head because the Atari can accommodate an audio track that plays through the TV speaker. When you load your program, you'll hear the old audio material along with the bleeps of the data. By typing POKE 65,0 before loading the tape, the bleeps will be silenced and you'll hear the audio only.

However, if you do want to erase a tape before using it, the best method is to use a bulk tape eraser (available at electronic supply stores).

Another way is to insert the tape into an audio cassette recorder, press the record button, and disable the microphone by switching it off or sticking a null plug into the microphone jack. If the recorder has level controls, turn them all the way down.

With an Atari Program Recorder, you can erase tapes by pressing the record and play buttons and typing POKE 54018,52. This switches on the cassette motor from the computer. You can turn off the motor by typing POKE 54018,60.

©

THE BEGINNER'S PAGE

Robert Alonso, Assistant Editor

ROM And RAM

Every group has its lingo. When you first start to get used to computers, there are a few words to learn, a bit of *computerese*. Two of the simplest, yet most misunderstood, of these terms are ROM (Read Only Memory) and RAM (Random Access Memory).

Absent-Minded Computer

Read only memory is called that way because it is a kind of memory in each computer which can only be read. No matter how many times you try, you may never write in ROM. Computers come equipped with ROM because they need to have instructions that tell them how to perform certain functions such as adding or subtracting or even how to send things to disk drives or printers. The ROM memory stores these vital instructions in chips that can't be erased by writing to them or by turning off the computer. Without ROM your computer would be useful only as a door-stop. Just imagine turning your computer off and having it forget how to work the next time you turned it on.

BASIC is usually a set of machine language routines stored in several ROM chips in each computer. Another set of important ROM routines are the input and output routines. These routines instruct your computer on how to communicate with external devices such as disk drives, cassette recorders, and printers. ROMs obviously contain important routines that are needed for doing any kind of work with your computer.

Extra ROM

Another useful and popular form of ROM is the game cartridge. Yes, those small black boxes that provide you with so many hours of fun are just preprogrammed read only memories. These ROMs in cartridge form are perhaps the fastest way to load a program into your computer. The reason they are so fast is that usually there is no
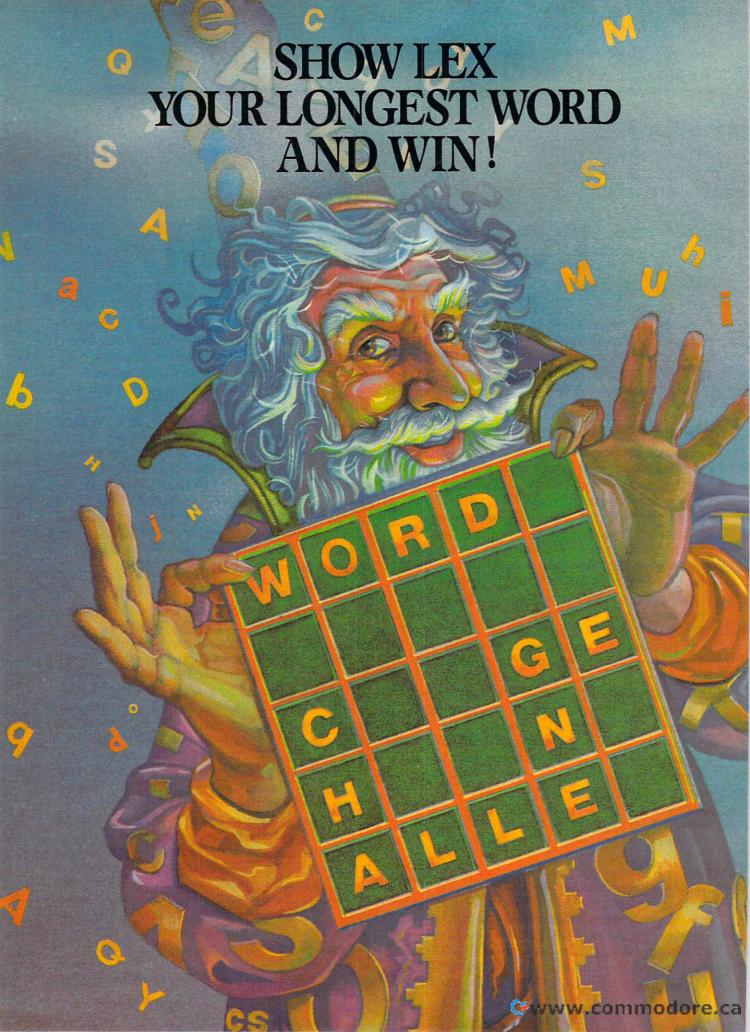
loading taking place. You plug in the cartridge, and it immediately becomes a part of the computer. The computer has immediate access to the programs and data stored on the cartridge.

RAM memory, although not as intrinsically important to the computer as ROM, serves important functions in the computer. It is the area of memory where user programs are typed in and run. RAM is also used by all the commercial software that comes on disk or cassette. The routines that are located in ROM often have to rely on pointers in RAM to function correctly. Without RAM memory, the computer might not have a screen display, and it would have no such thing as a keyboard buffer to hold extra keypresses. Graphics would not be available either.

Empty Boxes

All memory in the computer can be described as a series of boxes that can hold something. Let's say that you have 1024 different boxes in which to put things. You could choose to place objects in these boxes sequentially from box 0 on up to box 1023, or you could just decide to place objects into boxes in any haphazard way that you yourself could understand. RAM memory is just like a series of empty boxes that allow you to place numbers into them. The only restriction that these boxes impose on you is that you must never place a number over 256 into any one box. If you have a section of memory that has 1024 spaces, you can claim to have 1K of RAM memory. The difference between RAM and ROM when compared to boxes is that ROM memory is like a series of boxes that already have objects in them. Even more important, the boxes can't be emptied or altered by the user. The user can only look into the box to see what's in it. ROM could thus be compared to glass boxes. You can see in but can't remove anything without breaking something.

Any programs that run in RAM memory can

SHOW LEX
YOUR LONGEST WORD
AND WIN!

ENTER THE EXCITING HAYDEN NATIONAL WORD CHALLENGE™ CONTEST.

IF YOU'VE GOT A WINNING WAY WITH WORDS, THIS IS YOUR GAME.

Word Challenge. The mind-boggling game in which you compete head to head against LEX, the mighty Wizard of Words.

Select the size of your playing square, pick from 26 skill levels, set a time limit, and choose from 5 scoring methods. Then see how you stack up against the Wizard and his 90,000 word lexicon. It's a race against the clock.

You and LEX compete to form words by connecting randomly generated letters. Words are legal if their letters touch in sequence on any side or corner. The more words you find, the better your chances of beating LEX.

When your time's up, LEX displays the words you found and the words he found. You score points for every word you made that he didn't. Word Challenge keeps the score.

Word Challenge is based on Proximity linguistic technology. It operates on Apple™ II+, IIe, IIc, IBM PC,™ PC jr.™ and popular compatibles. Available soon for Macintosh™ and Commodore 64.™

Play Word Challenge alone or with a group. See if your family or friends together are wordy enough to bring LEX to his knees.

TO WIN OUR CONTEST, GIVE US THE GOOD WORD.

It's easy to play. First buy Word Challenge. Then follow the rules as described on the official Entry Form available only inside specially marked Word Challenge packages. Fill in the letters on the official contest grid to construct the longest possible word. Show your word to LEX. If he has it in his vocabulary, send it in.

First prize for the longest word is $10,000 cash.

Second prize is $3,000 cash.

The top 200 finishers will receive Word Challenge polo shirts.

Be one of the first 5,000 to buy Word Challenge and enter the contest, and we'll send you a free Word Challenge T-shirt.
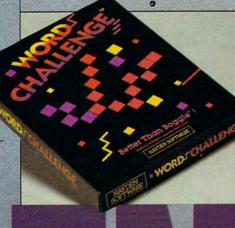
Enter now. The contest expires October 31, 1984. Purchase of Word Challenge is required. Contest void where prohibited.

JOIN OUR WORD ASSOCIATION. THE RIGHT WORD COULD BE WORTH $10,000.

To purchase your copy of Word Challenge, see your local software dealer, or contact Hayden Software, 600 Suffolk Street, Lowell, MA 01854, or phone toll-free 800-343-1218.

Word Challenge is a registered trademark of Hayden Software Company, Inc.
Apple and Macintosh are registered trademarks of Apple Computer Inc. IBM PC and PC jr., are registered trademarks of International Business Machines, Inc. Commodore 64 is a registered trademark of Commodore Business Machines.

HAYDEN. AMERICA'S FIRST FAMILY OF SOFTWARE.

HAYDEN

www.commodore.ca

be modified. For example, if you wrote a program in BASIC and then decided that you wanted to modify something in it, you could easily do so. One peculiarity of BASIC programs is that the RAM in which they are located must be continuous. If there were a gap of any sort, like a missing box, the program would not work properly.

Screen RAM

Perhaps one of the most interesting sections of RAM memory is the screen. It is one of the few RAM locations that are shared by both the central processing unit and the video chip in all computers. What this means is that there is a section of RAM—let's say a 1024 box section—that both the computer's main processing chip and the chip that produces the colorful TV image can look at. The reason it is shared is that you must be able to change the screen either by POKEing values into the screen RAM or by PRINTing values onto it. For you to be able to do this, the central processing unit must have access to it. The video chip needs to have access to the RAM because it needs to know what to display on the TV set. The video chip essentially scans screen RAM and evaluates what is stored there. Once it knows what is in the screen memory, it can reproduce an image of it on your TV.

One useful feature that many home computers come equipped with is that more RAM and ROM can usually be added by just plugging a cartridge into the side or back of the computer. In some computers you have to plug the RAM into the inside expansion areas, but this is usually an easy procedure. The reason expandability is such a desirable feature is that it enables the user to program or use larger programs with more functions and commands and that it facilitates the use of alternate computer languages or BASIC enhancements.

Memory Limitations

There is one limitation to the use of both RAM and ROM. Each microprocessor, the brain chip of each computer, can access only a predetermined amount of memory at any one time. For example, the popular 6502 processor found in the Apple, Atari, and Commodore computers can directly access only 64K of memory. The 8088 microprocessor found in the IBM PC and PCjr can access a little over one million bytes. There are other limitations that manufacturers either impose on their machines or are forced to adhere to. For example, Microsoft BASIC will only accept a program up to 64K long. An example of a self-imposed limitation is that IBM restricted the memory limits of the PCjr to 128K. ©

www.commodore.ca

The Educational Software Explosion

Kathy Yakal, Editorial Assistant

Even if you didn't buy your home computer as an educational tool, you've probably learned a lot from your interaction with it. Games, home productivity packages, and other computer programs can teach, but recently there's been a proliferation of dedicated educational software. Software developers and publishers are vying for top positions in educational software, potentially a billion-dollar business.

Experts have said that it was pretty easy to predict who the first 6 percent of computer owners would be, says Marilyn Rosenblum, vice president for product development at CBS Software. They're the same people who have expensive hi-fi systems and Sony Walkmans, people who like new technology.

"The important issue for us and for the future of this industry is who the next 6 percent will be," she says. "The thing that's going to determine that is how truly useful we can make computers."

Software publishing firms have been scrambling over the last few years to figure out how they can influence the next 6 percent to buy. The bets right now are on education. "There's been a tremendous proliferation of manufacturers into the educa-

tional software market," says Jordan Levy, vice president of marketing at Software Distribution Services.

Levy and many others in the industry who attended last summer's Consumer Electronics Show are overwhelmed by the number of publishers who have either entered the market or added educational software to their already existing line. Future Computing predicts that consumers and schools will purchase 6,787,000 units of educational software this year.

Bringing School Home

"I think you can trace the recent popularity of educational software to penetration of the home computer," says Richard Khaleel, president of Scholastic, Inc.'s Software Group. Game software was popular initially, he says, due to the popularity of the VCS. "And with the penetration of the personal computer into the business market, we saw the explosion of business software."

Further, home computer sales took off partly because children were using them in schools. "The use of computers in schools is probably one of the main reasons for the popularity of home computers," says Khaleel. Market analysts at the Minnesota Educational Computing Consortium (MECC) agree.
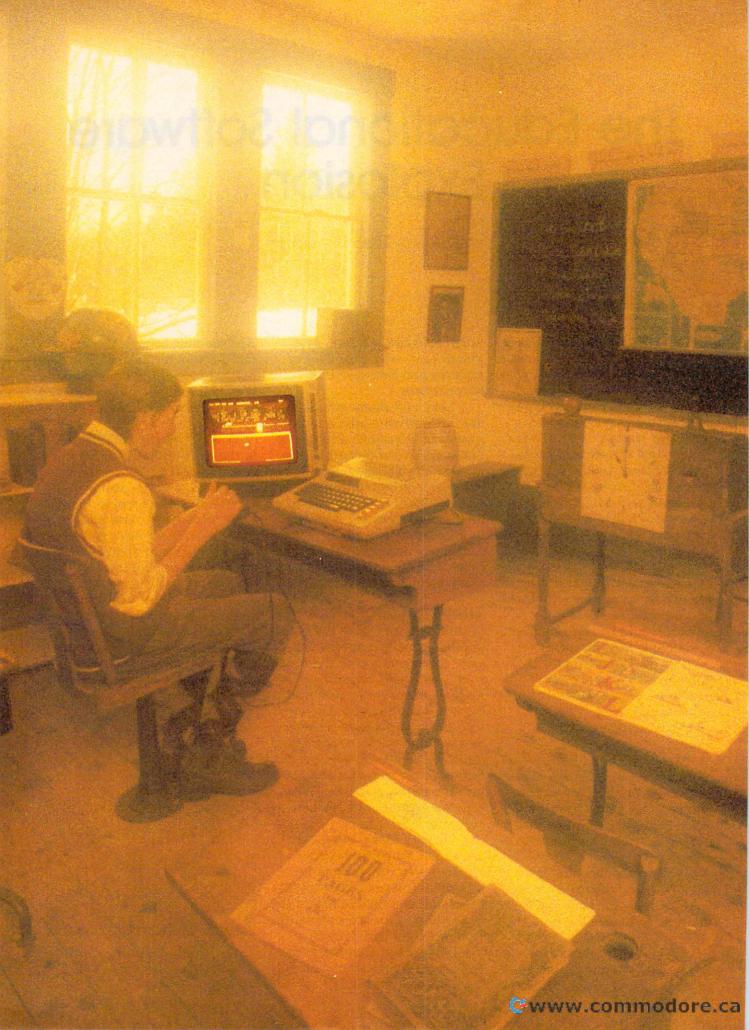
And as more school districts make computer literacy an important part of the daily curriculum, children are exposed in ever-increasing numbers.

Not Just A Shooting Star

In the rush to find uses for this newly installed base of home computers, a lot of potential applications and inappropriate game ideas have come and gone. With them have gone a lot of software companies. But the concept of education is no fad. "Everyone latched onto education because education doesn't change that much," says Marilyn Rosenblum. "The need to teach children to read and write well will always exist."

Marilyn Rosenblum, vice president for product development at CBS Software.

Today's most innovative educational software began here 60 years ago.

At Scholastic, we have something no other educational software company has: 60 years' experience making learning fun for kids.

We began in the schoolrooms of the 20's with the first national news magazine written especially for young people, The Scholastic. Since then, our one magazine has grown into 37, and we've become the largest educational publisher of books and magazines in the English-speaking world.

Now we've put everything we've learned from five generations of school children into the most innovative family of educational software available today. Scholastic Wizware.™

Our experience makes Wizware different from all other educational software. It turns learning subjects like geography, writing and spelling into exciting adventures for your children. And because every Wizware game is *interactive*, kids become deeply involved in what they're learning.

What's more, our experience has taught us the importance of teaching things most other educational software leaves out. Like teamwork, imagination, critical thinking and problem solving. You'll find them all in Wizware.

But by far the most important result of our experience is that your children will thoroughly enjoy learning with Wizware.

There are now Wizware programs for teaching everything from creative writing to computer programming. Here are a few of the ways we bring learning into the Computer Age.

Spelldiver.™
It's the most ingenious way ever devised for teaching spelling and improving reading skills. Deep beneath the sea lie giant words covered by a strange seaweed called lettermoss. Kids must face ferocious sharks and pesky flipper-nippers to remove the lettermoss and decipher the words.

Agent U.S.A.™
Agent U.S.A. turns geography into an exciting race to save the nation from the ruthless Fuzzbomb. Along with learning geography, children hone their problem-solving and reasoning abilities.

Story Tree.™
Story Tree can bring out the Mark Twain in every child. Budding authors create their own mystery and adventure stories. A remarkable feature lets them weave alternate choices into every turn of the plot, and challenges their imaginations to the maximum.

Bannercatch.™
Based on the classic game of Capture-the-Flag, this is the most sophisticated and fun strategy game for kids available today. Not only do players learn how to devise complex strategies, they also learn how to work together to solve difficult problems.

Scholastic Wizware. Our experience makes all the difference.

Look for Wizware at your local computer store. Or contact Scholastic Inc., 730 Broadway, New York, NY 10003, 212-505-3000.

Scholastic™ Software
The most trusted name in learning.

Spelldiver, Agent U.S.A. and Bannercatch designed and developed by Tom Snyder Productions, Inc. Story Tree designed and developed by George Brackett. • Agent U.S.A. and Bannercatch available in Atari, Commodore, Apple and IBM versions. • Spelldiver available in Atari, Commodore and Apple versions. • Story Tree available in Apple.

www.commodore.ca

Having worked in educational publishing for a number of years before coming to CBS, Rosenblum's perspective of this "explosion" is different. In educational book publishing, it wasn't so much an explosion as an ongoing event. Educational publishers began to add microcomputer software to their existing lines of books and audiovisual materials.

"What we're seeing here is that a lot of the stuff that's been used in the schools is perfectly appropriate for the home," she says. "A lot of the explosion is just recognizing a new market."

Pulling Away From The Pack

Jordan Levy thinks that many software companies sang a swan song at last summer's Consumer Electronics Show. "There will be a lot of people who won't be at the next CES," he says. "A lot of people are going to fall by the wayside, mostly entertainment."

The same thing is bound to happen with educational software over the next few years, say software publishers. But in 1989, consumers will buy about five times as much educational software as they're buying this year, says Future Computing. They predict that 35,072,000 units will be sold in that year.

Whose name is going to be on those packages? Those companies that have experience working with children, says Scholastic's Richard Khaleel. "People who have experience knowing how people learn have the best chance of creating software that is not really state of the art, but state of the *mind*," he says. "No new software publisher that does not have a consumer franchise in another market has been able to spend the kind of money yet to be able to establish a basis for consumer trust."

"It's very important for

people who know children, who have traditionally been selling educational materials to children, to get into the business," says Weekly Reader Software Manager Fritz Luecke. "You're going to find fewer unknowns entering the market," he says.

Though parents may feel comforted by familiar names, market analysts at MECC predict that the competitive situation in the educational software market will eliminate those without educational strength and technical skill, as well as adequate marketing ability. "The next 18 months will hold some very hard lessons for those publishers who think the educational buyer is an easy sell," says MECC's Ron Barnes.

No Fun

In all of the brouhaha over which educational software publishers will emerge as leaders, there's a basic problem that everyone's trying to solve: Who deems a program educational? How do you decide when the educational benefits balance the enjoyment?

The argument seems to focus on just how much emphasis there should be on the entertainment aspects of an educational program, if any. There aren't any real rules yet, though educators try to quantify and set standards, and software designers and publishers try to develop formats that they hope will please parents, teachers, and students.

Those formats vary tremendously. "My belief is that educational software should manifest itself in some kind of discernible change in behavior, like grades," says Thomas Garsh, president of American Educational Computer, Inc. "You don't get through high school by being a good games player or having superior cognitive skills. You get through high school by having good grades. And you do not get accepted

Thomas Garsh, president of American Educational Computer, Inc.

into college with superior kinds of thinking skills. You get into college by passing the SAT and having good high school grades.

"So what we've done is almost totally related our software to curriculum, by grade and subject," he says. "We've given up a few whistles and bangs, which may initially be a disadvantage because it doesn't look as flashy, but the subject is there. I'm not defending that. I'm just saying that is reality. That's our position. In this rush to the market, I don't see many companies taking that position."

Pupils Of *Pac-Man*?

Software developer Tom Snyder says it's too early to judge, that it's dangerous to impose restrictions or define too concretely right now. "Even *Pac-Man* would be educational if you could freeze the action," he says.

"If you freeze *Pac-Man* or any other game, a couple of things happen," he says. "You get to use your head instead of just knee-jerk reactions. You start verbalizing to yourself what the alternatives are and formalizing them a bit instead of having them remain in sort of an intuitive, physical reaction.

"Beyond that, it really opens the experience up to more

Software developer Tom Snyder.

than one person. If you slow it down, everyone gathers around and you get five people talking to each other about what a decent strategy might be, and introducing the whole social element into it. That kind of interaction multiplies the learning instantly."

Keeping The Scales Steady

Somewhere between those two opposing viewpoints lie hundreds of software styles, with varying degrees of entertainment and learning. MECC Deputy Executive Director Ron Barnes contends that the learning objective of any piece of software should come first, with the game aspects then designed to increase the child's motivation to learn. Often, he says, educational software is developed the other way around.

DLM Teaching Resources publishes software which is designed to supplement curriculum as first priority. But, they say, what makes their software sell to both schools and homes is that it does not ignore the element of fun. "I think that what software producers are looking for is a correct combination of ingredients to combine into their instructional courses, to keep the interest and focus," says Jim

Hafer, product evaluation supervisor for MicroD.

"There seems to be a bit of a gray line there," he says. "Some have a theory that seems to be working, and that is to make education fun. But there's only so much that can be learned from a simple game as opposed to an in-depth program."

Richard Khaleel believes that we will see more segmentation between software that makes learning fun and software that is just drill and practice. "It all depends on the age," he says. "In grade school, homework is fun. It's only when things outside of school get more important that schoolwork competes with other interests. There's always room for both."

Pinning Down The Beast

In the meantime, educators and parents puzzle over how to decide. Not just on what kind of software or computer, but something more basic: whether or not the microcomputer can be used effectively as a teaching tool.

And they can be fooled, says Tom Snyder. "There is an erroneous promise, an implication that you're going to be able to do quantifiable things with it, testing how the education is," he says. "Therefore, it's excited educators and educational consultants who say finally, we can start pinning down this educational beast in general." ©

www.commodore.ca

"Hey, Commodore and Atari,"

PQ
The **Party Quiz** Game

Move away from your keyboard, sit down and relax. As your home computer asks questions, race your mind for the correct answer.

Match wits with your friends or play alone. PQ's exclusive Quick Response Controllers let from 1 to 4 players or teams react instantly to this high energy, exciting computer challenge.

THE HOT NEW COMPUTER TRIVIA GAME
Includes over 2700 questions

General Edition 1 Question Disk for **COMMODORE 64**
History, The Arts, Science, Geography, Sports, and Entertainment

ALSO FOR THE **ATARI**

REQUIRES DISK DRIVE

Includes 4 Quick Response Controllers, PQ Disk, Question Disk, Interface Module, and Gamer Guide.

Suncom

"Let's Party!"

P.Q. The Party Quiz Game™ includes the following features:

- A TRIVIA PLAYER'S GAME — not a computer game that plays trivia.
- No keyboard input — all players responses are generated from the unique Quick Response™ controllers.

- General Edition I has over 2700 questions for the Commodore 64 and over 1400 questions for the Atari.
- Competition or team play.
- Random questions.
- Lightning bonus rounds.

$59.95
Special Introductory Offer:

Additional Modules from the P.Q. Question Library include:

Sport Edition I	Entertainment I	General Edition III
Education Edition I	General Edition II	

$19.95 each

To Order Call Toll Free
800-558-0003
For WI Orders 414-351-2007

MasterCard
VISA
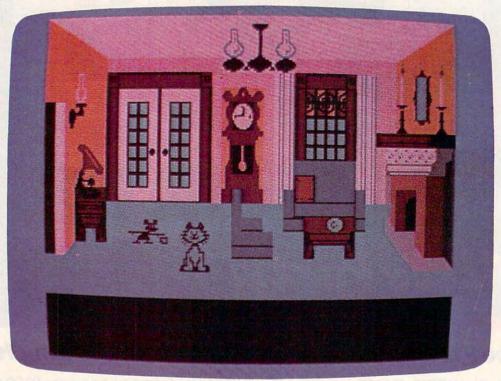
ComputAbility™
P.O. Box 17882   Milwaukee, WI 53217

Atari and Commodore are registered trademarks of Atari, Inc. and Commodore Business Machines respectively.

Dealer inquiries invited

www.commodore.ca

The Latest In Learning:

New Trends
In Educational Computing
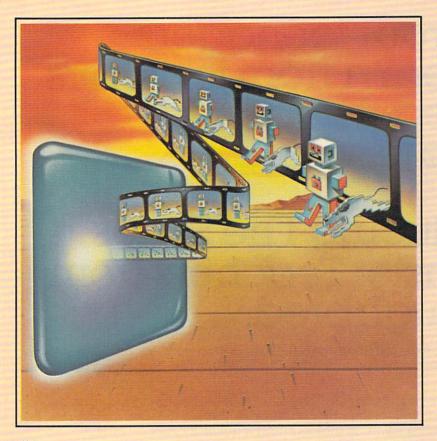
The House That Jill Built *lets youngsters design, build, and animate a variety of houses. Developed by Joyce Hakansson Associates for CBS Software.*

Selby Bateman, Features Editor

Construction sets, interactivity, networking—these are some of the trends in the quickly evolving educational software market. A swarm of new companies have entered the competition, and a variety of different educational approaches are being tried.

Few subjects will generate arguments faster in the microcomputer industry than what makes good educational software.

Interest in computer-based learning programs has never been greater, nor the debate over future directions more vocal. Parents and teachers are requesting more and better packages. Computer companies are courting software firms that specialize in educational products.

You imagine
    and a cast appears on the screen
You compose
    and music and sound are heard
You direct
    and an original story unfolds
You are a _MovieMaker._™

Creative Pastimes.™
Software as unique as you are.

Look for Creative Pastimes Software in your favorite book or computer store. You'll find it's suitable for all ages,
and compatible with most popular home computers. For more information call us at (800) 336-0338.
Or write Reston Computer Group, A Prentice-Hall Company, 11480 Sunset Hills Road, Reston, Virginia 22090.
MovieMaker is a registered trademark of Interactive Picture Systems, Inc.

| The Dolphin's Pearl | MovieMaker | Day | Middle of the Road Lizard |

C= www.commodore.ca

Now your teenagers can learn from teachers who teach teachers how to teach.

Introducing The Series by Master Teachers from Thoroughbred.

This unique series offers your teenagers the rare opportunity to learn—to reinforce and integrate concepts studied in school—through software developed by Master Teachers.

Teachers who have been recognized by other teachers as being truly outstanding for their knowledge. And for their ability to inspire people to learn—not by rote, but through *active participation*. So that learning is quick, thorough, and fun.

Master Teachers are so good at teaching, in fact, they teach *teachers* how to teach.

The first 11 titles are available now

Biology
□ Exploring that amazing food factory, the leaf
□ Photosynthesis: unlocking the power of the sun
□ The fascinating story of cell growth
□ How plants grow: the inside story

Math
□ Decimals made simple
□ Mastering units of measurement
□ Taking the mystery out of metrics

Social Science
□ The U.S. Constitution: our guarantee of liberty
□ The Constitutional amendments: what they mean to you

Chemistry
□ Molecules and atoms: exploring the essence of matter
□ The how's and why's of migrating molecules

And more to come—in these subjects, as well as English, physics...
All, with advanced color graphics.

Level of knowledge covered: from junior high to adult (except "Decimals," elementary to senior high).

For use with: IBM PC, PCjr.* Available soon for Apple IIe, IIc.**

Visit your dealer *today.* Or call us at 800-526-3968 (or 201-685-9000).
Or write to SMC Software Systems, 1011 Route 22, Bridgewater, NJ 08807.
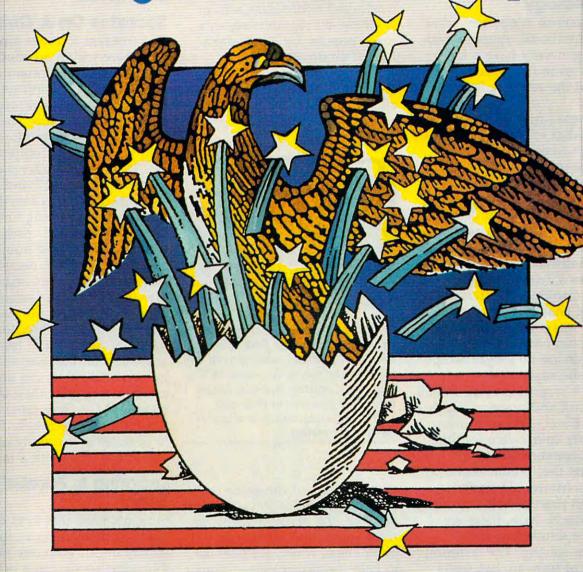
The Series by Master Teachers

The U.S. Constitution: our guarantee of liberty

Thoroughbred™

Social Science   Educational Software
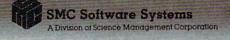
And greater numbers of educators are getting involved in the production and distribution of these programs.

Increased competition in this growing field is creating several important trends that will directly affect both the quality and the kind of future computer learning programs.

Challenging The Technology

Even staunch supporters admit that educational computer programs generally have a long way to go to reach their potential, but the software is evolving rapidly.

There's now "some clarity about what educational software is," says Marilyn Rosenblum, vice president of product development for CBS Software, a company that produces several lines of educational programs with such developers as Joyce Hakansson Associates and the CTW Software Group from Children's Television Workshop.

"I also think the technology is being challenged much more than it was a year ago. Things that would have been acceptable then are no longer acceptable," she says.

"We're getting away from a lot of drill and practice," adds Marge Kosel, director of microcomputer courseware for Sunburst Communications, a company that produces educational software. "Designwise, drill and practice is the easiest [to produce]. Now, there are a lot more simulations and problem-solving. The technical quality of software has really improved."
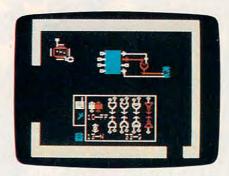
Design Your Own Robots

Greater emphasis is being placed on the amount of interaction that occurs between the student and the computer in several of the newer educational packages. And a higher level of sophistication has been built

into other programs, as some software developers reach for an older, teen-oriented educational market.

Take, for example, The Learning Company's new *Robot Odyssey I* (Apple II family, $49.95). Inspired by the company's earlier, *Rocky's Boots*, the new package is both a game and an educational tool. It's one of a growing number of products called *construction sets*, or *builders*—a programming style made popular by Bill Budge's *Pinball Construction Set*.

Programs written in this style are highly interactive and allow the player to control

In Robot Odyssey I, *players must build robots to escape from Robotropolis, a futuristic underground city inhabited by robots. Marketed by The Learning Company, Robot Odyssey I is the first program in their new DigiWorld series, which is aimed at the teenage market.*

many aspects of the game environment. In *Robot Odyssey I*, for instance, the player actually designs robots and builds computer chips, which are later used to help the player escape from a futuristic robot-run city, Robotropolis. The program features built-in tutorials to teach the basics of digital logic and an Innovation Lab in which players design and test chips and robots.

Atari's Futuremakers And Milestone

New programs from Atari, Inc., also reflect the trend toward

greater sophistication. The company's Milestone Series from Atari Learning Systems and its Futuremakers simulation programs are quite interactive and feature a hands-on approach to learning. The AtariLab science packages, part of the Milestone Series, let youngsters simulate more than 100 different experiments using the computer and a laboratory kit.

Socrates On A Chip

The freedom to move almost anywhere within an educational program is an important aspect of the Arrakis Advantage, a family of educational software marketed by Prentice-Hall for Apple, Atari, Commodore, and IBM PC and PCjr computers.

Directed at students in grades 7 through 12, the Arrakis Advantage series is based on the Socrates Learning Environment—a dialectical teaching technique borrowed from the Greek philosopher, notes Charles de Martigny, managing director of Arrakis Systems.

"Where other programs operate in only one direction, Socrates [the system] can move in any direction the user wishes," he says. "Students can interrupt, ask questions, test themselves, review and explore other subjects and, in effect, talk directly to Socrates as naturally as they would converse with a teacher."
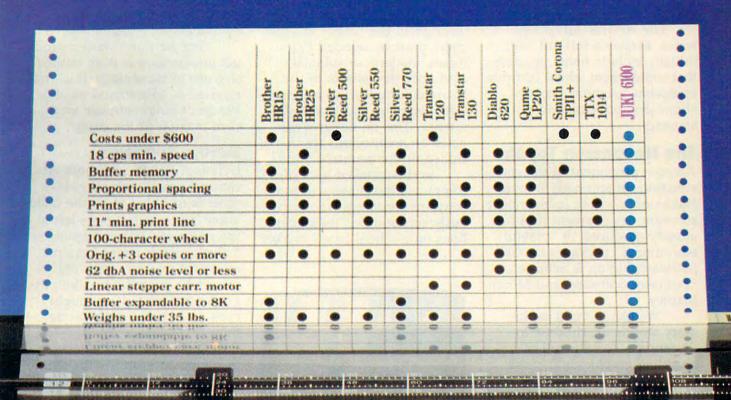
Learning Is Parallel, Not Serial

Students need that kind of flexibility since learning is not strictly linear, with one event following another in a predictable chain. Instead, information arrives in a *parallel* fashion, with information coming from a variety of complementary sources, points out Emiliano De Laurentiis, director of advanced research and development for Arrakis.
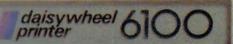
"People should start looking for software that's more intelligent; software that allows

JUKI 6100.
HARDWARE WITH NO SOFT SPOTS.

	Brother HR15	Brother HR25	Silver Reed 500	Silver Reed 550	Silver Reed 770	Transtar 120	Transtar 130	Diablo 620	Qume LP20	Smith Corona TPII+	TTX 1014	JUKI 6100
Costs under $600	●		●			●				●	●	●
18 cps min. speed		●			●		●	●	●			●
Buffer memory	●	●			●			●	●	●		●
Proportional spacing	●	●		●	●		●	●				●
Prints graphics	●	●	●	●	●						●	●
11" min. print line	●	●			●			●	●		●	●
100-character wheel												●
Orig. + 3 copies or more	●	●	●	●	●			●	●			●
62 dbA noise level or less								●	●			●
Linear stepper carr. motor							●	●				●
Buffer expandable to 8K	●				●							●
Weighs under 35 lbs.	●	●	●	●	●	●	●	●		●	●	●

Note: Above chart is an art treatment and was not printed by the JUKI 6100.

JUKI daisywheel printer 6100

THE JUKI 6100: Feature for feature, the best all-round printer. We challenge our competition to beat our overall record. Dollar for dollar, JUKI gives you more of what you want in a daisywheel printer. And no wonder: we've been building quality precision machinery for over 45 years. So for a printer that works . . . and works . . . and works, choose JUKI. Ask to see the JUKI 6100 at your local JUKI dealer.

JUKI®
The worker.
JUKI INDUSTRIES OF AMERICA, INC.

www.commodore.ca

you to interact on a normal level, to type in a sentence, to ask questions," he says. "The technology exists now. There's no reason why it shouldn't be implemented."

The Arrakis Advantage series, available this fall, will initially include home programs that complement school curricula in algebra, geometry, biology, chemistry, physics, and statistics.

The Hakansson Touch

Another proponent of the construction set approach is Joyce Hakansson, whose independent software development firm has already produced 19 "playful educational" packages for CBS Software, Simon & Schuster, Spinnaker Software, and Milton Bradley.

"We like to build exploratory worlds for children to investigate—worlds full of sparkling graphics and music and a good measure of giggles," she says. "I want every aspect of a program, even the most subtle, to serve an educational purpose. If a child builds words, then building words should be a powerful tool to make something happen so that the reward is inherent in the activity. We feel a program should validate and support a child's natural intuition and enthusiasm for discovery."

Among the newer programs developed by Joyce Hakansson Associates is *The House That Jill Built* (CBS Software), available now on disk for the Commodore 64 and this fall for the Apple II family.

Build It Your Way

Created for children five years of age and older, the program lets the player design and build a house from the ground up. Youngsters furnish, decorate, and animate the environments simply and easily, but the program also can be used by older children and even adults in more complex ways.

Working from electronic blueprints, you choose one of a variety of four-story houses to build. Doors, windows, and stairways must be placed throughout the house. Exterior styles must be selected (Victorian, modern, or colonial). And once the outside of the house is complete, you then design, decorate, and furnish the interior. Tips on design and notes on architecture are included in the manual.

As the installed base of computers has increased, some software producers who previously concentrated their attentions on a school-based market

Youngsters count as many crops as they can before the sun sets in DLM's Number Farm.

are coming into the home with programs to supplement their curricular packages.

It's "Arcademic"

DLM Teaching Resources of Allen, Texas, currently offers hundreds of educational materials for schools. DLM has made the transition into the home software market as well with its Arcademic Skill Builders series in math, language arts, and other areas.

Educational software companies with strength both in the school and in the home will have the best chance of survival, says DLM's Karen McGraw, project editor for microcomputer software and a former teacher. DLM emphasizes content in its packages, she adds, but with a presentation that is also fun for the user and with programs that fully use the capabilities of the computer.

"But we don't want a product just because it is an innovative use of technology, if it's not married to educational validity. We don't build software and then make it fit the child."

Sunup, Sundown

In one of DLM's preschool educational games, for example, a timer is included to let the child know how much time is left in the program. Rather than using a clock, which might have no relevance to a four-year-old who can't yet tell time, the program employs a day-to-night cycle to accomplish the same thing.
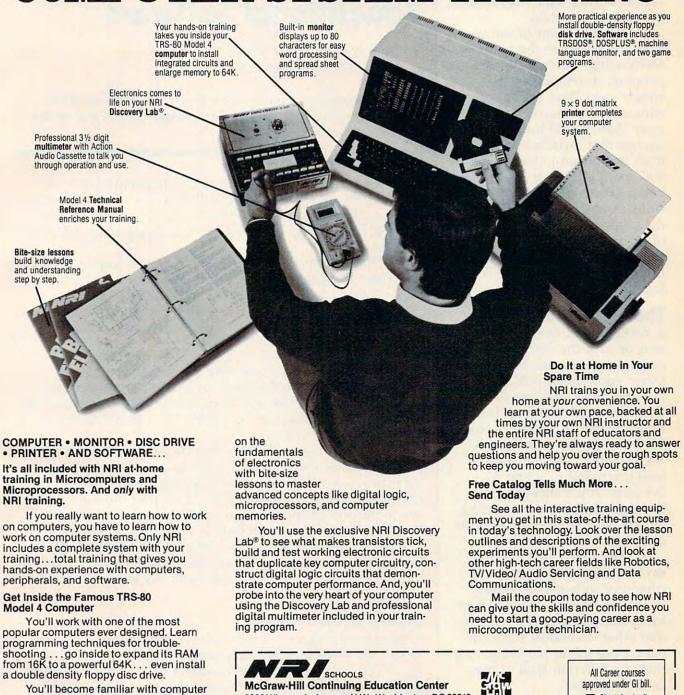
At the start of the game, the sun comes up. As the game progresses, the sun crosses the sky and then starts to come down as the game draws to a close. When the sun sets and night comes onto the screen, the game is over.

"We use real-world things that are important to a child, that a child can understand," says McGraw. "Unless you really know about education, unless you really understand the mind of a child, you would never think of it." Drawing on its experiences as a curriculum-based, educational-software producer, American Educational Computer (AEC) is also among those school-oriented software companies which have entered the home market.

"Tested And Tried"

An educational program in the home can be much more effective if it's first been shown to be successful in the schools, argues AEC president, Thomas B. Garsh. "It's been tested and tried and is based on sound educational pedagogy," he says.

ONLY NRI GIVES YOU TOTAL COMPUTER SYSTEM TRAINING

Your hands-on training takes you inside your TRS-80 Model 4 **computer** to install integrated circuits and enlarge memory to 64K.

Built-in **monitor** displays up to 80 characters for easy word processing and spread sheet programs.

More practical experience as you install double-density floppy **disk drive**. **Software** includes TRSDOS®, DOSPLUS®, machine language monitor, and two game programs.

Electronics comes to life on your NRI Discovery Lab®.

Professional 3½ digit **multimeter** with Action Audio Cassette to talk you through operation and use.

9×9 dot matrix **printer** completes your computer system.

Model 4 **Technical Reference Manual** enriches your training.

Bite-size lessons build knowledge and understanding step by step.

Do It at Home in Your Spare Time

NRI trains you in your own home at *your* convenience. You learn at your own pace, backed at all times by your own NRI instructor and the entire NRI staff of educators and engineers. They're always ready to answer questions and help you over the rough spots to keep you moving toward your goal.

COMPUTER • MONITOR • DISC DRIVE • PRINTER • AND SOFTWARE...

It's all included with NRI at-home training in Microcomputers and Microprocessors. And *only* with NRI training.

If you really want to learn how to work on computers, you have to learn how to work on computer systems. Only NRI includes a complete system with your training...total training that gives you hands-on experience with computers, peripherals, and software.

Get Inside the Famous TRS-80 Model 4 Computer

You'll work with one of the most popular computers ever designed. Learn programming techniques for trouble-shooting...go inside to expand its RAM from 16K to a powerful 64K... even install a double density floppy disc drive.

You'll become familiar with computer languages...run software that demonstrates computer operation...even become proficient at modifying programs to fit your business or family computing needs.

Add High Performance Peripheral Printer

Only NRI gives you total system training with the flexible Gemini 10X printer as part of your hands-on learning. Working with it, you'll get practical experience in adding peripherals, learn maintenance and servicing operations. And you'll use it for hard copies of important programs, business records, and word processing.

Understanding Built On Experience

You start with the basics, rapidly build on the fundamentals of electronics with bite-size lessons to master advanced concepts like digital logic, microprocessors, and computer memories.

You'll use the exclusive NRI Discovery Lab® to see what makes transistors tick, build and test working electronic circuits that duplicate key computer circuitry, construct digital logic circuits that demonstrate computer performance. And, you'll probe into the very heart of your computer using the Discovery Lab and professional digital multimeter included in your training program.

Free Catalog Tells Much More... Send Today

See all the interactive training equipment you get in this state-of-the-art course in today's technology. Look over the lesson outlines and descriptions of the exciting experiments you'll perform. And look at other high-tech career fields like Robotics, TV/Video/ Audio Servicing and Data Communications.

Mail the coupon today to see how NRI can give you the skills and confidence you need to start a good-paying career as a microcomputer technician.

NRI SCHOOLS
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, N.W., Washington, DC 20016
We'll give you tomorrow.

All Career courses approved under GI bill.
☐ Check for details

CHECK ONE FREE CATALOG ONLY

☐ Computer Electronics with Microcomputers
☐ Data Communications
☐ Robotics & Industrial Controls
☐ Color TV, Audio, and Video System Servicing
☐ Electronics Design Technology
☐ Digital Electronics

☐ Communications Electronics
  • Mobile • Aircraft • Marine
☐ CB & Cellular Radio
☐ Industrial Electronics
☐ Basic Electronics
☐ Small Engine Servicing

☐ Appliance Servicing
☐ Automotive Servicing
☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
☐ Building Construction

Name (Please Print) _____ Age _____

Street _____

City/State/Zip _____

Accredited by the National Home Study Council

198-094

TRS-80 is a trademark of the Radio Shack division of Tandy Corp.

www.commodore.ca

"We know that if a parent buys it for a fifth-grade student and he or she gets better grades, then the parent comes back, says it worked, and buys it for the sixth-grade child."

The movement of software from the school to the home is accelerating. Control Data Corporation, developers of the widely used Plato system of educational software, now has more than 12,000 hours of courses that include everything from instruction in basic skills and computer literacy to educational games and advanced calculus. And the company is in the process of converting many of its packages for use on personal computers like the Apple II family, Atari, IBM, and Texas Instruments.

The More The Merrier

A related trend over the last year or so is the entry of large corporations, especially established educational publishers, into the home software field. Companies like Random House, Prentice-Hall (Arrakis Systems), SFN Companies (Mindscape), Xerox (Weekly Reader Software), Scholastic, Simon & Schuster, and Fisher-Price are bringing their talent, marketing muscle, and reputations into the computer learning field.

"And that's very good," says Fritz Luecke, manager of computer software for Weekly Reader Software. "Most of them bring something to the business that other traditional software companies might not bring, and that is a different kind of marketing expertise.

"They also bring, in many cases, a new way of packaging, a new way of adding things to a program to make it a total program, which is what we feel we have done with our products," he adds.

What Children And Parents Want

Giving children items with

Choosing The Best Educational Software

Selby Bateman, Features Editor
Sharon Darling, Research Assistant

Of the thousands of educational software programs available, which ones are worth your time and money? Here's a short lesson from the experts.

"One recommendation I always make to parents—one that not enough people are making—is that you should really begin to look for educational software for your three- or four-year old child," says William Bowman, chairman of Spinnaker Software.

"That's the time to begin thinking about buying a home computer and getting educational software. It's going to be easier for you to find things that are generally applicable to the learning skill areas of young children than it will be to find the more specific software for older kids," he adds.

Getting Involved
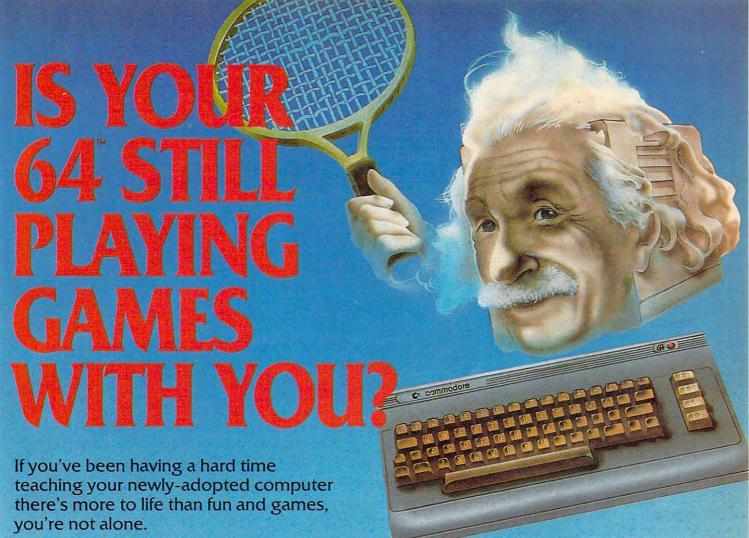
Spinnaker's educational software lines include the new Fisher-Price Learning Software for children from three to twelve years of age, as well as such best-selling programs as *Kindercomp*, *Alphabet Zoo*, *In Search of the Most Amazing Thing*, and *Kids On Keys*.

"The next real criterion is that the software's got to be fun," says Bowman. "If it's not fun to use, kids won't use it no matter what their ages are—and that's where an awful lot of educational software falls down. The next thing for a parent to do is to consider how much he or she is going to be involved with the child. A little involvement is always required. But some software, like Spinnaker's, really expects the parent to be more involved," he adds. "The software is a little bit deeper, it's a little more advanced in terms of what you can do with it."

Trying It On For Size

Getting educators and software developers to agree on a set of specific guidelines would be almost impossible, given the many conflicting views which abound in the computer-based learning field. But there are a few fundamental pieces of advice for anyone buying educational software. The following tips from educators, software houses, and leading independent program developers may be of help as you wade through the flood of educational software packages:

• **Shop at a software store that allows you to run some programs prior to purchase, or that lets you buy packages on approval.** Such stores may not be easy to find right now, but retailers are discovering that an increasing number of customers are demanding more than just the promises on the outside of a package to buy an educational software product.

IS YOUR 64™ STILL PLAYING GAMES WITH YOU?

If you've been having a hard time teaching your newly-adopted computer there's more to life than fun and games, you're not alone.

Now, you can introduce your Commodore 64™ to the Work Force: affordable, easy-to-use software and hardware that will unleash the power you always expected from your Commodore 64™, but thought you might never see.

PaperClip™

is simply the best word processing program of its kind—loaded with advanced features, yet so easy to use even a novice can get professional results. With **SpellPack™**, it even corrects your spelling! Once you've tried it, you'll never use a typewriter again.

The Consultant™
(formerly Delphi's Oracle)

is like a computerized filing cabinet with a brain. Organize files for recipes, albums, or the membership of your service club. Then search, sort, arrange and analyze your information with speed and flexibility that's simply astounding.

SpellPack™

teaches your 64 to spell. It checks an entire document in 2 to 4 minutes against a dictionary of over 20,000 words. And you can add up to 5,000 of your own specialized terms. Type letter perfect every time!

BusCard II™

is a magic box that lets you transform your humble home computer into a powerful business machine. It gives you the added power of BASIC 4.0, and lets you add IEEE disk drives, hard disk, virtually any parallel printer, and other peripherals without extra interfaces. Completely software invisible.

B.I.-80™ Column Adaptor

gives you crystal clear 80 column display. Using the highest quality hardware, we've eliminated the problems of snow, fuzziness and interference. Basic 4.0 commands greatly simplify disk drive access. Switches easily from 40 to 80 column display.

Discover the true power of your Commodore 64™. Ask your dealer about the Commodore 64™ Work Force, from Batteries Included—the company that doesn't leave anything out when it comes to making things simple for you.

BATTERIES INCLUDED
"Excellence in Software"

These products have been developed specifically for Commodore computers by Batteries Included and are totally compatible with each other. For a full color brochure write to:

186 Queen Street West, Toronto, Canada M5V 1Z1 (416) 596-1405 / 3303 Harbor Blvd., Costa Mesa, CA. 92626 (714) 979-0920

64 AND COMMODORE 64 ARE REGISTERED TRADE MARKS OF COMMODORE BUSINESS MACHINES

Cwww.commodore.ca

which they are familiar—books, stickers, posters—in addition to the software package makes it easier for the youngsters to relate to the product, Luecke says.

While numerous educational software companies are expanding their operations from school orientation to include the home market, the growth is very much a two-way street. Other companies, which have previously sold software strictly to the home, are approaching schools with their products. And the success which some microcomputer hardware manufacturers—especially Apple Computer—have had in donating products to school systems, is now spurring more software companies to try similar tactics.

Using the slogan, "Be a hero, software your school," Scarborough Systems of Tarrytown, New York, is embarking on a major school-donation project this fall.

Those who purchase Scarborough educational programs between September 15 and December 15 will find a coupon in each package enabling them to donate a piece of educational software to the school and teacher of their choice. The donated program does not have to be the same one purchased.

The purchaser sends the coupon and $3.50 for postage and handling to Scarborough. The company will then donate the software to the school. A gift card telling the name of the donor and the name of the dealer where the software was purchased will be included in the donation.

"We think it's a unique program, and it may provide up to $4 million in the value of computer software—that's 125,000 programs," says Sanford K. Bain, vice president of marketing for Scarborough Systems. The company has computer software for four machines—the Apple II family, Commodore 64, IBM PC/XT/jr, and the Atari

"Buying blind" is the way Dr. Sam Barkliss, chief executive officer of Computerose, Inc., an educational software firm, describes the predicament most parents and some teachers find themselves in when purchasing educational packages. They should be offered the opportunity to test the educational value of software before a purchase.

But What Does It Really Do?

• **Determine what age and learning levels the software is designed for.** Once that is established, find out exactly what the program intends to teach, says Leigh Mosley, an educational consultant at Peachtree Software. "A parent should ask, 'Is my son or daughter going to learn from it?'"

Some companies are better than others at telling you who the target audience is, what level of learning is required, and what the goals of the packages are. Always keep in mind the software user—the child's age, learning level, interests, and dislikes.

• **Be aware that the nature of "educational content" is often difficult to assess—and usually the subject of much debate among educators and software houses.** Many educators and software producers believe that specific learning objectives are crucial in producing good computer-based educational products. Others stress the validity of programs which invite youngsters to explore and "play" in a less structured learning environment.

The General And The Specific

For example, software that stimulates a student's creativity might be more useful than a program that deals with a specific learning problem in school, notes Kent Kehrberg, director of software for the Minnesota Educational Computing Consortium.

"It may be difficult for a parent to match up a very specific program with a problem a child is having in, say, algebra," he says. "In a case like that, it's very difficult for someone besides the teacher to pinpoint [the problem]."

• **Read published reviews and other articles about software packages and the goals of various software companies.** The more knowledgeable you are about manufacturers and their products, the easier your task when picking out new software.
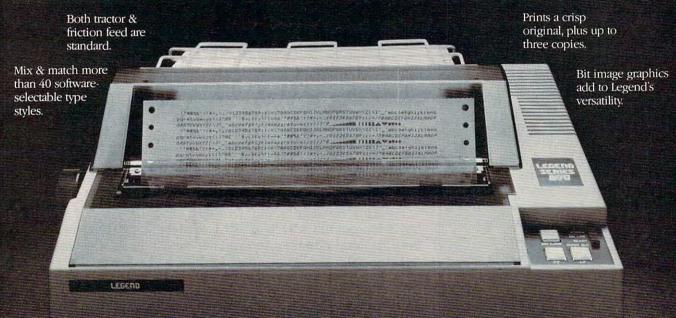
Tutorials, Simulations, And Drill-And-Practice

For example, when shopping for software, parents should know the three basic types of educational programs—tutorials, drill-and-practice, and simulations, notes Sherwin A. Steffin, vice president for research and development at Eduware Services, Inc.

"Tutorials help you attain a new skill or understanding," he says. "They generally ask the question, 'How to?' With drill-and-practice, you already know how, but want to know better, so you need repetitive exercises." Simulations can offer examples of the way things work in just about any field.

• **Consider how much replay value, or depth, a product has.** Will the child use the package a few times and tire of it, or is there enough variety and challenge to offer a stimulating environment?

family.

"Obviously, we're doing it because we think it reflects a favorable image for the company," Bain says. "We strongly believe the validity for these programs has to come from the schools. And our donation program allows schools to sample software without great expense on their part."

To Network Or Not To Network?

One school-based trend that disturbs DLM President Andrew W. Bingham is the tendency for schools to network their computers, which Bingham says can work against innovative use of the computers.

"It almost scares me, because I think the evolution may ultimately lead us to miss the revolutionary potential of software," he says. "When there are just two or three computers in a school, teachers will use them flexibly and individually.

"When the school gets more computers, they bring them all together into one room, because it's easier to control, easier to administer. Then they want a networking system," says Bingham. "What you get is students being marched to the computer for their 20 minutes of instruction and marched back to their seats."

Bingham urges parents and teachers to explore innovative uses of the computer rather than trying to emphasize convenience of management.

No matter what the area of computer-based educational programs, software companies and educators all agree that the field is still in its infancy in many ways. And expectations for the future are bright. "I think we're going to see an explosion in the next couple of years," says Sam Barkliss, chairman of Computerose, Inc., an educational software company. "I think we're going to see some astonishing things."

As William Bowman puts it, "Does the software provide multiple paths to creativity, and does it accept multiple paces from different children?"

The Price Tag For Learning

• **Price and future availability of software from the same company can be important factors.** Depending on the needs of the student and the goals of a company's software line, you may wish to purchase an entire series of complementary programs over time. How much this will cost you and its effects on the student's learning goals then become very important.

"A lot of people out there are getting too great a price for what they're selling, and for what the market will support," says George Esbensen, national sales coordinator for MicroEd, a Minnesota-based company which produces educational software. "A lot of what's being passed off as educational software is not."

• **Take your child to the store when selecting software.** This can be especially helpful—and save you later disappointments—if the child can actually see the program working in the store. If that's not possible, at least you can get a better idea of some of the likes and dislikes of the youngster for whom you're purchasing the program.

Sophistication, Power, And Interaction

• **How flexible is the software program? Are there built-in options which allow a variety of challenges and motivational changes as the user works with the program?** As computers become more powerful and software becomes more sophisticated, the level of interactivity between user and program is increasing dramatically. The best educational software takes advantage of that power and sophistication.

• **Is the software both easy to use and error-free?** Educational programs which freeze up or frustrate a user can immediately discourage users, especially younger children. Be aware of both potential problems as you evaluate software you see or that you've purchased.

Fritz Luecke, manager of computer software for Weekly Reader Software, suggests that you determine how easy and helpful the program guide booklet is that comes with the package. Many parents, teachers, and students want to be able to insert a program into the computer and use it without having to use a guide, particularly if that guide is confusing or incomplete.

NEA Teacher Certified

Finally, you might want to look at the NEA catalog of educational software. With the proliferation of educational software packages, the job of separating the good from the mediocre gets tougher every day. For more than a year now, the National Education Association (NEA) has been trying to give some guidance in this area. Approximately 50 NEA reviewers have been testing educational programs submitted by software authors and publishers. Those which meet the NEA's stiff requirements are given an "NEA Teacher Certified" stamp of approval, and are included in a catalog of approved software. ©

C=www.commodore.ca

The Tester

Joseph Kattan

Multiple-choice tests have been standard for years. Now anyone who wants to create a multiple-choice test on his home computer can easily do so. Tests can be devised to teach youngsters anything from history to sports trivia. Just fill in the DATA statements with the questions and answers and the program will do the rest. Versions that work on the Commodore 64, VIC-20, PETs, Apples, Radio Shack Color Computer, Coleco Adam, IBMs and TI-99/4A are included.

One of the most frequently cited reasons for purchasing a personal computer is its great potential as an educational tool. The home computer is widely used to teach children to spell or perform simple arithmetic calculations and to drill adults on anything from foreign languages to organic chemistry. A well-designed instructional program can produce spectacular results in improving a student's command of a subject. A poorly designed program, on the other hand, will frustrate, rather than teach.

Computers, for better or worse, cannot yet think. If the programmer tells the computer that Washington is the capital of the United States, the program will be less than kind to the hapless user who responds that Washington, D.C., is the capital. If the programmer tells the computer that *avoir* means "to have," pity the French student who answers "to possess." These are, of course, soluble problems, but they illustrate one weakness in computerized instruction. No matter how well designed the program, there will always exist a correct answer to some question that the program will not recognize. You can instruct a program to accept Washington, Washington, D.C., and Washington, DC, as the answer to the capital question, but how do you cover all of the synonyms of a word like *fantastic* in a foreign language translation program?

No Ambiguity

There is one kind of testing that a computer handles exceptionally well, because it is not required to reason: multiple-choice. It takes little effort to insure that the answers to a multiple-choice test are free of ambiguity, which is why all of the standardized testing in our schools tends to be multiple-choice. These tests, moreover, require less effort on the part of the user than answer-oriented tests, and can be used together with the more rigorous answer-oriented tests to form a very effective instructional package.

Remembering To Answer

Designing an effective program for a multiple-choice test is no easy matter, however. A simple and commonly used algorithm selects a question from DATA statements at random and then reads four or five different answers from the DATA statement, together with a code that identifies the correct answer.

There are several deficiencies to this solution. For one, it consumes tremendous amounts of memory, as it gobbles up bytes both for the correct answer and for the dummy answers that have no use other than to serve as the incorrect choices. In addition, the program user is always presented with the same set of choices, and in the same order, for each question. The user may well get into the habit of remembering that the answer to a question is C without learning the answer itself.
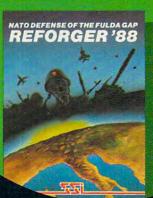
A more elegant solution should present truly random choices for each question. The user should rarely, if ever, encounter the same choices for a given question. The program, moreover, will be more compact because every answer in its DATA statements will be a correct answer to some question. With this method, the program will select a question from a DATA statement at random, will read the answer to that question from the same DATA statement, and then read four more answers at random to present as false choices. This method insures that the same answer is not presented as two separate choices (since random selection could cause that result) and arranges the order of the answers at random. The U.S. capital may be C on one run of the test, but A or B or D on another.

COMPUTE! BOOKS

COMPUTE's Second Book of Commodore 64 Games

Sixteen outstanding games, from detailed simulations to fast-paced machine language arcade games. Run for president in "Campaign Manager," defend yourself against sharks and acid rain in "The Frantic Fisherman," or defend a world in "Astro-PANIC!" Photograph the Loch Ness monster, battle in an arena, or establish trade routes to India. The best games from *COMPUTE!* and *COMPUTE!'s Gazette*, and some appearing here for the first time anywhere. Articles show you how to design a videogame and explain the intricacies of writing your own text adventure.

$12.95
ISBN 0-942386-64-7

COMPUTE!'s Beginner's Guide To Commodore 64 Sound

Clear descriptions and numerous examples show how to program sound and music on the 64. Well-known Commodore author John Heilborn teaches both beginning and advanced programmers the ins and outs of the SID chip, explaining how to create simple sounds, how to utilize the computer's advanced sound features, and even how to combine sound and graphics in an arcade-style game. Also included are two sound editors, which make composing easy, and dozens of sample sound effects you can type in and hear.

$12.95
ISBN 0-942386-54-X
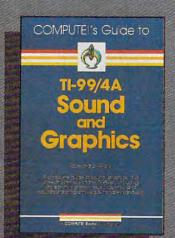
COMPUTE!'s Machine Language Routines For The Commodore 64

Sometimes BASIC is just not fast enough, or versatile enough, for programming on the Commodore 64. Machine language (ML) is. But creating ML routines can be difficult for the beginning or intermediate programmer. This collection offers ready-to-use ML routines that can be easily inserted into BASIC programs, as well as stand-alone routines that make the 64 even more powerful. Valuable programming aids such as one-key commands, automatic line numbering, and program searches are here. Routines useful in games and graphic design are also included. There are even complete machine language programs to help in the creation of custom characters and sprites.

$14.95
ISBN 0-942386-48-5
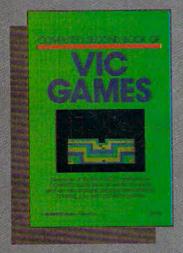
COMPUTE!'s Guide to TI Sound & Graphics

Using dozens of examples and clear, nontechnical explanations, *COMPUTE!'s Guide to TI Sound and Graphics* introduces you to the tremendous sound and graphics capabilities of your TI home computer. This book includes exciting arcade-style games, challenging educational programs, a versatile sprite editor and many more useful ready-to-type-in programs. Whether you're a beginning TI user or an experienced programmer, this is a book you'll refer to again and again.

$12.95
ISBN 0-942386-46-9

Segregated Data

Even this method, however, has a potential pitfall, and the solution discussed here resolves it. Let's take a U.S. history test as an example. The answers to such a test may be George Washington, or Philadelphia, or 1776. Obviously, it would be quite absurd to present Philadelphia or 1776 as possible answers to a question calling for the name of the first president. The solution to this problem is to segregate the DATA statements containing the questions and answers into different areas of the program and to select answers to each question only from a valid area. For example, DATA statements between 1000 and 1999 could contain questions concerning names of persons; DATA statements between 2000 and 2999 could contain questions concerning places; and so forth. The program chooses an area at random and then stays in that area to present the incorrect answers. If the answer to the question selected is George Washington, the four incorrect answers will always be names of persons.

This solution has an added advantage. It allows for the inclusion of diverse subjects on a single test, with the testing either confined to a subject chosen by the user or mixed at random by the computer. This program is intended primarily for single-subject tests that require the segregation of answers by types, as in the U.S. history example above.

How It Works

The program relies on two arrays—Q, which stores the number of questions in each category, and T, which selects the answers at random. The variable N is used to store the group of questions and answers to be called. The question and answer are selected by the computer at line 300. Since answer groups are stored in DATA statements beginning with 1001, 2001, and so forth, the program adds that number to a random number from 1 to the number of questions for the appropriate group, as indicated in the Q array. For example, if the question and answer are to come from group 1, for which questions and answers are stored in lines 1001 through 1011, the program looks to the variable Q(1) to ascertain the range of random numbers to generate.

Once a question and answer have been read, the program uses the same random number formula to look up the incorrect answers. It stores the random numbers (data line numbers) in a T array (line 330) and makes sure that none of the numbers in that array is equal to the line number of the correct answer or to the number of another element in the array (line 335). At line 340, the program chooses where to place the correct answer, which can be any choice from one to five, and then proceeds to place all of the choices on the monitor or television screen. Once an answer is entered, the program indicates whether or not it was correct. In the case of a correct answer, the program waits for two seconds (lines 400 and 410) and then constructs a new screen. If the answer entered is incorrect, the program waits for the RETURN key to be pressed before moving on to the next screen.

Screening Keys

One other matter is the little subroutine beginning at line 800. The entire subroutine could be replaced with a single INPUT statement. The advantage of the subroutine is that it screens out unwanted keys (in this case, anything but a number) and maintains the integrity of the screen display. In addition, the subroutine does not attach a question mark to a prompt, allowing you to insert it where it is appropriate and omit it where it is not. Study the DATA statements beginning at line 1001 and you can see the flexibility afforded by the subroutine.

This program is obviously meant to be modified. When modifying it for your own use, pay special attention to the Q array. The array should be DIMensioned to the number of answer categories in the program. The same number should be placed in the variable SUBJ. The elements of the Q array should be equated to one more than the number of questions in the appropriate category. Finally, the questions and answers should be placed on the same DATA statement, and the DATA statements should be arranged in increments of one beginning with a line number of N*1000+1, N being the number of the group. Make sure that neither the questions nor the answers contain any commas, since the BASIC interpreter will take the commas to indicate the end of a string.

Program 1: The Tester, Atari Version

Refer to the "Automatic Proofreader" article before typing this program in.

```
PN 10 REM   THE TESTER
DK 20 REM   MULTIPLE-CHOICE TEST MAKER
BL 40 REM   THIS PROGRAM WILL GENERATE
OJ 50 REM   MULTIPLE-CHOICE TESTS FOR
        ONE
IB 60 REM   OR MORE SUBJECTS
CI 100 POKE 710,78:POKE 712,78:POKE
        709,4:POKE 752,1:REM SET SCRE
        EN COLORS, TURN CURSOR OFF
MJ 110 OPEN #1,4,0,"K:"
JF 120 DIM TITLE$(38),QUES$(38),ANS$
        (30),TEMP$(30),NUM$(3),BL$(30
        ),Q(4),T(3)
DJ 130 TITLE$=" ":TITLE$(38)=TITLE$:
        TITLE$(2)=TITLE$:QUES$=TITLE$
        :ANS$=TITLE$(1,30):TEMP$=ANS$
        :BL$=ANS$
```
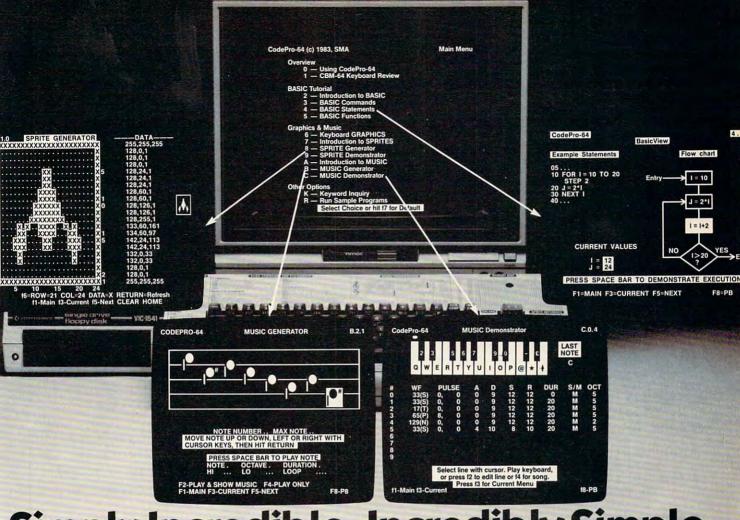
Simply Incredible. Incredibly Simple. CodePro-64 For Your Commodore.
Instructional Software That Will Amaze You With What You Learn.

CodePro-64 is the easy way to learn.
CodePro-64 astonishes you with how much you can learn. And how easy learning can be. With CodePro-64, you'll learn to write your own programs. You'll develop advanced programming skills with graphics, sprites and music. And best of all, you'll learn visually.

CodePro-64 is a complete package.
First and foremost, it's a comprehensive BASIC language tutorial. With it, you learn by seeing as well as by doing. You watch variables change value. You see statement branching occur. You can even see "For-Next" statements executed.

There's graphics instruction, too.
CodePro-64 also has a SPRITE Demonstrator and Generator. With it, you'll master the concepts and techniques of graphics programming with sprites. And you'll be able to create and save your own sprites for use in your programs.

And we don't forget the music.
You'll use the MUSIC Demonstrator and Generator to learn how to get the musical effect you want. Exactly. And in seconds. You'll use it to compose simple tunes on the screen and then save the tune to a diskette file.

Discover how good CodePro-64 really is.
We ship CodePro-64 as a package of 16 integrated programs, distributed on two diskettes, and containing sample programs for unlimited use. It is supported by a 140-page manual and is packaged in an attractive three-ring binder.

CodePro-64. Incredible. And simple.

HOW TO ORDER

Send personal check, money order or MasterCard/VISA credit card information. Please add $3.00 for shipping and handling per order; foreign orders must add $15.00 per unit (except Canada). US funds only. Sorry, but no COD's. NC residents add 4½% sales tax. Corporate quantity discounts available. Dealer inquiries invited.

And for faster service on credit card orders call Toll Free 1-800-762-7874. In North Carolina call 919-787-7703.

SYSTEMS MANAGEMENT ASSOCIATES
3700 Computer Drive, Dept. CPC
Raleigh, North Carolina 27609

CodePro-64 is a trademark of Systems Management Associates. ©SMA 1984

Our Guarantee. Use CodePro-64 for 10 days. If you are not completely satisfied return it to us (undamaged) for a full refund.

```
OL 140 Q(1)=11:Q(2)=11:Q(3)=11:Q(4)=
        11:REM INSERT HERE ONE LESS T
        HAN NUMBER OF QUESTIONS FOR E
        ACH CATEGORY
DJ 200 ? "{CLEAR}{4 SPACES}PLEASE CH
        OOSE ONE SUBJECT: "
MC 205 SUBJ=4:REM SUBJ IS NUMBER OF
        SUBJECTS IN TEST
PK 210 ? :RESTORE 900:FOR I=1 TO SUB
        J:READ TITLE$:? I;") ";TITLE$
        :NEXT I:?
FP 220 GOSUB 800:IF N1<1 OR N1>4 THE
        N POSITION C,R:GOTO 220
DN 230 N=N1:? :? "HOW MANY QUESTIONS
        ? ";
NB 240 GOSUB 800:IF N1<1 THEN POSITI
        ON C,R:GOTO 240
HJ 250 TOTAL=N1:RESTORE 900:FOR I=1
        TO N:READ TITLE$:NEXT I:TITLE
        $(LEN(TITLE$)+1)=" TEST"
KI 300 SCORE=0:FOR COUNT=1 TO TOTAL:
        ? "{CLEAR}":POKE 85,(40-LEN(T
        ITLE$))/2:? TITLE$:?
NG 310 ANS=N*1000+1+INT(RND(0)*Q(N))
        :RESTORE ANS:READ QUES$:? QUE
        S$:? :READ ANS$
AI 320 FOR I=0 TO 3
NH 330 T(I)=N*1000+1+INT(RND(0)*Q(N)
        ):IF T(I)=ANS THEN 330
PJ 335 FOR J=0 TO 3:IF I<>J AND T(I)
        =T(J) THEN POP :GOTO 330
DM 340 NEXT J:NEXT I:J=1+INT(RND(0)*
        5):A=0:FOR I=1 TO 5:? I;") ";
AK 350 IF I<>J THEN RESTORE T(A):REA
        D QUES$:READ TEMP$:? TEMP$:A=
        A+1:GOTO 370
NO 360 ? ANS$
JL 370 NEXT I:?
GO 380 GOSUB 800:IF N1<1 OR N1>5 THE
        N POSITION C,R:GOTO 380
DE 390 IF N1<>J THEN 430
DH 400 SCORE=SCORE+1:? :POKE 85,18:?
        "GOOD !":POKE 20,0
GF 410 IF PEEK(20)<90 THEN 410
GJ 420 GOTO 460
PK 430 ? :? "SORRY, THE ANSWER IS ";
        :IF LEN(ANS$)>18 THEN ?
II 440 ? ANS$:? :? "{6 SPACES}PRESS
        RETURN TO CONTINUE"
PD 450 POKE 764,255:GET #1,A:IF A<>1
        55 THEN 450
GC 460 NEXT COUNT
EI 500 ? "{CLEAR}":POKE 85,(40-LEN(T
        ITLE$))/2:? TITLE$:?
CM 510 ? "OUT OF ";TOTAL;" QUESTIONS
        , YOU ANSWERED":? SCORE;" COR
        RECTLY. YOUR SCORE IS ";
NN 520 ? INT((SCORE/TOTAL)*100+0.5);
        "%."
HM 530 ? :? "RETURN FOR ANOTHER TEST
        , * TO END"
OH 540 POKE 764,255:GET #1,A:IF A<>1
        55 AND A<>42 THEN 540
AD 550 IF A=155 THEN 200
KM 560 ? "{CLEAR}":END
KG 800 C=PEEK(85):R=PEEK(84)
NG 805 POSITION C,R:? BL$:TRAP 805:Y
        =1:S=0:NUM$=""
KH 810 POSITION C+Y-1,R:? "{T}";:POK
        E 764,255:GET #1,A:POSITION C
        +Y-1,R:IF A<>126 THEN 825
```

```
AP 815 IF Y>1 THEN Y=Y-1:? " ";:NUM$
        =NUM$(1,Y)
GM 820 GOTO 810
AI 825 IF A=155 THEN ? " ":GOTO 840
DO 830 NUM$(Y,Y)=CHR$(A):IF Y=1 THEN
        S=A
DO 835 ? CHR$(A);:Y=Y+1:GOTO 810
NO 840 POKE ADR(NUM$),S:N1=VAL(NUM$)
        :RETURN
EJ 900 DATA HISTORICAL FIGURES,WORLD
        CITIES,NEWSMAKERS,COMPUTERS
KM 1000 REM HISTORICAL FIGURES
NK 1001 DATA THE FIRST PRESIDENT,GEO
         RGE WASHINGTON
BK 1002 DATA FIRST CHIEF JUSTICE,JOH
         N JAY
EL 1003 DATA WHO INVENTED COTTON GIN
         ? ELI WHITNEY
GL 1004 DATA PRESIDENT OF THE CONFED
         ERACY,JEFFERSON DAVIS
FH 1005 DATA WHO DEBATED LINCOLN?,ST
         EPHEN DOUGLAS
FK 1006 DATA FAMOUS ABOLITIONIST,FRE
         DERICK DOUGLASS
DL 1007 DATA BRITISH NAVIGATOR,FRANC
         IS DRAKE
IH 1008 DATA THE THIRD PRESIDENT,THO
         MAS JEFFERSON
```

C-www.commodore.ca

```
NF 1009 DATA HE PRESIDED OVER THE NE
        W DEAL,FRANKLIN DELANO ROOSE
        VELT
JM 1010 DATA WHO WAS ASSASSINATED IN
        DALLAS?,JOHN F. KENNEDY
PD 1011 DATA RAILROAD MAGNATE,CORNEL
        IUS VANDERBILT
HP 2000 REM WORLD CAPITALS
IM 2001 DATA WHAT IS THE CAPITAL OF
        JAPAN?,TOKYO
IP 2002 DATA WHERE IS TRAFALGAR SQUA
        RE?,LONDON
EC 2003 DATA WHERE IS THE WESTERN WA
        LL?,JERUSALEM
NH 2004 DATA WHERE IS THE COLISEUM?,
        ROME
OH 2005 DATA WHERE IS RED SQUARE?,MO
        SCOW
KC 2006 DATA WHERE IS THE TAJ MAHAL?
        ,AGRA
CF 2007 DATA WHAT IS THE CAPITAL OF
        AFGHANISTAN?,KABUL
AH 2008 DATA THE PARTHENON IS IN THI
        S CITY,ATHENS
HO 2009 DATA CITY LEASED BY ENGLAND
        FROM CHINA,HONG KONG
KH 2010 DATA WHERE IS THE LINCOLN ME
        MORIAL?,WASHINGTON
CF 2011 DATA THIS CITY IS FAMOUS FOR
         ITS CANALS,VENICE
KH 3000 REM NEWSMAKERS
KE 3001 DATA PRESIDENT OF FRANCE,MIT
        TERAND
OD 3002 DATA FORMER SECRETARY OF STA
        TE UNDER REAGAN,HAIG
ID 3003 DATA SOVIET FOREIGN MINISTER
        ,GROMYKO
NA 3004 DATA GERMAN CHANCELLOR,KOHL
EB 3005 DATA SECRETARY OF TREASURY,R
        EGAN
KN 3006 DATA CHAIRMAN OF SOVIET COMM
        UNIST PARTY,CHERNENKO
EB 3007 DATA CANADIAN PRIME MINISTER
        ,TRUDEAU
JA 3008 DATA JAPANESE PRIME MINISTER
        ,NAKASONE
ME 3009 DATA SECRETARY OF STATE,SHUL
        TZ
HL 3010 DATA MAYOR OF NEW YORK,KOCH
FL 3011 DATA BRITISH PRIME MINISTER,
        THATCHER
GK 4000 REM COMPUTERS
ME 4001 DATA THE BRAIN OF THE COMPUT
        ER IS CALLED,CENTRAL PROCESS
        ING UNIT
AN 4002 DATA THE FASTEST WAY TO PROC
        ESS DATA,MACHINE LANGUAGE
CN 4003 DATA A SIMPLE COMPUTER LANGU
        AGE,BASIC
CI 4004 DATA USES REVERSE POLISH NOT
        ATION,FORTH
PI 4005 DATA 8 BITS ON THE 6502,BYTE
PM 4006 DATA A 16-BIT MICROPROCESSOR
        ,68000
GF 4007 DATA TRANSFERS DATA FROM MEM
        ORY TO CPU,BUS
AA 4008 DATA DATA ARE PUSHED AND PUL
        LED HERE,STACK
DO 4009 DATA USED FOR INDEXING,X REG
        ISTER
NG 4010 DATA FASTEST COMPUTATIONS TA
```

```
        KE PLACE HERE,PAGE ZERO
OO 4011 DATA A NUMBERING SYSTEM,HEXA
        DECIMAL
```

Program 2: The Tester, Microsoft Version

Refer to the "Automatic Proofreader" article before typing this program in.

```
20 REM *** MULTIPLE CHOICE TEST MAKER ***
                                        :rem 9
110 DIM QA$(44),AN$(44):GOSUB 930:FOR I=1
    TO 44                              :rem 172
120 READ QA$(I),AN$(I):NEXT I:FORJ=1 TO 4
    :Q(J)=10:READ TL$(J):NEXT J        :rem 38
200 GOSUB 930:PRINT "PLEASE CHOOSE ONE SU
    BJECT:"                            :rem 96
210 PRINT:FOR I=1 TO 4:PRINT I;") ";TL$(I
    ):NEXT I                           :rem 226
220 PRINT:PRINT "YOUR CHOICE";:INPUT Z$:N
    =VAL(Z$):IFN<1 OR N>4 THEN 200
                                        :rem 198
230 PRINT:PRINT:PRINT "HOW MANY QUESTIONS
    ";:INPUT Z$:N1=VAL(Z$)             :rem 13
240 IF N1=0 THEN 230                   :rem 213
300 TL=N1:SC=0:FOR CT=1 TO TL:GOSUB 930:P
    RINT TL$(N);" TEST"                :rem 212
310 ANS=INT(RND(1)*Q(N))+(N*11)-10:PRINT:
    PRINT QA$(ANS)                     :rem 202
320 FOR I=0 TO 3:GOSUB 800:J=0         :rem 75
330 IF I<>J AND T(I)=T(J) THEN GOSUB 800:
    J=0:GOTO 330                       :rem 96
335 J=J+1:IF J<=3 THEN 330             :rem 76
340 NEXT I:J=1+INT(RND(1)*5):A=0:PRINT:PR
    INT:FOR I=1 TO 5:PRINT I;") ";     :rem 86
350 IF I<>J THEN PRINT AN$(T(A)):A=A+1:GO
    TO 370                             :rem 60
360 PRINT AN$(ANS)                     :rem 12
370 NEXT I:PRINT                       :rem 233
380 PRINT:PRINT "ENTER YOUR ANSWER";:INPU
    T Z$:N1=VAL(Z$)                    :rem 251
390 IF N1<1 OR N1>5 THEN 380           :rem 116
395 IF N1<>J THEN 430                  :rem 57
400 SC=SC+1:PRINT:PRINTTAB(9) "GOOD !":FO
    R I=1 TO 1000:NEXT I:GOTO 460      :rem 225
430 PRINT:PRINT "SORRY, THE ANSWER IS:":P
    RINT AN$(ANS)                      :rem 46
440 PRINT:PRINT "<RETURN> TO CONTINUE"
                                        :rem 146
450 INPUT Z$                           :rem 167
460 NEXT CT                            :rem 112
500 GOSUB 930:PRINT TL$(N);" TEST"
                                        :rem 154
510 PRINT:PRINT:PRINT "TOTAL QUESTIONS ";
    TL:PRINT "CORRECT ANSWERS ";SC:rem 48
520 PRINT "YOUR SCORE IS ";INT((SC/TL)*10
    0+.5);"%"                          :rem 233
530 PRINT:PRINT "WANT ANOTHER TEST(Y/N)"
                                        :rem 226
540 Z$="":INPUT Z$:IF Z$<>"Y" AND Z$<>"N"
    THEN 530                           :rem 98
550 IF Z$="Y" THEN 200                 :rem 66
560 GOSUB 930:END                      :rem 200
800 T(I)=INT(RND(1)*Q(N))+(N*11)-10:IF T(
    I)=ANS THEN 800                    :rem 0
810 RETURN                             :rem 121
900 REM THIS SUBROUTINE CLEARS THE SCREEN
    . FOR APPLE COMPUTERS USE "HOME"
                                        :rem 195
910 REM FOR IBM PC AND PCJR.,          :rem 44
920 REM AND TRS-80 COLOR COMPUTERS USE "C
    LS" STATEMENT.                     :rem 23
```

```
930 PRINT"{CLR}":RETURN              :rem 26
1000 REM HISTORICAL FIGURES          :rem 172
1001 DATA THE FIRST PRESIDENT,GEORGE WASH
     INGTON                          :rem 218
1002 DATA FIRST CHIEF JUSTICE,JOHN JAY
                                     :rem 26
1003 DATA WHO INVENTED COTTON GIN,ELI WHI
     TNEY                            :rem 12
1004 DATA PRESIDENT OF CONFEDERACY,JEFFER
     SON DAVIS                       :rem 138
1005 DATA WHO DEBATED LINCOLN?,STEPHEN DO
     UGLAS                           :rem 87
1006 DATA FAMOUS ABOLITIONIST,FREDERICK D
     OUGLASS                         :rem 90
1007 DATA BRITISH NAVIGATOR,FRANCIS DRAKE
                                     :rem 59
1008 DATA THE THIRD PRESIDENT,THOMAS JEFF
     ERSON                           :rem 135
1009 DATA HE PRESIDED OVER THE NEW DEAL,F
     RANKLIN DELANO ROOSEVELT        :rem 213
1010 DATA WHO WAS ASSASSINATED IN DALLAS?
     ,JOHN F. KENNEDY                :rem 156
1011 DATA RAILROAD MAGNATE,CORNELIUS VAND
     ERBILT                          :rem 243
2000 REM WORLD CITIES                :rem 239
2001 DATA WHAT IS THE CAPITAL OF JAPAN?,T
     OKYO                            :rem 140
2002 DATA WHERE IS TRAFALGAR SQUARE?,LOND
     ON                              :rem 143
2003 DATA WHERE IS THE WESTERN WALL?,JERU
     SALEM                           :rem 66
2004 DATA WHERE IS THE COLISEUM?,ROME
                                     :rem 215
2005 DATA WHERE IS RED SQUARE?,MOSCOW
                                     :rem 231
2006 DATA WHERE IS THE TAJ MAHAL?,AGRA
                                     :rem 162
2007 DATA WHAT IS THE CAPITAL OF AFGHANIS
     TAN?,KABUL                      :rem 37
2008 DATA THE PARTHENON IS IN THIS CITY,A
     THENS                           :rem 7
2009 DATA CITY LEASED BY ENGLAND FROM CHI
     NA,HONG KONG                    :rem 126
2010 DATA WHERE IS THE LINCOLN MEMORIAL?,
     WASHINGTON                      :rem 167
2011 DATA THIS CITY IS FAMOUS FOR ITS CAN
     ALS,VENICE                      :rem 37
3000 REM NEWSMAKERS                  :rem 167
3001 DATA PRESIDENT OF FRANCE,MITTERAND
                                     :rem 164
3002 DATA FORMER SECRETARY OF STATE UNDER
     REAGAN,HAIG                     :rem 227
3003 DATA SOVIET FOREIGN MINISTER,GROMYKO
                                     :rem 131
3004 DATA GERMAN CHANCELLOR,KOHL  :rem 208
3005 DATA SECRETARY OF TREASURY,REGAN
                                     :rem 65
3006 DATA CHAIRMAN OF SOVIET COMMUNIST PA
     RTY,CHERNENKO                   :rem 173
3007 DATA CANADIAN PRIME MINISTER,TRUDEAU
                                     :rem 65
3008 DATA JAPANESE PRIME MINISTER,NAKASON
     E                               :rem 144
3009 DATA SECRETARY OF STATE,SHULTZ
                                     :rem 196
3010 DATA MAYOR OF NEW YORK,KOCH   :rem 123
3011 DATA BRITISH PRIME MINISTER,THATCHER
                                     :rem 91
4000 REM COMPUTERS                   :rem 106
4001 DATA THE BRAIN OF THE COMPUTER IS CA
```

```
LLED,CENTRAL PROCESSING UNIT:rem 196
4002 DATA THE FASTEST WAY TO PROCESS DATA
     ,MACHINE LANGUAGE               :rem 13
4003 DATA A SIMPLE COMPUTER LANGUAGE,BASI
     C                               :rem 45
4004 DATA USES REVERSE POLISH NOTATION,FO
     RTH                             :rem 40
4005 DATA 8 BITS ON THE 6502,BYTE:rem 248
4006 DATA A 16-BIT MICROPROCESSOR,68000
                                     :rem 252
4007 DATA TRANSFER DATA FROM MEMORY TO CP
     U,BUS                           :rem 18
4008 DATA DATA ARE PUSHED AND PULLED HERE
     ,STACK                          :rem 0
4009 DATA USED FOR INDEXING,X REGISTER
                                     :rem 62
4010 DATA FASTEST COMPUTATIONS TAKE PLACE
     HERE,PAGE ZERO                  :rem 214
4011 DATA A NUMBERING SYSTEM,HEXADECIMAL
                                     :rem 238
5000 DATA HISTORICAL FIGURES,WORLD CITIES
     ,NEWSMAKERS,COMPUTERS           :rem 117
```

Program 3: The Tester, TI-99/4A Version

```
100 REM   EXTENDED BASIC NOT REQUIRE
    D
110 DIM QA$(44),AN$(44)
112 CALL CLEAR
114 FOR I=1 TO 44
120 READ QA$(I),AN$(I)
122 NEXT I
123 SUBJ=4
124 REM   SET SUBJ TO # OF CATEGORIE
    S OF QUESTIONS
125 FOR J=1 TO SUBJ
126 Q(J)=10
128 READ TL$(J)
129 NEXT J
200 CALL CLEAR
202 PRINT "PLEASE CHOOSE ONE SUBJEC
    T:"::
210 FOR I=1 TO 4
212 PRINT I;") ";TL$(I)
214 NEXT I
220 PRINT
222 PRINT "YOUR CHOICE";
224 INPUT N
226 IF (N<1)+(N>4)THEN 224
230 PRINT ::::
232 INPUT "HOW MANY QUESTIONS ?":N1
234 IF N1=0 THEN 232
300 TL=N1
302 SC=0
304 FOR CT=1 TO TL
306 CALL CLEAR
308 PRINT TL$(N);" TEST"
310 RANDOMIZE
312 ANS=INT(RND*Q(N))+(N*11)-10
314 PRINT ::
316 PRINT QA$(ANS)
320 FOR I=0 TO 3
322 GOSUB 800
324 J=0
330 IF (I=J)+(T(I)<>T(J))THEN 335
332 GOSUB 800
333 J=0
334 GOTO 330
335 J=J+1
337 IF J<=3 THEN 330
340 NEXT I
```

```
341 RANDOMIZE
342 J=1+INT(RND*5)
344 A=0
346 PRINT ::
348 FOR I=1 TO 5
349 PRINT I;") ";
350 IF I=J THEN 360
352 PRINT AN$(T(A))
354 A=A+1
356 GOTO 370
360 PRINT AN$(ANS)
370 NEXT I
372 PRINT
380 PRINT
382 INPUT "ENTER YOUR ANSWER: ":N1
390 IF (N1<1)+(N1>5)THEN 380
395 IF N1<>J THEN 430
400 SC=SC+1
402 PRINT
404 PRINT TAB(9);"GOOD !"
406 FOR I=1 TO 500
408 NEXT I
410 GOTO 460
430 PRINT
432 PRINT "SORRY, THE ANSWER IS:"
434 PRINT AN$(ANS)
440 PRINT
442 PRINT "HIT <ENTER> TO CONTINUE"
450 INPUT Z$
460 NEXT CT
500 CALL CLEAR
502 PRINT TL$(N)
510 PRINT ::::
512 PRINT "TOTAL QUESTIONS ";TL
514 PRINT "CORRECT ANSWERS ";SC
520 PRINT "YOUR SCORE IS ON THE ";T
    L$(N);" TEST IS";INT((SC/TL)*10
    0+.5);"%."
530 PRINT
532 INPUT "WANT ANOTHER TEST (Y/N)?
    ":Z$
534 IF (Z$<>"Y")*(Z$<>"N")THEN 530
550 IF Z$="Y" THEN 200
560 CALL CLEAR
562 STOP
800 RANDOMIZE
801 T(I)=INT(RND*Q(N))+(N*11)-10
802 IF T(I)=ANS THEN 800
810 RETURN
1000 REM  HISTORICAL FIGURES
1001 DATA THE FIRST PRESIDENT,GEORG
     E WASHINGTON
1002 DATA FIRST CHIEF JUSTICE,JOHN
     JAY
1003 DATA INVENTED THE COTTON GIN,E
     LI WHITNEY
1004 DATA PRESIDENT OF CONFEDERACY,
     JEFFERSON DAVIS
1005 DATA WHO DEBATED LINCOLN?,STEP
     HEN DOUGLAS
1006 DATA FAMOUS ABOLITIONIST,FREDE
     RICK DOUGLASS
1007 DATA BRITISH NAVIGATOR,FRANCIS
      DRAKE
1008 DATA THE THIRD PRESIDENT,THOMA
     S JEFFERSON
1009 DATA HE PRESIDED OVER THE NEW
     DEAL,F. D. ROOSEVELT
1010 DATA WHO WAS ASSASSINATED IN D
     ALLAS?,JOHN F. KENNEDY
1011 DATA RAILROAD MAGNATE,CORNELIU
     S VANDERBILT
2000 REM   WORLD CITIES
2001 DATA WHAT IS THE CAPITAL OF JA
     PAN?,TOKYO
2002 DATA WHERE IS TRAFALGAR SQUARE
     ?,LONDON
2003 DATA WHERE IS THE WESTERN WALL
     ?,JERUSALEM
2004 DATA WHERE IS THE COLISEUM?,RO
     ME
2005 DATA WHERE IS THE RED SQUARE?,
     MOSCOW
2006 DATA WHERE IS THE TAJ MAHAL?,A
     GRA
2007 DATA WHERE IS THE CAPITAL OF A
     FGHANISTAN?,KABUL
2008 DATA THE PARTHENON IS IN THIS
     CITY,ATHENS
2009 DATA CITY LEASED BY ENGLAND FR
     OM CHINA,HONG KONG
2010 DATA WHERE IS THE LINCOLN MEMO
     RIAL?,WASHINGTON
2011 DATA THIS CITY IS FAMOUS FOR I
     TS CANALS,VENICE
3000 REM   NEWSMAKERS
3001 DATA PRESIDENT OF FRANCE,MITTE
     RAND
3002 DATA FORMER SECRETARY OF STATE
      UNDER REAGAN,HAIG
3003 DATA SOVIET FOREIGN MINISTER,G
     ROMYKO
3004 DATA GERMAN CHANCELLOR,KOHL
3005 DATA SECRETARY OF TREASURY,REG
     AN
3006 DATA CHAIRMAN OF SOVIET COMMUN
     IST PARTY,CHERNENKO
3007 DATA CANADIAN PRIME MINISTER,T
     RUDEAU
3008 DATA JAPANESE PRIME MINISTER,N
     AKASONE
3009 DATA SECRETARY OF STATE,SHULTZ
3010 DATA MAYOR OF NEW YORK,KOCH
3011 DATA BRITISH PRIME MINISTER,TH
     ATCHER
4000 REM   COMPUTERS
4001 DATA THE BRAIN OF THE COMPUTER
      IS CALLED,CENTRAL PROCESSING
     UNIT
4002 DATA THE FASTEST WAY TO PROCES
     S DATA,MACHINE LANGUAGE
4003 DATA A SIMPLE COMPUTER LANGUAG
     E,BASIC
4004 DATA USES REVERSE POLISH NOTAT
     ION,FORTH
4005 DATA 8 BITS ON THE 6502,BYTE
4006 DATA A 16-BIT MICROPROCESSOR,6
     8000
4007 DATA TRANSFER DATA FROM MEMORY
      TO CPU,BUS
4008 DATA DATA ARE PUSHED AND PULLE
     D HERE,STACK
4009 DATA USED FOR INDEXING,X REGIS
     TER
4010 DATA FASTEST COMPUTATIONS TAKE
      PLACE HERE,PAGE ZERO
4011 DATA A NUMBERING SYSTEM,HEXADE
     CIMAL
5000 DATA HISTORICAL FIGURES,WORLD
     CITIES,NEWSMAKERS,COMPUTERS ©
```

A Beginner's Guide To Typing In Programs

What Is A Program?

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. Most of the programs published in COMPUTE! are written in a computer language called BASIC. BASIC is easy to learn and is built into most computers (on some computers, you have to purchase an optional BASIC cartridge).

BASIC Programs

Each month, COMPUTE! publishes programs for many machines. To start out, type in only programs written for your machine, e.g., "TI Version" if you have a TI-99/4. Later, when you gain experience with your computer's BASIC, you can try typing in and converting certain programs from one computer to yours.

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one "right way" of stating something. Every letter, character, or number is significant. A common mistake is substituting a letter such as O for the numeral 0, a lowercase l for the numeral 1, or an uppercase B for the numeral 8. Also, you must enter all punctuation such as colons and commas just as they appear in the magazine. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

Braces And Special Characters

The exception to this typing rule is when you see the braces, such as {DOWN}. Anything within a set of braces is a special character or characters that cannot easily be listed in a printer. When you come across such a special statement, refer to the appropriate key for your computer. For example, if you have an Atari, refer to the "Atari" section in "How To Type COMPUTE!'s Programs."

About DATA Statements

Some programs contain a section or sections of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (called machine language); others contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could "lock up," or "crash." The keyboard, break key, and RESET (or STOP) keys may all seem "dead," and the screen may go blank. Don't panic – no damage is done. To regain control, you have to turn off your computer, then turn it back on. This will erase whatever program was in memory, so always SAVE a copy of your program before you RUN it. If your computer crashes, you can LOAD the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is RUN. The error message may refer to the program line that READs the data. *The error is still in the DATA statements, though.*

Get To Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program, so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you made a mistake? You can always retype the line, but you at least need to know how to backspace. Do you know how to enter inverse video, lowercase, and control characters? It's all explained in your computer's manuals.

A Quick Review

1. Type in the program a line at a time, in order. Press RETURN or ENTER at the end of each line. Use backspace or the back arrow to correct mistakes.

2. Check the line you've typed against the line in the magazine. You can check the entire program again if you get an error when you RUN the program.

3. Make sure you've entered statements in braces as the appropriate control key (see "How To Type COMPUTE!'s Programs" elsewhere in the magazine).

www.commodore.ca

How To Type COMPUTE!'s Programs

Many of the programs which are listed in COMPUTE! contain special control characters (cursor control, color keys, inverse video, etc.). To make it easy to tell exactly what to type when entering one of these programs into your computer, we have established the following listing conventions. There is a separate key for each computer. Refer to the appropriate tables when you come across an unusual symbol in a program listing. If you are unsure how to actually enter a control character, consult your computer's manuals.

Atari 400/800

Characters in inverse video will appear like: ▓INVERSE VIDEO▓ Enter these characters with the Atari logo key, {𝘈}.

When you see	Type		See	
{CLEAR}	ESC SHIFT <	⌐	Clear Screen	
{UP}	ESC CTRL –	↑	Cursor Up	
{DOWN}	ESC CTRL =	↓	Cursor Down	
{LEFT}	ESC CTRL +	←	Cursor Left	
{RIGHT}	ESC CTRL *	→	Cursor Right	
{BACK S}	ESC DELETE	◄	Backspace	
{DELETE}	ESC CTRL DELETE	▟	Delete character	
{INSERT}	ESC CTRL INSERT	▙	Insert character	
{DEL LINE}	ESC SHIFT DELETE	▚	Delete line	
{INS LINE}	ESC SHIFT INSERT	▞	Insert line	
{TAB}	ESC TAB	►	TAB key	
{CLR TAB}	ESC CTRL TAB	◧	Clear tab	
{SET TAB}	ESC SHIFT TAB	◨	Set tab stop	
{BELL}	ESC CTRL 2	◩	Ring buzzer	
{ESC}	ESC ESC	▛	ESCape key	

Graphics characters, such as CTRL-T, the ball character ● will appear as the "normal" letter enclosed in braces, e.g. {T}.

A series of identical control characters, such as 10 spaces, three cursor-lefts, or 20 CTRL-R's, will appear as {10 SPACES}, {3 LEFT}, {20 R}, etc. If the character in braces is in inverse video, that character or characters should be entered with the Atari logo key. For example, {▮} means to enter a reverse-field heart with CTRL-comma, {5▮} means to enter five inverse-video CTRL-U's.

Commodore PET/CBM/VIC/64

Generally, any PET/CBM/VIC/64 program listings will contain words within braces which spell out any special characters: {DOWN} would mean to press the cursor down key. {5 SPACES} would mean to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listings. For example, S would mean to type the S key while holding the shift key. If you find an underlined key enclosed in braces (e.g., {10 N}), you should type the key as many times as indicated (in our example, you would enter ten shifted N's). Some graphics characters are inaccessible from the keyboard on CBM Business models (32N, 8032).

For the VIC and 64, if a key is enclosed in special brackets, ⟨ ⟩, you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as indicated.

Rarely, you'll see in a Commodore 64 program a solitary letter of the alphabet enclosed in braces. These characters can be entered by holding down the CTRL key while typing the letter in the braces. For example, {A} would indicate that you should press CTRL-A.

About the *quote mode*: you know that you can move the cursor around the screen with the CRSR keys. Sometimes a programmer will want to move the cursor under program control. That's why you see all the {LEFT}'s, {HOME}'s, and {BLU}'s in our programs. The only way the computer can tell the difference between direct and programmed cursor control is the quote mode.

Once you press the quote (the double quote, SHIFT-2), you are in the quote mode. If you type something and then try to change it by moving the cursor left, you'll only get a bunch of reverse-video lines. These are the symbols for cursor left. The only editing key that isn't programmable is the DEL key; you can still use DEL to back up and edit the line. Once you type another quote, you are out of quote mode.

You also go into quote mode when you INSerT spaces into a line. In any case, the easiest way to get out of quote mode is to just press RETURN. You'll then be out of quote mode and you can cursor up to the mistyped line and fix it.

Use the following tables when entering special characters:

VIC And 64

When You Read:	Press:		See:	When You Read:	Press:		See:
{CLR}	SHIFT	CLR/HOME	▥	{GRN}	CTRL	6	⊹
{HOME}		CLR/HOME	▧	{BLU}	CTRL	7	⊹
{UP}	SHIFT	↕ CRSR ↕	▦	{YEL}	CTRL	8	⊓
{DOWN}		↕ CRSR ↕	▨	{F1}	f1		▦
{LEFT}	SHIFT	↔ CRSR ↔	‖	{F2}	f2		◥
{RIGHT}		↔ CRSR ↔	▐	{F3}	f3		▦
{RVS}	CTRL	9	�oⓇ	{F4}	f4		◣
{OFF}	CTRL	0	▦	{F5}	f5		▐
{BLK}	CTRL	1	■	{F6}	f6		▐
{WHT}	CTRL	2	▣	{F7}	f7		▐
{RED}	CTRL	3	⬩	{F8}	f8		■
{CYN}	CTRL	4	◸	←	←		←
{PUR}	CTRL	5	▩	↑	SHIFT	↕	⊓

All Commodore Machines

Clear Screen	{CLR}	Cursor Left	{LEFT}
Home Cursor	{HOME}	Insert Character	{INST}
Cursor Up	{UP}	Delete Character	{DEL}
Cursor Down	{DOWN}	Reverse Field On	{RVS}
Cursor Right	{RIGHT}	Reverse Field Off	{OFF}

Apple II / Apple II Plus

All programs are in Applesoft BASIC, unless otherwise stated. Control characters are printed as the "normal" character enclosed in braces, such as {D} for CTRL-D. Hold down CTRL while pressing the control key. You will not see the special character on the screen.

Texas Instruments 99/4

The only special characters used are in PRINT statements to indicate where two or more spaces should be left between words. For example, ENERGY {10 SPACES} MANAGEMENT means that ten spaces should be left between the words ENERGY and MANAGEMENT. Do not type in the braces or the words 10 SPACES. Enter all programs with the ALPHA LOCK on (in the down position). Release the ALPHA LOCK to enter lowercase text.

The Automatic Proofreader
For VIC, 64, And Atari

Charles Brannon, Program Editor

At last there's a way for your computer to help you check your typing. "The Automatic Proofreader" will make entering programs faster, easier, and more accurate.

The strong point of computers is that they excel at tedious, exacting tasks. So why not get your computer to check your typing for you?

With "The Automatic Proofreader" nestled in your VIC-20, Commodore 64, or Atari computer, every line you type in will be verified. It displays a special code, called a *checksum*, at the top of the screen. The checksum, either a number (VIC/64) or a pair of letters (Atari), corresponds to the line you've just typed. It represents every character in the line summed together. A matching code in the program listing lets you compare it to the checksum which the Proofreader displays. A glance is all it takes to confirm that you've typed the line correctly.

Entering The Automatic Proofreader

Commodore (VIC/64) owners should type in Program 1. Program 2 is for Atari users. Since the Proofreader is a machine language program, be especially diligent. Watch out for typing extra commas, or a letter O for a zero, and check every number carefully. If you make a mistake when typing in the DATA statements, you'll get the message "Error in DATA statements" when you RUN the program. Check your typing and try again.

When you've typed in The Automatic Proofreader, SAVE it to tape or disk at least twice *before running it for the first time*. If you mistype the Proofreader, it may cause a system crash when you first run it. By SAVEing a copy beforehand, you can reLOAD it and hunt for your error. Also, you'll want a backup copy of the Proofreader because you'll use it again and again—every time you enter a program from COMPUTE!.

When you RUN the Proofreader, the program will be POKEd safely into memory, then it will activate itself. If you ever need to reactivate it (RUN/STOP—RESTORE or SYSTEM RESET will disable it), just enter the command SYS 886 (VIC/64) or PRINT USR(1536) for the Atari.

Using The Proofreader

Now, let's see how it works. LIST the Proofreader program, move the cursor up to one of the lines, and press RETURN. If you've entered the Proofreader correctly, a checksum will appear in the top-left corner of your screen.

Try making a change in the line and hit RETURN. Notice that the checksum has changed. All VIC and 64 listings in COMPUTE! now have a number appended to the end of each line, for example, :rem 123. *Don't*

enter this statement. It is just for your information. The rem is used to make the number harmless if someone does type it in. It will, however, use up memory if you enter it, and it will cause the checksum displayed at the top of the screen to be different, even if you entered the rest of the line correctly.

The Atari checksum is found immediately to the left of each line number. This makes it impossible to type in the checksum accidentally, since a program line must start with a number.

Just type in each line (without the printed checksum), and check the checksum displayed at the top of the screen against the checksum in the listing. If they match, go on to the next line. If they don't, there's a mistake. You can correct the line immediately, instead of waiting to find the error when you RUN the program.

The Proofreader is not picky with spaces. It will not notice extra spaces or missing ones. This is for your convenience, since spacing is generally not important. Occasionally proper spacing is important, but the article describing the program will warn you to be careful in these cases.

Nobody's Perfect

Although the Proofreader is an important aid, there are a few things to watch out for. If you enter a line by using abbreviations for commands, the checksum will not match up. This is because the Proofreader is very literal: It looks at the individual letters in a line, not at tokens such as PRINT. There is a way to make the Proofreader check such a line. After entering the line, LIST it. This makes the computer spell out the abbreviations. Then move the cursor up to the line and press RETURN. It should now match the checksum. You can check whole groups of lines this way. Atari users should beware of using ? as an abbreviation for PRINT—they're not the same thing in the Proofreader's eyes.

The checksum is a sum of the ASCII values of the characters in a line. VIC and 64 owners may wonder why the numbers are so small, never exceeding 255. This is because the addition is done only in eight bits. A result over 255 will roll over past zero, like an odometer past 99999. On the Atari, the number is turned into two letters, both for increased convenience and to make the Proofreader shorter. For the curious, the letters correspond to the values of the left and right nybbles added to 33 (to offset them into the alphabet). This number is then stored directly into screen memory.

Due to the nature of a checksum, the Proofreader will not catch all errors. Since $1+3+5=3+1+5$, the Proofreader cannot catch errors of transposition. In fact, you could type in the line in any order, and the Proofreader wouldn't notice. Anytime the Proofreader

C-www.commodore.ca

seems to act strange, keep this in mind. Since the ASCII values of the number 18 (49 + 56) and 63 (54 + 51) both equal 105, these numbers are equal according to the Proofreader. There really is no simple way to catch these kinds of errors. Fortunately, the Proofreader will catch the majority of the typing mistakes most people make.

If you want the Proofreader out of your way, just press SYSTEM RESET or RUN/STOP—RESTORE. If you need it again, enter SYS 828 (VIC/64) or PRINT USR(1536) (Atari). You must disable the Proofreader before doing any tape operations on the VIC or 64.

Hidden Perils

The Proofreader's home in the VIC and 64 is not a very safe haven. Since the cassette buffer is wiped out during tape operations, you need to disable the Proofreader with RUN/STOP—RESTORE before you SAVE your program. This applies only to tape use. Disk users or Atari owners have nothing to worry about.

Not so for VIC and 64 owners with tape drives. What if you type in a program in several sittings? The next day, you come to your computer, LOAD and RUN the Proofreader, then try to LOAD the partially completed program so you can add to it. But since the Proofreader is trying to hide in the cassette buffer, it is wiped out!

What you need is a way to LOAD the Proofreader after you've LOADed the partial program. The problem is, a tape load to the buffer destroys what it's supposed to load.

After you've typed in and RUN the Proofreader, enter the following lines in direct mode (without line numbers) *exactly* as shown:

```
A$="PROOFREADER.T": B$="{10 SPACES}": FOR
  X = 1 TO 4: A$=A$+B$: NEXTX

FOR X = 886 TO 1018: A$=A$+CHR$(PEEK(X)):
  NEXTX

OPEN 1,1,1,A$:CLOSE1
```

After you enter the last line, you will be asked to press record and play on your cassette recorder. Put this program at the beginning of a new tape. This gives you a new way to load the Proofreader. Anytime you want to bring the Proofreader into memory without disturbing anything else, put the cassette in the tape drive, rewind, and enter:

```
OPEN1:CLOSE1
```

You can now start the Proofreader by typing SYS 886. To test this, PRINT PEEK(886) should return the number 173. If it does not, repeat the steps above, making sure that A$ ("PROOFREADER.T") contains 13 characters and that B$ contains 10 spaces.

You can now reload the Proofreader into memory whenever LOAD or SAVE destroys it, restoring your personal typing helper.

Incidentally, you can protect the cassette buffer on the Commodore 64 with POKE 178,251. This POKE should work on the VIC, but it has caused numerous problems, probably due to a bug in the VIC operating system. With this POKE, the 64 will not wipe out the cassette buffer during tape LOADs and SAVEs.

Program 1: VIC/64 Proofreader

```
100 PRINT"{CLR}PLEASE WAIT...":FORI=886TO
    1018:READA:CK=CK+A:POKEI,A:NEXT
110 IF CK<>17539 THEN PRINT"{DOWN}YOU MAD
    E AN ERROR":PRINT"IN DATA STATEMENTS.
    ":END
120 SYS886:PRINT"{CLR}{2 DOWN}PROOFREADER
     ACTIVATED.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
```

Program 2: Atari Proofreader

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:READ A:POKE I
    ,A:CK=CK+A:NEXT I
120 IF CK<>19072 THEN ? "Error in DA
    TA statements. Check typing":END

130 A=USR(1536)
140 ? :? "Automatic Proofreader now
    activated."
150 END
1536 DATA 104,160,0,185,26,3
1542 DATA 201,69,240,7,200,200
1548 DATA 192,34,208,243,96,200
1554 DATA 169,74,153,26,3,200
1560 DATA 169,6,153,26,3,162
1566 DATA 0,189,0,228,157,74
1572 DATA 6,232,224,16,208,245
1578 DATA 169,93,141,78,6,169
1584 DATA 6,141,79,6,24,173
1590 DATA 4,228,105,1,141,95
1596 DATA 6,173,5,228,105,0
1602 DATA 141,96,6,169,0,133
1608 DATA 203,96,247,238,125,241
1614 DATA 93,6,244,241,115,241
1620 DATA 124,241,76,205,238,0
1626 DATA 0,0,0,0,32,62
1632 DATA 246,8,201,155,240,13
1638 DATA 201,32,240,7,72,24
1644 DATA 101,203,133,203,104,40
1650 DATA 96,72,152,72,138,72
1656 DATA 160,0,169,128,145,88
1662 DATA 200,192,40,208,249,165
1668 DATA 203,74,74,74,74,24
1674 DATA 105,161,160,3,145,88
1680 DATA 165,203,41,15,24,105
1686 DATA 161,200,145,88,169,0
1692 DATA 133,203,104,170,104,168
1698 DATA 104,40,96
```

Missile Math

Garry S. Wick

Educational programs are usually designed to reward correct answers. "Missile Math" does this, but also gives extra points for speed. Here's an entertaining way for young students to learn their math. Versions for the Atari, Commodore VIC-20 and 64, the TI-99/4A, Apple, and IBM PC/PCjr.

"Missile Math" starts with an animated introduction screen, then a joystick-controlled menu appears on the screen. You have a choice of addition and subtraction or multiplication and division at a slow or fast speed. Using the joystick to select the menu item eliminates possible errors from incorrect typing. The joystick does not give any unacceptable entries when the fire button is pressed to start the game.

Different Difficulty Levels

After a short pause for the initialization of the player/missile figures, the player sees a screen that displays ENTERING LEVEL 1. As you advance from one level to the next, the math problems become increasingly difficult.

The problem appears at the top of the screen. On the bottom there are five possible answers, together with a missile gun which you control with the joystick. The object is to position the gun over the correct answer and launch a missile so that it destroys an enemy spaceship as it traverses the screen. On the upper left corner of the screen are spades representing the number of remaining guns. You begin with three guns. The score is displayed in the upper right corner of the screen.

A special kind of problem appears in random locations. For example, you could see 3+4=? or 3+?=7 or ?+4=7. Addition and subtraction are combined. It is similar for multiplication and division. The correct answer randomly appears in one of five possible locations, so the player never knows in advance where to position the gun. The values of the incorrect answers are chosen so that the correct answer is not obvious. This discourages guessing.

A Feisty UFO

The UFO moves across the screen at three different heights. The first height is near the top of the screen, and on the two successive flights, the UFO moves closer to the position of the gun. Of course, if the player destroys the UFO on the first pass, it does not appear at the lower altitudes. Instead a new problem appears, and the UFO starts again at the highest position.
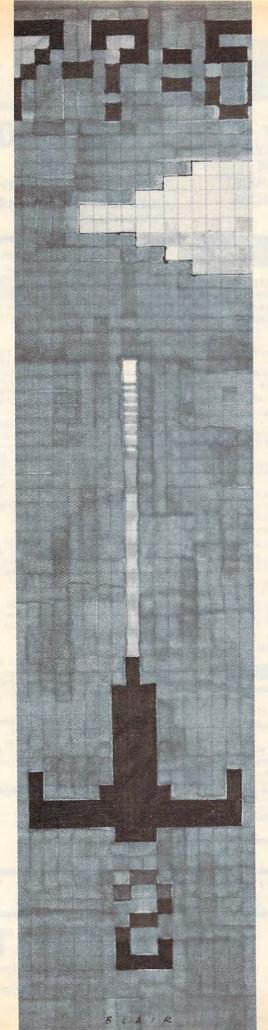
It is most difficult to destroy the UFO at the highest altitude because there is less time to calculate the correct answer and to fire the missile at the right time to hit the UFO. Destroying the UFO at the middle altitude is easier, and at the lowest altitude, it is easiest. Thus the player gets 25 points for a correct hit on the first pass, 10 points for the second pass, and 5 points for the third pass. When the gun is positioned over the correct answer and scores a hit, the UFO explodes with sound effects, disappears, and the number of points earned appears in its place. Then a new problem appears on the screen.

If you score a direct hit on the UFO, but have the gun over an incorrect answer, the UFO briefly changes colors, makes a funny sound and continues on its way. You must then try again on the next pass of the UFO. Three consecutive misses or incorrect answers and the UFO destroys the gun and one of the spades disappears. Sound effects accompany the disappearance of the spade and the correct answer blinks on and off as an encouraging message flies onto the screen. The game ends when all three guns have been destroyed.

To advance to the next level, you must score 50 points. If the present level is too easy, you can enter the next level by solving as few as two problems, receiving 25 points for each correct answer. If you only succeed in destroying the UFO on its third pass each time, then you will have to solve ten problems before moving to the next level. Thus you get more practice on problems that stretch your abilities. You can quickly pass by the problems that you find easy.

Bonus Points

You can earn the 50 points necessary to advance to higher levels with any combination of 5, 10, or 25 points, but you can earn bonus points for speed and accuracy. If the average score for the problems solved in a level is 25, the player receives 50 bonus points. The only way to get 50 bonus points is to score correct hits on the first two problems in a level during the first pass of

the UFO. If you average ten points or better per problem (but less than 25), you will earn 25 bonus points. There are no bonus points if you average less than ten points per problem. Bonus points are displayed with suitable fanfare.

There are a few features of Program 1 that require special mention. The joystick-controlled menu appears in Program 1 at lines 5000 through 5230. It uses screen memory locations to identify the choices available so that it is impossible to make an incorrect entry and cause an error.

Player/Missile Machine Language

The UFO is Atari Player 1 and the explosion character is Player 3. In order to rapidly exchange them when a correct hit has occurred, it is necessary that the two players always be at the same vertical position. (Then it is easy to POKE the UFO horizontally off the screen and to POKE the explosion at the former position of the UFO.)

BASIC was too slow to move both players vertically. A machine language program to move two players vertically lower on the screen is in lines 4000–4060. The parameters for the current location and length of the players are set by the subroutine at line 3500.

It is a little tricky and not at all obvious how to determine when the gun is over the correct answer. The playfield characters (the answers) and the players use different coordinate systems. It is necessary to establish an equation that maps one coordinate system to the other. The appropriate equations are:

$$X(P/M) = 4*X(Playfield) + 45$$
$$Y(P/M) = 4*Y(Playfield) + 17$$

where X and Y are the horizontal and vertical coordinates respectively. The X equation is used in line 2510 to determine whether the gun is over the correct answer, and the Y equation is used in line 3025 to position the number of points earned at the same height as the explosion of the UFO.

Program 1: Atari Missile Math

Refer to the "Automatic Proofreader" article before typing this program in.

```
JB 10 DIM M$(20),MIS$(13),BL$(40),A$
       (6),B$(6),C$(6),CH(5),KEEP$(21)
HL 15 GRAPHICS 2+16
BD 20 BL$=" ":BL$(20)=" ":BL$(2)=BL$
DB 30 MIS$="MISSILE MATH "
AN 60 FOR I=1 TO 15
BK 70 BL$(20-I,20)=MIS$
AI 80 SOUND 0,75-I,8,I
PA 90 POSITION 0,5:? #6;BL$:NEXT I
HM 100 RESTORE 110:FOR I=1 TO 35:REA
       D A:SOUND 0,A,10,8:FOR J=1 TO
       45:NEXT J:NEXT I
```

```
PE 110 DATA 81,81,81,60,60,60,60,60,
       60,60,60,60,81,81,81,60,68,72
       ,81,81,81,81,81,81,81
CP 111 DATA 0,81,81,81,81,60,60,60,6
       0,60,60,60
DJ 120 FOR I=0 TO 15
JG 130 BL$=BL$(2,20-I):SOUND 0,75-I,
       8,I
CD 140 POSITION 0,5:? #6;BL$:NEXT I:
       SOUND 0,0,0,0
DA 150 FOR I=1 TO 250:NEXT I
NL 160 GOSUB 4000
NN 170 GOSUB 5000
NI 175 ? CHR$(125):? :? :? "TO PLAY:
       ":? "POSITION GUN OVER CORREC
       T ANSWER":? "AND SHOOT THE UF
       O."
60 180 ? :? :? "BE CAREFUL NOT TO WA
       STE SHOTS."
OL 185 ? :? :? :? :? "PLEASE WAIT A
       MOMENT....."
NM 200 GOSUB 5500
CN 210 LVL=1:SC=0:NP=3:HIT=0:GUN=0:X
       1=165
MM 220 ? CHR$(125):POKE 752,1:POSITI
       ON 12,10:SETCOLOR 2,2,10:SETC
       OLOR 1,2,0:? "ENTERING LEVEL
       ";LVL
CB 225 PB=0:SCL=0
IF 226 RESTORE 110:FOR I=1 TO 35:REA
       D A:SOUND 0,A,10,8:FOR J=1 TO
       20:NEXT J:NEXT I:SOUND 0,0,0
       ,0
JE 230 A=INT(RND(0)*4+1)+4*(LVL-1)
IM 235 B=INT(RND(0)*4*LVL+1)
DE 240 IF MENU=1 THEN C=A+B:GOTO 255
ME 250 C=A*B
AJ 255 A$=STR$(A):B$=STR$(B):C$=STR$
       (C)
NC 260 RP=INT(RND(0)*4)+1
FO 270 IF RP=1 THEN ANS=A:A$=CHR$(63
       )
GC 280 IF RP=2 THEN ANS=B:B$=CHR$(63
       )
BK 290 IF RP=3 OR RP=4 THEN ANS=C:C$
       =CHR$(63)
GD 295 FOR I=1 TO 500:NEXT I:? CHR$(
       125):POSITION 13,5:SETCOLOR 1
       ,0,14:SETCOLOR 2,0,2
EJ 300 IF MENU=1 THEN ? A$;" + ";B$;
       " = ";C$:GOTO 320
AO 310 ? A$;" X ";B$;" = ";C$
EJ 320 FOR I=1 TO NP:POSITION 2*I,2:
       ? CHR$(123):NEXT I
NC 340 POSITION 25,2:? "SCORE ";SC:P
       B=PB+1
GC 345 IF ANS>90 THEN DEL=10:GOTO 36
       0
DA 346 IF ANS>20 THEN DEL=5:GOTO 360
AF 350 DEL=INT((ANS/10)+1)
MK 360 DT=INT(RND(0)*5)+1
FO 370 FOR I=1 TO 5:CH(I)=ANS+(I-DT)
       *DEL:POSITION 6*I,21:? CH(I):
       NEXT I
OH 375 GOSUB 3500
MG 380 POKE 53278,1:SHT=0
AM 390 FOR X0=200 TO 25 STEP -DX:POK
       E 53248,X0
OC 400 ST=STICK(0):X1=X1+4*((ST=7)+(
```

```
       X1<210)):X1=X1-4*((ST=11)+(X1
       >40)):POKE 53249,X1
BN 410 IF STRIG(0)=0 AND SHT=0 THEN
       GUN=1:POKE PMBASE+384+Y1,0:Y1
       =102:XM1=X1+7:POKE 53253,XM1:
       SHT=1
PC 415 IF GUN=1 THEN GOSUB 2000
AE 420 IF HIT=1 THEN HIT=0:GOTO 440
CE 430 NEXT X0:POKE 53253,250:SOUND
       0,0,0,0:GUN=0
60 435 GOTO 450
MH 440 IF SCL>=50 THEN LVL=LVL+1:POK
       E 53249,250:GOTO 220
JO 445 POKE 53249,250:GOTO 230
OD 450 GOSUB 3700
IC 460 IF Y0<100 THEN GOTO 380
AM 470 FOR X0=205 TO 30 STEP -DX:POK
       E 53248,X0
ED 480 IF PEEK(53260)<>2 THEN NEXT X
       0
OE 490 POKE 53248,250:POKE 53249,250
       :POKE 53250,X0
NP 500 GOSUB 2800
EM 530 POSITION 2*NP,2:? "*"
IB 540 FOR I=1 TO 30:SOUND 0,250-4*I
       ,10,8:NEXT I:SOUND 0,0,0,0
OB 550 POSITION 2*NP,2:? " ":FOR I=
       15 TO 0 STEP -1:SOUND 0,100,0
       ,I:NEXT I
HC 560 NP=NP-1
EC 565 FOR I=1 TO 10
FF 570 POSITION 6*DT,21:? "
       {4 SPACES}"
FK 575 K=INT(RND(0)*15)
KI 580 FOR J=40 TO 25 STEP -1:SOUND
       0,J-K,10,8:NEXT J
PF 590 POSITION 6*DT,21:? ANS
FN 600 K=INT(RND(0)*15):FOR J=40 TO
       25 STEP -1:SOUND 0,J-K,10,8:N
       EXT J:NEXT I:SOUND 0,0,0,0
FB 610 BL$=" ":BL$(38)=" ":BL$(2)=BL$
GE 620 KEEP$="KEEP TRYING "
HK 630 FOR I=1 TO 23:BL$(38-I,38)=KE
       EP$
GD 640 SOUND 0,90-I,8,INT(I/2)+3:POS
       ITION 0,9:? BL$:NEXT I
FF 650 BL$=" ":BL$(38)=" ":BL$(2)=BL
       $
ON 660 KEEP$="I KNOW YOU CAN DO IT "
ID 670 FOR I=1 TO 28:BL$(38-I,38)=KE
       EP$
FK 680 SOUND 0,100-I,8,INT(I/2):POSI
       TION 0,11:? BL$:NEXT I
IJ 690 RESTORE 110:FOR I=1 TO 35:REA
       D A:SOUND 0,A,10,8:FOR J=1 TO
       10:NEXT J:NEXT I:SOUND 0,0,0
       ,0
DG 700 IF NP=0 THEN GOTO 715
GN 710 SOUND 0,0,0,0:GOTO 230
DH 715 GRAPHICS 18:POSITION 5,2:? #6
       ;"SCORE ";SC
PI 720 POSITION 4,6:? #6;"PUSH TRIGG
       ER":POSITION 3,7:? #6;"TO PLA
       Y AGAIN"
KI 740 IF STRIG(0)=0 THEN SOUND 0,0,
       0,0:GOTO 170
II 750 RESTORE 110:FOR I=1 TO 35:REA
       D A:SOUND 0,A,10,8:FOR J=1 TO
       12:NEXT J:NEXT I:SOUND 0,0,0,0
```

```
HB 760 GOTO 740
HF 770 END
EB 1999 REM MISSILE MOVEMENT
FL 2000 YH1=Y1
HG 2010 Y1=Y1-5
PF 2020 IF Y1<5 THEN GUN=0:SOUND 0,0
        ,0,0:RETURN
GJ 2030 POKE PMBASE+384+YH1,0:POKE P
        MBASE+384+Y1,4:SOUND 0,Y1,8,
        10
JB 2035 IF PEEK(53257)<>3THEN RETURN
AN 2040 GOSUB 2500
MA 2050 POKE 77,0
AG 2100 POKE PMBASE+384+Y1,0:GUN=0:P
        OKE 53278,1:SOUND 0,0,0,0:RE
        TURN
MG 2499 REM CHECK COLLISION
KL 2500 REM
CL 2510 XCOR=4*6*DT+45
PH 2520 IF XM1<XCOR+15 AND XM1>XCOR-
        3 THEN GOSUB 3000:HIT=1:GOTO
         2540
BD 2530 GOSUB 2700
DM 2540 GUN=0:RETURN
HI 2699 REM UFO CLUNKS
CE 2700 POKE 704,14
EO 2710 FOR I=1 TO 25:SOUND 0,150+I,
        10,8:NEXT I
PN 2720 SOUND 0,0,0,0:POKE 704,24:PO
        KE 53278,1
KM 2730 RETURN
AJ 2799 REM EXPLOSION SOUND
BF 2800 FOR I=1 TO 2:N=INT(10*RND(0)
        +1)*10
DC 2810 FOR L=15 TO 0 STEP -1:SOUND
        0,N,0,L:SOUND 1,N+1,0,L:SOUN
        D 2,N+2,0,L:SOUND 3,N/2,0,L
PH 2820 FOR J=1 TO 10:NEXT J:NEXT L
CE 2830 POKE 53258,3:NEXT I
IM 2840 POKE 53258,1:POKE 53250,250
KP 2850 RETURN
GB 2999 REM DIRECT HIT
AF 3000 POKE 53248,250:POKE 53250,X0
        :POKE 53253,250
GD 3025 POSITION 6*DT,INT((Y0-14)/4)
JK 3030 IF Y0=40 THEN SC=SC+25:SCL=S
        CL+25:? "25"
IL 3040 IF Y0=60 THEN SC=SC+10:SCL=S
        CL+10:? "10"
AK 3050 IF Y0=80 THEN SC=SC+5:SCL=SC
        L+5:? "5"
BH 3055 GOSUB 2800
GB 3060 POSITION 30,2:? " ";SC:FOR I
        =1 TO 250:NEXT I
AJ 3070 IF SCL>=50 AND SCL/PB>20 THE
        N POSITION 13,12:HU=5:? "50
        BONUS POINTS":SC=SC+50:GOTO
        3090
EJ 3080 IF SCL>=50 AND SCL/PB>=10 TH
        EN POSITION 13,12:HU=4:? "25
        BONUS POINTS":SC=SC+25:GOTO
        3090
NA 3085 GOTO 3130
EA 3090 FOR I=1 TO 3
KF 3100 SETCOLOR 2,HU,4:SOUND 0,25,1
        0,8:FOR J=1 TO 50:NEXT J
AN 3110 SETCOLOR 2,0,2:SOUND 0,75,10
        ,8:FOR J=1 TO 50:NEXT J:NEXT
        I:SOUND 0,0,0,0:SOUND 1,0,0,0
```

```
IK 3120 POSITION 30,2:? " ";SC
KH 3130 RETURN
AG 3499 REM CHANGE UFO HEIGHT
PB 3500 FOR I=PMBASE+512+Y01 TO PMBA
        SE+518+Y01:GOSUB 3600:POKE I
        ,0:NEXT I
AF 3510 FOR I=PMBASE+768+Y01 TO PMBA
        SE+776+Y01:GOSUB 3600:POKE I
        ,0:NEXT I
PE 3520 Y0=40
GI 3530 RESTORE 3535:FOR I=PMBASE+51
        2+Y0 TO PMBASE+518+Y0:GOSUB
        3600:READ A:POKE I,A:NEXT I
FO 3535 DATA 24,60,255,255,255,60,24
HN 3540 RESTORE 3545:FOR I=PMBASE+76
        8+Y0 TO PMBASE+776+Y0:GOSUB
        3600:READ A:POKE I,A:NEXT I
GF 3545 DATA 16,68,82,137,66,161,34,
        68,16
FC 3546 Y01=Y0
IJ 3550 PM1=PMBASE+512+Y0:PM2=PMBASE
        +768+Y0
JK 3560 POKE 204,INT(PM1/256):POKE 2
        07,INT(PM2/256)
OK 3570 POKE 203,PM1-(PEEK(204)*256)
        -1:POKE 206,PM2-(PEEK(207)*2
        56)-1
GB 3580 POKE 205,7:POKE 208,9
LB 3590 RETURN
BH 3600 ST=STICK(0):X1=X1+4*((ST=7)+
        (X1<210)):X1=X1-4*((ST=11)+(
        X1>40)):POKE 53249,X1
KK 3610 RETURN
OB 3699 REM UFO DOWN
KG 3700 Y0=Y0+20
GO 3710 FOR I=1 TO 20
PM 3720 AB=USR(1536):BA=USR(1551):GO
        SUB 3600:NEXT I
FB 3725 Y01=Y0
KN 3730 RETURN
LP 3999 REM VERTICAL MOVEMENT DOWN
LB 4000 RESTORE 4050
HF 4010 FOR I=1536 TO 1565
AE 4020 READ A:POKE I,A:NEXT I
KH 4030 RETURN
NI 4050 DATA 104,164,205,177,203,200
        ,145,203,136,136,16,247,230,
        203,96
OF 4060 DATA 104,164,208,177,206,200
        ,145,206,136,136,16,247,230,
        206,96
PI 4999 REM MENU
AN 5000 GRAPHICS 0:? CHR$(125):POKE
        752,1
CO 5020 POSITION 11,2:? "MISSILE MAT
        H MENU"
MM 5030 ? :? "MOVE JOYSTICK AND PUSH
         TRIGGER":? "TO SELECT GAME"
BC 5060 M$="{Q}{W}{E}{DOWN}{3 LEFT}
        {A} {D}{DOWN}{3 LEFT}{Z}{X}
        {C}"
GE 5070 POSITION 8,11:? "ADDITION &
        {4 SPACES}MULTIPLICATION"
KD 5075 POSITION 8,12:? "SUBTRACTION
        {3 SPACES}& DIVISION"
EP 5080 ? :? :? "SLOW{4 SPACES}{UP}"
        ;M$
DN 5085 ? :? :? "FAST{4 SPACES}{UP}"
        ;M$
```

```
LK 5090 POSITION 25,14:? M$:POSITION
         25,18:? M$
HF 5100 SCR=PEEK(88)+256*PEEK(89):XC
        =611:MENU=1:DX=1
BN 5110 POKE SCR+XC,83:FOR DLY=1 TO
        50:NEXT DLY
OD 5120 POKE SCR+XC,0:FOR DLY=1 TO 5
        0:NEXT DLY
AB 5130 IF STRIG(0)=0 THEN GOTO 5230
LB 5135 ST=STICK(0)
DK 5140 IF ST=7 AND XC=611 THEN XC=6
        26:MENU=2
NB 5150 IF ST=13 AND XC=611 THEN XC=
        771:DX=3
NO 5160 IF ST=13 AND XC=626 THEN XC=
        786:DX=3
GH 5170 IF ST=11 AND XC=626 THEN XC=
        611:MENU=1
ND 5180 IF ST=14 AND XC=771 THEN XC=
        611:DX=1
EN 5190 IF ST=7 AND XC=771 THEN XC=7
        86:MENU=2
GP 5200 IF ST=11 AND XC=786 THEN XC=
        771:MENU=1
NJ 5210 IF ST=14 AND XC=786 THEN XC=
        626:DX=1
MJ 5220 GOTO 5110
KK 5230 RETURN
JP 5499 REM INITIALIZE PM GRAPHICS
JA 5500 X1=250:Y1=95:X0=250:Y0=40
OO 5510 J=PEEK(106)-8:POKE 54279,J:P
        MBASE=256*J
AE 5520 POKE 559,46:POKE 53277,3
LN 5530 POKE 53256,1:POKE 53257,1:PO
        KE 53258,1:POKE 53260,1
LK 5540 FOR I=PMBASE+384 TO PMBASE+8
        96:POKE I,0:NEXT I
PG 5570 RESTORE 5580:FOR I=PMBASE+64
        0+Y1 TO PMBASE+649+Y1:READ A
        :POKE I,A:NEXT I
AA 5580 DATA 8,8,8,8,8,28,127,127,54
        ,54
IC 5610 POKE 704,24:POKE 705,200:POK
        E 706,14
PH 5620 POKE 53248,X0:POKE 53249,X1:
        POKE 53250,X0
KO 5630 RETURN
```

Program 2:
VIC Missile Math, Machine Language

Refer to the "Automatic Proofreader" article before typing this
program in.

```
12288 :160,000,141,106,003,152,050
12294 :153,064,003,200,192,120,226
12300 :208,248,169,001,141,068,079
12306 :003,032,102,048,096,032,075
12312 :001,049,173,062,003,072,128
12318 :168,169,035,153,206,017,010
12324 :200,169,036,153,206,017,049
12330 :200,169,037,153,206,017,056
12336 :104,168,162,000,173,068,211
12342 :003,153,206,149,200,232,229
12348 :224,003,208,244,172,062,205
12354 :003,096,172,062,003,162,052
12360 :000,169,032,153,206,017,137
12366 :200,232,224,003,208,245,166
12372 :172,062,003,200,169,032,210
12378 :153,184,017,169,007,153,005
12384 :184,149,172,062,003,096,250
12390 :169,207,141,019,145,173,188
12396 :017,145,141,070,003,169,141
12402 :127,141,034,145,173,032,254
12408 :145,073,255,041,128,013,007
12414 :070,003,041,176,141,074,119
12420 :003,169,255,141,034,145,111
12426 :173,074,003,201,048,240,109
12432 :054,201,016,208,019,173,047
12438 :112,003,205,100,003,240,045
12444 :005,169,001,141,092,003,055
12450 :173,100,003,141,112,003,182
12456 :173,074,003,201,032,208,091
12462 :010,172,078,003,192,000,117
12468 :240,003,206,078,003,201,143
12474 :176,208,010,172,078,003,065
12480 :192,003,176,003,238,078,114
12486 :003,172,078,003,185,252,123
12492 :048,141,062,003,072,152,170
12498 :072,032,023,048,104,168,145
12504 :104,172,078,003,204,066,075
12510 :003,240,012,172,066,003,206
12516 :185,252,048,141,062,003,151
12522 :032,068,048,172,078,003,123
12528 :140,066,003,173,092,003,205
12534 :240,003,032,065,049,096,219
12540 :003,007,011,015,019,024,075
12546 :173,062,003,105,206,133,172
12552 :253,169,017,105,000,133,173
12558 :254,056,165,253,233,021,228
12564 :133,253,165,254,233,000,034
12570 :133,254,024,165,253,105,192
12576 :000,133,253,165,254,105,174
12582 :132,133,254,169,002,160,120
12588 :000,145,253,056,165,253,148
12594 :233,000,133,253,165,254,064
12600 :233,132,133,254,169,038,247
12606 :145,253,096,169,022,141,120
12612 :104,003,162,005,160,021,011
12618 :189,021,050,133,163,189,051
12624 :253,049,133,164,202,189,046
12630 :021,050,133,168,189,253,132
12636 :049,133,169,032,212,050,225
12642 :177,163,201,038,144,061,114
12648 :201,040,176,057,072,177,059
12654 :168,201,057,144,013,104,029
12660 :140,110,003,032,203,049,141
12666 :172,110,003,076,129,049,149
12672 :104,145,168,024,165,168,134
12678 :105,000,133,168,165,169,106
12684 :105,132,133,169,169,002,082
12690 :145,168,056,165,168,233,057
12696 :000,133,168,165,169,233,252
12702 :132,133,169,169,032,145,170
12708 :163,136,016,183,232,232,102
12714 :224,023,208,154,206,104,065
12720 :003,173,104,003,016,144,107
12726 :169,032,141,092,016,141,005
12732 :096,016,141,100,016,141,186
12738 :104,016,136,169,000,141,248
12744 :092,003,096,072,173,133,001
12750 :003,205,078,003,208,015,206
12756 :032,228,050,104,169,000,027
12762 :072,169,003,141,132,003,226
12768 :076,231,049,104,177,168,005
12774 :072,104,096,024,169,206,133
12780 :109,062,003,133,253,169,197
12786 :017,105,000,133,254,169,152
12792 :000,141,065,049,096,016,103
```

```
12798 :016,016,016,016,016,016,094
12804 :016,016,016,016,016,017,101
12810 :017,017,017,017,017,017,112
12816 :017,017,017,017,017,000,101
12822 :022,044,066,088,110,132,228
12828 :154,176,198,220,242,008,002
12834 :030,052,074,096,118,140,032
12840 :162,184,206,228,250,173,219
12846 :106,003,208,029,169,110,159
12852 :133,251,169,016,133,252,238
12858 :160,019,140,096,003,169,133
12864 :005,141,098,003,169,001,225
12870 :141,106,003,169,000,141,118
12876 :100,003,096,173,100,003,039
12882 :201,003,144,013,169,000,100
12888 :141,106,003,169,002,141,138
12894 :132,003,076,211,050,172,226
12900 :096,003,169,058,145,251,054
12906 :200,169,060,145,251,200,107
12912 :169,062,145,251,172,096,239
12918 :003,200,200,200,169,032,154
12924 :145,251,200,145,251,200,036
12930 :145,251,172,096,003,024,053
12936 :165,251,105,000,133,251,017
12942 :165,252,105,132,133,252,157
12948 :173,098,003,145,251,200,250
12954 :145,251,200,145,251,056,178
12960 :165,251,233,000,133,251,169
12966 :165,252,233,132,133,252,053
12972 :206,096,003,173,096,003,237
12978 :208,031,169,032,145,251,246
12984 :136,145,251,136,145,251,224
12990 :024,165,251,105,132,133,232
12996 :251,165,252,105,000,133,078
13002 :252,160,019,140,096,003,104
13008 :238,100,003,096,072,152,101
13014 :072,138,072,160,050,136,074
13020 :208,253,104,170,104,168,203
13026 :104,096,169,015,141,014,253
13032 :144,169,128,141,013,144,203
13038 :162,255,032,212,050,202,127
13044 :208,250,169,000,141,014,002
13050 :144,169,032,145,168,172,056
13056 :096,003,200,169,000,145,101
13062 :251,200,169,000,145,251,254
13068 :200,169,000,145,251,024,033
13074 :165,251,105,000,133,251,155
13080 :165,252,105,132,133,252,039
13086 :172,096,003,200,145,251,129
13092 :200,145,251,200,145,251,204
13098 :056,165,251,233,000,133,112
13104 :251,165,252,233,132,133,190
13110 :252,162,255,032,212,050,249
13116 :202,208,250,202,208,253,103
13122 :096,160,000,185,000,128,123
13128 :153,000,020,200,208,247,132
13134 :160,000,185,000,129,153,193
13140 :000,021,200,208,247,160,152
13146 :000,185,000,130,153,000,046
13152 :022,200,208,247,160,000,165
13158 :185,000,131,153,000,023,082
13164 :200,208,247,160,007,185,091
13170 :172,051,153,000,020,136,134
13176 :016,247,160,031,185,180,171
13182 :051,153,024,021,136,016,015
13188 :247,160,007,185,212,051,226
13194 :153,208,021,136,016,247,151
13200 :160,007,185,220,051,153,152
13206 :224,021,136,016,247,160,186
```

```
13212 :007,185,228,051,153,240,252
13218 :021,136,016,247,169,205,188
13224 :141,005,144,096,146,073,005
13230 :164,073,018,164,146,000,227
13236 :000,000,001,003,031,063,022
13242 :112,240,024,255,255,255,047
13248 :255,255,000,000,000,000,190
13254 :128,192,248,252,014,015,023
13260 :000,033,030,030,030,033,104
13266 :000,000,000,000,000,007,217
13272 :031,063,127,127,000,000,052
13278 :000,255,255,255,255,255,217
13284 :001,007,031,255,255,255,008
13290 :255,255,013,013,013,013,028
```

Program 3: VIC Missile Math, BASIC

Refer to the "Automatic Proofreader" article before typing this
program in.

```
5 POKE 36879,8:POKE56,48:CLR:SYS65017
                                       :rem 156
10 PRINT"{CLR}{2 DOWN}{WHT}":A$="MISSILE
   {SPACE}MATH":GOSUB900            :rem 159
15 POKE36869,192                    :rem 109
20 PRINT"{3 DOWN}{5 RIGHT}{RED}DO YOU WAN
   T TO:"                           :rem 186
30 PRINT"{2 DOWN}{RIGHT}{CYN}ADD&
   {7 SPACES}{WHT}MULTIPLY"         :rem 134
40 PRINT"{CYN}{DOWN}{RIGHT}SUBTRACT
   {3 SPACES}{WHT}& DIVIDE"         :rem 74
50 PRINT"{2 DOWN}{GRN}SLOW":PRINT"
   {2 DOWN}{CYN}FAST{YEL}":G=15:H=5
                                    :rem 107
55 H=5 :G=15:GF=20                  :rem 174
60 V1=37139:REM DDR FOR SWITCH UPDNLFTFIR
   E                                :rem 37
61 R1 =37137:REM PORTAB2=UP;B3=DN;B4=LFT:
   B5=FIRE                          :rem 141
62 V2=37154:REM DDR FOR RIGHT       :rem 142
63 R2= 37152:REM PORTB;B7= RIGHT    :rem 64
64 POKE37139,195:J1=PEEK(37137):POKE37139
   ,128                            :rem 15
65 POKE37154,127:J2=PEEK(37152)AND128:POK
   E37154,255                      :rem 114
66 B=J1ORJ2                         :rem 131
70 IF B= 156THEN 90                 :rem 169
80 IFB=172THENH=5                   :rem 249
81 IFB=60THENH=14                   :rem 246
82 IFB=184THENG=15:GF=20            :rem 148
83 IFB=180THENG=18:GF=10            :rem 147
84 POKE781,G:POKE782,H:POKE783,0:SYS65520
   :PRINT"Q"                       :rem 170
85 IF OG<>G OR OH<>H THENPOKE781,OG:POKE7
   82,OH:POKE783,0:SYS65520:PRINT" "
                                    :rem 135
86 OG=G:OH=H:GOTO60                 :rem 183
90 L=1:SC=0:S=3                     :rem 93
100 PRINT"{CLR}{3 DOWN}{2 RIGHT}{WHT}POSI
    TION GUN OVER{7 SPACES}CORRECT ANSWER
    "                               :rem 228
105 BN=0:BO=0:B=0:SH=0:SYS13123     :rem 41
110 PRINT"{4 DOWN}{4 RIGHT}AND SHOOT UFO"
                                    :rem 101
120 PRINT"{3 DOWN}{4 RIGHT}{WHT}BE CAREFU
    L NOT{8 SPACES}TO WASTE SHOTS":rem 66
130 PRINT"{2 DOWN}{3 RIGHT}{CYN}ENTERING
    {SPACE}LEVEL";L                 :rem 216
135 GOSUB 1000                      :rem 218
140 FORI=1TO5000:NEXT               :rem 20
141 PRINT"{CLR}{11 SPACES}{WHT}SCORE{RED}
    "SC"{HOME}{BLU}";                :rem 222
```

C-www.commodore.ca

```
142 FORI=1TOS:PRINT"{RIGHT}A";:NEXT
                                        :rem 200
150 A=INT(RND(0)*4+1)+4*(L-1)           :rem 243
160 B=INT(RND(0)*4*L+1)                 :rem 231
170 IFH=5THENC=A+B:GOTO180              :rem 74
175 C=A*B                               :rem 202
180 A$=STR$(A):B$=STR$(B):C$=STR$(C)
                                        :rem 6
190 RP=INT(RND(0)*4)+1                  :rem 212
200 IFRP=1THENANS=A:A$=" ?"             :rem 31
210 IFRP=2THENANS=B:B$=" ?"             :rem 35
220 IFRP>2THENANS=C:C$=" ?"             :rem 39
230 PRINT"{HOME}{2 DOWN}{4 RIGHT}";:IFH=5
    THENPRINTA$" +"B$" ="C$:GOTO250
                                        :rem 181
240 PRINTA$" X"B$" ="C$                 :rem 114
250 IFANS>90THENDEL=10:GOTO270          :rem 93
260 IFANS>20THENDEL=5:GOTO270           :rem 43
265 DEL=INT((ANS/10)+1)                 :rem 10
270 DT=INT(RND(0)*4)+1                  :rem 201
275 PRINT"{HOME}{21 DOWN}"              :rem 231
277 SYS12288                            :rem 164
280 FORI=1TO4:CH(I)=ANS+(I-DT)*DEL:PRINTT
    AB((I*4)-2)STR$(CH(I));"{2 LEFT}";:NE
    XT                                  :rem 92
285 FORI=1TO4:IFCH(I)=ANSTHENPOKE901,I-1
                                        :rem 236
286 NEXT:POKE880,255                    :rem 176
290 W=PEEK(868):SYS 12307:FORQ=1TOGF*10:N
    EXT:IF PEEK(900)THEN300             :rem 47
293 SYS12845                            :rem 161
295 GOTO 290                            :rem 116
300 X=PEEK(900):SH=SH+1                 :rem 64
310 IFX=3THEN500                        :rem 175
320 IFX=2THEN600                        :rem 176
500 IFW=0THENB=25                       :rem 253
510 IFW=1THENB=10                       :rem 249
520 IFW=2THENB=5                        :rem 207
525 BN=BN+B                             :rem 102
526 SC=SC+B:PRINT"{HOME}{8 DOWN}
    {10 RIGHT}"B                        :rem 125
530 IFBN>=50THENL=L+1:GOTO700           :rem 1
550 GOTO 140                            :rem 104
600 S=S-1:IFS=0THEN800                  :rem 39
601 PRINT"{HOME}{BLU}{8 DOWN}":A$="KEEP T
    RYING":GOSUB 900                    :rem 151
602 PRINT"{2 DOWN}":A$="I KNOW YOU CAN DO
     IT":GOSUB 900                      :rem 133
603 PRINT"{2 DOWN}":A$="THE ANSWER WAS":G
    OSUB900:PRINT"{PUR}{DOWN}":A$=STR$(AN
    S):GOSUB900                         :rem 210
610 FORI=1TO1000:NEXT:GOTO140           :rem 26
700 BO=INT(BN/SH+.5)                    :rem 137
710 IFBO>=25THENBO=50:GOTO750           :rem 11
720 IFBO>=10THENBO=25:GOTO750           :rem 8
730 GOTO 760                            :rem 112
750 PRINT"{HOME}{14 DOWN}{5 RIGHT}B{CYN}O
    {GRN}N{YEL}U{BLU}S"BO:SC=SC+BO
                                        :rem 240
760 FORI=1TO5000:NEXT:GOTO 100          :rem 32
800 PRINT"{CLR}{3 DOWN}{6 RIGHT}{CYN}SCOR
    E ";SC                              :rem 201
805 PRINT"{3 DOWN}{6 RIGHT}{BLU}GAME OVER
    "                                   :rem 196
820 PRINT"{5 DOWN}{4 RIGHT}{CYN}PRESS ANY
     KEY {DOWN}{8 SPACES}TO PLAY AGAIN"
                                        :rem 123
830 POKE198,0                           :rem 200
835 WAIT 198,1                          :rem 212
840 GOTO5                               :rem 10
```

```
900 X=LEN(A$):A$=A$+"{2 SPACES}"    :rem 115
910 FORI=1TOX+10-INT(X/2+.5):POKE211,21-I
    :PRINTMID$(A$,1,I);                :rem 249
915 FOR T=1TO60:NEXTT:NEXTI           :rem 224
920 RETURN                            :rem 123
999 END                               :rem 130
1000 POKE36878,15:S2=36876:POKE36877,127
                                      :rem 218
1005 RESTORE                          :rem 234
1010 READ P,LE                        :rem 235
1020 IF P=-5THENPOKES2,0:RETURN       :rem 154
1025 POKES2,P                         :rem 248
1030 FOR T= 1 TO LE*2:NEXT            :rem 118
1040 GOTO1010                         :rem 192
1050 DATA 173,50,127,50,173,50,192,200,12
     7,100,192,200,127,100             :rem 205
1060 DATA 192,200,192,50,189,50,181,50,17
     3,100,173,50,192,300,-5,-5        :rem 201
```

Programmer's Notes: VIC Version

The VIC-20 version of "Missile Math" requires at least an 8K expander. To make it easier to enter this version, the machine language portion of the program has been listed in MLX format. To enter Missile Math, you must enter Program 2 using the VIC MLX Program elsewhere in this issue. The starting address is 12288 and the ending address is 13295. After you have entered and saved Program 2, enter Program 3 as you would a normal BASIC program and save it to tape or disk.

To run Missile Math, you must first enter these POKEs.
POKE 43,1:POKE 44,24:POKE 641,0:POKE 642,24:POKE 6144,0
Type NEW and load Program 2, bypassing the normal relocator by adding ,1 to the LOAD command. Here's how your LOAD command should look:

From tape:

LOAD "filename" ,1,1

From disk:

LOAD "filename" ,8,1

Now type NEW, then load and run Program 3.

Program 4: 64 Missile Math

Refer to the "Automatic Proofreader" article before typing this program in.

```
0 GOSUB 1000                          :rem 113
1 V=54272:FORI=0TO24:POKEV+I,0:NEXT
                                      :rem 176
5 POKE 53280,0:POKE 53281,0           :rem 138
10 PRINT"{CLR}{4 DOWN}{BLU}":A$="MISSILE
   {SPACE}MATH":GOSUB900:GOSUB 950:rem 51
15 PRINT"{DOWN}{9 RIGHT}USE JOYSTICK IN P
   ORT #2"                            :rem 219
20 PRINT"{3 DOWN}{13 RIGHT}{RED}PLEASE SE
```

```
      LECT:"                        :rem 175
30 PRINT"{2 DOWN}{5 RIGHT}{CYN}ADDITION &
   {7 SPACES}{7}MULTIPLICATION"  :rem 192
40 PRINT"{5 RIGHT}{CYN}SUBTRACTION
   {9 SPACES}{7}& DIVISION"       :rem 216
50 PRINT"{2 DOWN}{GRN}SLOW":PRINT"
   {2 DOWN}{6}FAST{YEL}":G=18:H=9:POKE830
   ,H:POKE831,G                   :rem 92
60 X=PEEK(56320):J=XAND15:B=XAND16:rem 14
70 IFB=0THEN90                    :rem 61
80 IFJ=11THENH=9                  :rem 205
81 IFJ=7THENH=29                  :rem 213
82 IFJ=14THENG=18                 :rem 1
83 IFJ=13THENG=21                 :rem 251
84 POKE828,H:POKE829,G:SYS49152   :rem 110
85 GOTO60                         :rem 12
90 L=1:SC=0:S=3                   :rem 93
100 POKE 53269,0:PRINT"{CLR}{8 DOWN}
    {4 RIGHT}{8}POSITION GUN OVER CORRECT
     ANSWER"                      :rem 215
105 BN=0:BO=0:B=0:SH=0            :rem 246
110 PRINT"{2 DOWN}{14 RIGHT}AND SHOOT UFO
    "                            :rem 101
120 PRINT"{3 DOWN}{5 RIGHT}{3}BE CAREFUL
    {SPACE}NOT TO WASTE SHOTS"    :rem 240
130 PRINT"{2 DOWN}{12 RIGHT}{PUR}ENTERING
     LEVEL";L                     :rem 218
135 GOSUB 950                     :rem 183
140 FORI=1TO3000:NEXT             :rem 18
141 PRINT"{CLR}{30 SPACES}{GRN}SCORE{RED}
    "SC"{HOME}{BLU}";             :rem 247
142 FORI=1TOS:PRINT"{RIGHT}A";:NEXT
                                 :rem 200
150 A=INT(RND(0)*4+1)+4*(L-1)    :rem 243
160 B=INT(RND(0)*4*L+1)          :rem 231
170 IFH=9THENC=A+B:GOTO180       :rem 78
175 C=A*B                        :rem 202
180 A$=STR$(A):B$=STR$(B):C$=STR$(C)
                                 :rem 6
190 RP=INT(RND(0)*4)+1           :rem 212
200 IFRP=1THENANS=A:A$=" ?"      :rem 31
210 IFRP=2THENANS=B:B$=" ?"      :rem 35
220 IFRP>2THENANS=C:C$=" ?"      :rem 39
230 PRINT"{HOME}{4 DOWN}{15 RIGHT}{PUR}";
    :IFH=9THENPRINTA$" +"B$" ="C$:GOTO250
                                 :rem 182
240 PRINTA$" X"B$" ="C$          :rem 114
250 IFANS>90THENDEL=10:GOTO270   :rem 93
260 IFANS>20THENDEL=5:GOTO270    :rem 43
265 DEL=INT((ANS/10)+1)          :rem 10
270 DT=INT(RND(0)*5)+1           :rem 202
275 PRINT"{HOME}{21 DOWN}{YEL}"  :rem 133
280 FORI=1TO5:CH(I)=ANS+(I-DT)*DEL:PRINTT
    AB(I*5)CH(I);:NEXT           :rem 135
285 FORI=1TO5:IFCH(I)=ANSTHENPOKE901,I-1
                                 :rem 237
286 NEXT                         :rem 223
290 SYS49195                     :rem 166
300 X=PEEK(900):SH=SH+1          :rem 64
310 IFX=2THEN500                 :rem 174
320 IFX=1THEN600                 :rem 175
500 W=PEEK(833):IFW=0THENB=25    :rem 223
510 IFW=1THENB=10                :rem 249
520 IFW=2THENB=5                 :rem 207
525 BN=BN+B                      :rem 102
526 SC=SC+B:PRINT"{HOME}{18 DOWN}
    {17 RIGHT}{GRN}"B             :rem 16
530 IFBN>=50THENL=L+1:GOTO700    :rem 1
550 GOTO 140                     :rem 104
600 S=S-1                        :rem 215

601 PRINT"{HOME}{7}{9 DOWN}":A$="KEEP TRY
    ING":GOSUB 900               :rem 35
602 PRINT"{3 DOWN}":A$="I KNOW YOU CAN DO
     IT":GOSUB 900               :rem 150
603 PRINT"{2 DOWN}":A$="THE ANSWER WAS:":
    GOSUB900:PRINT"{PUR}":A$=STR$(ANS):GO
    SUB900                       :rem 251
610 GOSUB950:IFS=0THEN800        :rem 5
620 GOTO140                      :rem 102
700 BO=INT(BN/SH+.5)             :rem 137
710 IFBO>=25THENBO=50:GOTO750    :rem 11
720 IFBO>=10THENBO=25:GOTO750    :rem 8
730 GOTO 760                     :rem 112
750 PRINT"{HOME}{14 DOWN}{14 RIGHT}{3}B
    {CYN}O{GRN}N{YEL}U{BLU}S{RED}:{WHT}"B
    O:SC=SC+BO                   :rem 230
760 FORI=1TO5000:NEXT:GOTO 100   :rem 32
800 POKE53269,0:PRINT"{CLR}{3 DOWN}
    {15 RIGHT}{CYN}SCORE:{PUR}";SC
                                 :rem 114
805 PRINT"{7 DOWN}{18 RIGHT}{BLU}GAME"
                                 :rem 40
810 PRINT"{3 DOWN}{18 RIGHT}OVER":rem 227
820 PRINT"{5 DOWN}{6 RIGHT}{GRN}PRESS
    {YEL}ANY KEY{GRN} TO PLAY AGAIN"
                                 :rem 223
830 POKE198,0                    :rem 200
835 WAIT 198,1                   :rem 212
840 GOTO5                        :rem 10
900 X=LEN(A$):A$=A$+"{24 SPACES}":rem 115
910 FORI=1TOX+19-INT(X/2+.5):POKE211,39-I
    :PRINTMID$(A$,1,I);:NEXT I    :rem 205
920 RETURN                       :rem 123
950 V=54272:POKEV+24,15:POKEV+5,0:POKEV+6
    ,240                         :rem 116
955 POKEV+1,25:POKEV,30:POKEV+4,33:FORI=1
    TO300:NEXT:POKEV+4,32        :rem 75
960 POKEV+1,33:POKEV,135:POKEV+4,33:FORI=
    1TO900:NEXT:POKEV+4,32       :rem 130
970 POKEV+1,25:POKEV,30:POKEV+4,33:FORI=1
    TO400:NEXT:POKEV+4,32        :rem 73
980 POKEV+1,33:POKEV,135:POKEV+4,33:FORI=
    1TO100:NEXT:POKEV+4,32       :rem 124
990 POKEV+1,29:POKEV,223:POKEV+4,33:FORI=
    1TO100:NEXT:POKEV+4,32       :rem 128
992 POKEV+1,28:POKEV,49:POKEV+4,33:FORI=1
    TO100:NEXT:POKEV+4,32        :rem 87
994 POKEV+1,25:POKEV,30:POKEV+4,33:FORI=1
    TO700:NEXT:POKEV+4,32        :rem 82
996 FORI=1TO100:NEXT             :rem 243
998 POKEV+1,25:POKEV,30:POKEV+4,33:FORI=1
    TO360:NEXT:POKEV+4,32        :rem 88
999 POKEV+1,33:POKEV,135:POKEV+4,33:FORI=
    1TO600:NEXT:POKEV+4,32:RETURN:rem 165
1000 PRINT"{CLR}{10 DOWN}{8 RIGHT}LOADING
      MACHINE LANGUAGE":I=49152:CK=0
                                 :rem 232
1010 READ A:IF A=256 THEN 1030   :rem 246
1020 POKE I,A:CK=CK+A:I=I+1:GOTO 1010
                                 :rem 65
1030 IFCK<>80338THENPRINT"ERROR IN DATA":
     STOP                        :rem 28
1040 RETURN                      :rem 165
1050 DATA 174,63,3,172,62,3      :rem 41
1060 DATA 24,32,240,255,169,32   :rem 191
1070 DATA 32,210,255,174,61,3    :rem 136
1080 DATA 172,60,3,24,32,240     :rem 83
1090 DATA 255,169,209,32,210,255 :rem 43
1100 DATA 173,61,3,141,63,3      :rem 31
1110 DATA 173,60,3,141,62,3      :rem 30
```

C=www.commodore.ca

Notes On The Commodore 64, TI, Apple, And IBM PC/PCjr Versions

Instructions for all these versions are included within the programs themselves. The 64 version (Program 4) requires a joystick in port 2. The TI-99/4A version (Program 5) requires Extended BASIC and may be played with either a joystick or the keyboard. The Apple version (Program 6) requires a paddle controller. The IBM PC/PCjr version (Program 7) is controlled from the keyboard.

```
1120 DATA 96,32,106,194,169,2      :rem 149
1130 DATA 141,64,3,169,0,141       :rem 84
1140 DATA 65,3,169,1,141,66        :rem 45
1150 DATA 3,169,100,141,67,3       :rem 87
1160 DATA 169,250,141,248,7,169     :rem 0
1170 DATA 251,141,249,7,169,252    :rem 252
1180 DATA 141,250,7,169,7,141      :rem 145
1190 DATA 21,208,169,6,141,39      :rem 150
1200 DATA 208,169,8,141,40,208     :rem 191
1210 DATA 169,15,141,41,208,169    :rem 245
1220 DATA 0,141,23,208,169,2       :rem 82
1230 DATA 141,29,208,169,255,141   :rem 41
1240 DATA 27,208,141,68,3,169      :rem 151
1250 DATA 0,141,70,3,173,30        :rem 28
1260 DATA 208,169,0,162,24,157     :rem 197
1270 DATA 0,212,202,224,255,208    :rem 231
1280 DATA 248,169,15,141,24,212    :rem 246
1290 DATA 173,61,3,201,21,208      :rem 135
1300 DATA 8,169,15,141,77,3        :rem 47
1310 DATA 76,164,192,169,22,141    :rem 249
1320 DATA 77,3,32,227,192,32       :rem 94
1330 DATA 54,193,32,160,193,32     :rem 193
1340 DATA 230,193,32,218,193,173   :rem 40
1350 DATA 68,3,208,236,169,17      :rem 157
1360 DATA 141,5,212,141,6,212      :rem 129
1370 DATA 169,129,141,4,212,169    :rem 252
1380 DATA 1,141,1,212,169,30       :rem 82
1390 DATA 141,76,3,32,218,193      :rem 150
1400 DATA 206,76,3,208,248,169     :rem 203
1410 DATA 0,170,157,0,212,232      :rem 125
1420 DATA 224,25,208,248,96,174    :rem 255
1430 DATA 64,3,189,82,194,141      :rem 155
1440 DATA 0,208,169,200,141,1      :rem 130
1450 DATA 208,173,67,3,141,2        :rem 93
1460 DATA 208,173,66,3,10,141      :rem 140
1470 DATA 16,208,174,65,3,189      :rem 159
1480 DATA 102,194,141,3,208,173    :rem 242
1490 DATA 70,3,201,1,240,14        :rem 29
1500 DATA 174,64,3,189,82,194      :rem 159
1510 DATA 141,4,208,169,200,141    :rem 233
1520 DATA 5,208,173,76,3,240        :rem 94
1530 DATA 20,206,76,3,173,76        :rem 97
1540 DATA 3,10,10,141,1,212        :rem 17
1550 DATA 201,0,208,5,169,128      :rem 142
1560 DATA 141,4,212,96,173,67      :rem 152
1570 DATA 3,208,3,206,66,3         :rem 250
1580 DATA 206,67,3,173,66,3         :rem 54
1590 DATA 24,109,67,3,201,0        :rem 40
1600 DATA 208,23,238,65,3,169      :rem 151
1610 DATA 1,141,66,3,169,80        :rem 44
1620 DATA 141,67,3,169,0,141       :rem 91
1630 DATA 70,3,169,8,141,40        :rem 44
1640 DATA 208,173,0,220,141,69     :rem 191
1650 DATA 3,41,8,240,18,173        :rem 44
1660 DATA 69,3,41,4,208,21         :rem 251
1670 DATA 173,64,3,240,16,206      :rem 145
1680 DATA 64,3,76,137,193,173      :rem 162
1690 DATA 64,3,201,16,240,3         :rem 38
1700 DATA 238,64,3,173,70,3        :rem 45
1710 DATA 201,1,208,15,206,5       :rem 80
1720 DATA 208,173,5,208,201,30     :rem 186
1730 DATA 208,5,169,2,141,70       :rem 95
1740 DATA 3,96,173,69,3,41          :rem 6
1750 DATA 16,201,0,208,36,173      :rem 139
1760 DATA 70,3,208,31,169,1        :rem 45
1770 DATA 141,70,3,173,64,3        :rem 45
1780 DATA 74,74,141,71,3,169       :rem 109
1790 DATA 17,141,5,212,141,6       :rem 91
1800 DATA 212,169,129,141,4,212    :rem 239
1810 DATA 169,10,141,76,3,173      :rem 146
1820 DATA 70,3,201,1,240,5         :rem 234
1830 DATA 169,5,141,71,3,96        :rem 55
1840 DATA 174,77,3,160,0,136       :rem 97
1850 DATA 208,253,202,208,250,96   :rem 44
1860 DATA 173,30,208,141,72,3      :rem 143
1870 DATA 41,1,141,73,3,173        :rem 42
1880 DATA 72,3,41,2,74,141         :rem 251
1890 DATA 74,3,173,72,3,41          :rem 2
1900 DATA 4,74,74,141,75,3         :rem 255
1910 DATA 173,73,3,45,74,3          :rem 0
1920 DATA 201,1,208,3,76,31        :rem 36
1930 DATA 194,173,74,3,45,75       :rem 111
1940 DATA 3,201,1,208,3,76         :rem 245
1950 DATA 52,194,96,169,253,141     :rem 9
1960 DATA 248,7,169,1,141,21       :rem 101
1970 DATA 208,169,1,141,132,3      :rem 145
1980 DATA 169,0,141,68,3,96        :rem 62
1990 DATA 173,133,3,205,71,3       :rem 96
2000 DATA 208,16,169,253,141,249   :rem 40
2010 DATA 7,169,2,141,132,3        :rem 33
2020 DATA 169,0,141,68,3,96        :rem 48
2030 DATA 169,1,141,40,208,96      :rem 143
2040 DATA 70,70,70,70,110,110      :rem 124
2050 DATA 110,110,150,150,150,150  :rem 57
2060 DATA 190,190,190,190,230,230  :rem 80
2070 DATA 230,230,100,133,167,200  :rem 65
2080 DATA 162,0,189,118,194,157     :rem 0
2090 DATA 128,62,232,208,247,96     :rem 1
2100 DATA 0,0,0,0,0,0              :rem 217
2110 DATA 0,24,0,0,24,0            :rem 70
2120 DATA 0,24,0,0,24,0            :rem 71
2130 DATA 0,24,0,0,24,0            :rem 72
2140 DATA 0,24,0,0,24,0            :rem 73
2150 DATA 0,60,0,0,60,0            :rem 74
2160 DATA 0,60,0,0,60,0            :rem 75
2170 DATA 1,255,128,15,255,240     :rem 192
2180 DATA 127,255,254,127,0,254    :rem 247
2190 DATA 112,0,14,96,0,6          :rem 192
2200 DATA 96,0,6,192,0,0           :rem 139
2210 DATA 0,0,0,0,0,0              :rem 219
2220 DATA 0,0,0,0,0,0              :rem 220
2230 DATA 0,0,255,0,1,255          :rem 182
2240 DATA 128,7,255,224,31,255     :rem 196
2250 DATA 248,255,255,255,255,255  :rem 105
2260 DATA 255,255,255,255,255,255  :rem 104
2270 DATA 255,31,255,248,7,255     :rem 206
2280 DATA 224,1,255,128,0,255      :rem 142
2290 DATA 0,0,0,0,0,0              :rem 227
2300 DATA 0,0,0,0,0,0              :rem 219
2310 DATA 0,164,0,24,0,0           :rem 125
```

```
2320 DATA 24,0,0,0,0,0              :rem 19
2330 DATA 0,0,0,0,0,0               :rem 222
2340 DATA 0,0,0,0,0,0               :rem 223
2350 DATA 0,0,0,0,0,0               :rem 224
2360 DATA 0,0,0,0,0,0               :rem 225
2370 DATA 0,0,0,0,0,0               :rem 226
2380 DATA 0,0,0,0,0,0               :rem 227
2390 DATA 0,0,0,0,0,0               :rem 228
2400 DATA 0,0,0,0,0,0               :rem 220
2410 DATA 0,0,0,0,0,247             :rem 74
2420 DATA 0,0,0,0,32,0              :rem 19
2430 DATA 40,128,64,0,136,0         :rem 34
2440 DATA 2,2,0,130,33,128          :rem 233
2450 DATA 2,18,196,16,168,0         :rem 50
2460 DATA 17,11,0,1,169,0           :rem 189
2470 DATA 1,45,72,17,146,8          :rem 1
2480 DATA 0,136,144,0,8,8           :rem 199
2490 DATA 8,120,0,2,24,128          :rem 243
2500 DATA 16,8,16,0,0,32            :rem 136
2510 DATA 2,3,0,1,32,0              :rem 25
2520 DATA 0,0,0,193,256             :rem 93
```

Program 5: TI-99/4A Missile Math

```
90 REM EXTENDED BASIC REQUIRED
100 DIM NOTE(14),DUR(14):: GOSUB 99
    0 :: GOTO 150
110 CALL KEY(0,K,ST):: CALL KEY(1,K
    K,S):: IF ST=0 THEN CALL JOYST(
    1,X,Y)ELSE X=(K=83)-(K=68):: Y=
    (K=69)-(K=88)
120 RETURN
130 FOR I=1 TO L :: B$=SEG$(E$,1,I)
    :: DISPLAY AT(Y,X-I):B$ :: CALL
    SOUND(-10,-6,4)::NEXT I ::RETURN
140 FOR I=1 TO 14 :: CALL SOUND(DUR
    (I),NOTE(I),3):: NEXT I :: RETU
    RN
150 E$="MISSILE MATH"
160 CALL CLEAR
170 CALL SCREEN(5)
180 Y=12 :: X=25 :: L=16 :: GOSUB 1
    30 :: FOR T=1 TO 100 :: NEXT T
190 GOSUB 140 :: FOR T=1 TO 300 ::
    NEXT T
200 FOR I=16 TO 24 :: DISPLAY AT(12
    ,25-I):B$ :: NEXT I :: FOR I=2
    TO 13 :: B$=SEG$(E$,I,12):: DIS
    PLAY AT(12,1):B$ :: NEXT I
210 GOSUB 840
220 CALL CLEAR :: CALL SCREEN(3)::
    PRINT "TO PLAY:": : :: PRINT "P
    OSITION GUN OVER CORRECT": : ::
     PRINT "ANSWER AND SHOOT THE UF
    O": :
230 PRINT "WITH FIREBUTTON OR SPACE
     BAR.": : : :: PRINT "BE CAREF
    UL NOT TO WASTE": : :: PRINT "S
    HOTS.": : : :
240 PRINT "FIRE OR <ENTER> TO START
    .": : : :
250 GOSUB 110 :: IF (KK<>18)*(K<>13
    )THEN 250
260 LEVEL=1 :: SC,HIT,GUN=0 :: NP=3
     :: PT(1)=25 :: PT(2)=10 :: PT(
    3)=5 :: SPR=136
270 CALL CLEAR :: CALL SCREEN(11)::
     DISPLAY AT(12,7):"ENTERING LEV
    EL";LEVEL :: GOSUB 140 :: FOR T
    =1 TO 100 :: NEXT T
280 PB,SCL=0
```

```
290 RANDOMIZE :: Y0=4 :: PF=5 :: A=
    INT(RND*4+1)+4*(LEVEL-1):: B=IN
    T(RND*4*LEVEL+1)
300 IF MENU=1 THEN C=A+B ELSE C=A*B
310 A$=STR$(A):: B$=STR$(B):: C$=ST
    R$(C)
320 RANDOMIZE :: RP=INT(RND*4)+1
330 IF RP=1 THEN ANS=A :: A$=CHR$(6
    3)
340 IF RP=2 THEN ANS=B :: B$=CHR$(6
    3)
350 IF (RP=3)+(RP=4)THEN ANS=C :: C
    $=CHR$(63)
360 CALL CLEAR :: CALL SCREEN(15)
370 D$=" + " :: IF MENU=2 THEN D$="
    z "
380 DISPLAY AT(4,10):A$;D$;B$;" = "
    ;C$
390 FOR I=1 TO NP :: CALL HCHAR(1,I
    *2+1,112):: NEXT I
400 DISPLAY AT(1,17):"SCORE:";SC ::
    PB=PB+1
410 IF ANS>90 THEN DEL=10 :: GOTO 4
    40
420 IF ANS>20 THEN DEL=5 :: GOTO 44
    0
430 DEL=INT((ANS/10)+1)
440 RANDOMIZE :: DT=INT(RND*5)+1
450 X1=256 :: X2=1
460 FOR I=1 TO 5 :: CH(I)=ANS+(I-DT
    )*DEL :: LOK(I)=5*I-1 :: DISPLA
    Y AT(24,LOK(I)):STR$(CH(I)):::
    NEXT I
470 CALL SPRITE(#2,96,6,150,LOK(PF)
    *8+4)
480 SHT=0 :: SP=14 :: FOR X0=X1 TO
    X2 STEP -DX :: CALL SPRITE(#1,1
    28,SP,Y0*8,X0):: SP=24-SP
490 CALL SOUND(-10,2000,3):: GOSUB
    110 :: PF=PF+SGN(X):: PF=PF+(PF
    =6)*5-(PF=0)*5 :: CALL LOCATE(#
    2,150,LOK(PF)*8+4):: GOSUB 110
500 IF ((KK=18)+(K=32))*(SHT=0)THEN
     CALL SPRITE(#3,124,2,150,LOK(P
    F)*8+4,-20,0):: CALL SOUND(200,
    -6,3):: GUN=1 :: SHT=1
510 CALL POSITION(#3,DR,DC):: IF (D
    R<10)+(DR>192)THEN CALL DELSPRI
    TE(#3)
520 CALL COINC(#1,#3,15,C):: IF C T
    HEN GOSUB 730
530 IF HIT=1 THEN X0=X2
540 NEXT X0 :: IF HIT=1 THEN HIT=0
    :: GOTO 550 ELSE CALL DELSPRITE
    (#3):: GUN=0 :: GOTO 570
550 IF (SCL>=50)THEN LEVEL=LEVEL+1
    :: CALL DELSPRITE(ALL):: GOTO 2
    70
560 CALL DELSPRITE(#1,#3):: GOTO 29
    0
570 Y0=Y0+4
580 IF Y0<13 THEN CALL DELSPRITE(#1
    ):: GOTO 480
590 FOR X0=X1 TO X2 STEP -DX :: CAL
    L LOCATE(#1,150,X0)
600 CALL COINC(#1,#2,10,C):: IF C T
    HEN CALL DELSPRITE(#2):: X0=X2
610 NEXT X0
620 FOR I=3 TO 15 STEP 3 :: SPR=276
```

```
     -SPR :: CALL SPRITE(#1,SPR,I+1,
     150,LOK(PF)*8+4):: CALL SCREEN
     (I):: CALL SOUND(-1 0+I*10,-5,3)
     :: NEXT I
630  CALL DELSPRITE(#1)
640  CALL HCHAR(1,NP*2+1,42):: FOR I
     =1 TO 30 STEP 3 :: FOR T=1 TO 3
     0 :: NEXT T :: CALL SOUND(10,22
     0,I):: NEXT I :: CALL HCHAR(1,N
     P*2+1,32)
650  NP=NP-1 :: FOR T=1 TO 5 :: DISP
     LAY AT(24,LOK(DT)):"{3 SPACES}"
     ;:: CALL SOUND(30,220,3):: CALL
      SOUND(40,440,3):: DISPLAY AT(2
     4,LOK(DT)-1):ANS;
660  CALL SOUND(30,220,3):: NEXT T
670  E$="KEEP TRYING !" :: X=25 :: Y
     =12 :: L=16 :: GOSUB 130
680  E$="I KNOW YOU CAN DO IT !" ::
     X=26 :: L=22 :: Y=14 :: GOSUB 1
     30
690  GOSUB 140 :: IF NP=0 THEN 700 E
     LSE 290
700  DISPLAY AT(16,6):"FINAL SCORE "
     ;SC :: DISPLAY AT(18,9):"PLAY A
     GAIN?" :: DISPLAY AT(20,4):"(<F
     IRE> OR PRESS 'Y')"
710  GOSUB 110 :: IF (KK<>18)*(K<>89
     )THEN 710 ELSE 210
720  REM CHECK COLLISION
730  IF ANS=CH(PF)THEN CALL DELSPRIT
     E(#3):: GOSUB 770 :: HIT=1 :: G
     OTO 750
740  CALL COLOR(#1,2):: CALL SOUND(1
     00,147,4,587,4,294,4,-1,3):: CA
     LL COLOR(#1,14)
750  GUN=0 :: RETURN
760  REM DIRECT HIT
770  FOR I=10 TO 30 STEP 5 :: SPR=27
     6-SPR :: CALL SCREEN(I/10+8)::
     CALL SOUND(-100,-6,I):: CALL
     SPRITE(#1,SPR,2+I/4,Y0*8,X0)::
     NEXT I
780  CALL SCREEN(15):: I=Y0/4 :: SC=
     SC+PT(I):: SCL=SCL+PT(I):: DISP
     LAY AT(14,9):PT(I);"POINTS"
790  IF (SCL>=50)*(SCL/PB>20)THEN DI
     SPLAY AT(16,7):"50 BONUS POINTS
     " :: SC=SC+50 :: GOTO 820
800  IF (SCL>=50)*(SCL/PB>=10)THEN D
     ISPLAY AT(16,7):"25 BONUS POINT
     S" :: SC=SC+25 :: GOTO 820
810  GOTO 830
820  FOR I=15 TO 3 STEP -1 :: CALL S
     CREEN(I):: NEXT I :: FOR I=3 TO
      15 :: CALL SCREEN(I):: NEXT I
830  DISPLAY AT(1,23):SC;:: FOR I=1
     TO 350 :: NEXT I :: CALL DELSPR
     ITE(ALL):: RETURN
840  CALL CLEAR :: CALL SCREEN(14)::
      PRINT TAB(6);"MISSILE MATH MEN
     U":: : :
850  PRINT "POSITION BALL WITH JOYST
     ICK/":: :
860  PRINT "ARROW KEYS AND FIRE/ENTE
     R TO":: :
870  PRINT "SELECT GAME.":: :
880  PRINT "TO:": : :: PRINT TAB(7);
     "ADD &";TAB(18);"MULTIPLY &"
890  PRINT TAB(7);"SUBTRACT";TAB(18)
     ;"DIVIDE": : : :
900  PRINT "SLOW": : : : :: PRINT "F
     AST": :
910  FOR J=17 TO 21 STEP 4 :: FOR CO
     L=11 TO 13 :: CALL VCHAR(J,COL,
     120,3):: CALL VCHAR(J,COL+11,12
     0,3):: NEXT COL :: NEXT J
920  FOR J=18 TO 22 STEP 4 :: CALL H
     CHAR(J,12,32):: CALL HCHAR(J,23
     ,32):: NEXT J
930  KHAR=122 :: PX=12 :: PY=18 :: M
     ENU=1 :: DX=9
940  CALL HCHAR(PY,PX,KHAR):: GOSUB
     110 :: KHAR=154-KHAR
950  IF ((SGN(X)=0)*(SGN(Y)=0))THEN
     970
960  CALL HCHAR(PY,PX,32):: IF (SGN(
     X)<>0)THEN MENU=3-MENU :: PX=35
     -PX ELSE PY=40-PY :: DX=27-DX
970  IF (KK=18)+(K=13)THEN RETURN EL
     SE 940
980  REM DEFINE CHARS, COLORS, & MUS
     ICAL SCORE
990  CALL CHAR(120,RPT$("F",16),112,
     "0000183C7E7E183C",122,"8142241
     818244281")
1000 CALL CHAR(96,"0000000003030303
     03030303F3F3F1818"&RPT$("0",22)
     &"C0F0F06060"):: REM BASE
1010 CALL CHAR(124,RPT$("0",11)&RPT
     $("10",6)&RPT$("0",51)):: REM
     MISSILE
1020 CALL CHAR(136,"0000201000000002
     80051030008"&RPT$("0",14)&"C4C
     000AC088"&RPT$("0",12)):: REM
     EXPLOSION 1
1030 CALL CHAR(140,"0010010000000080
     00100002"&RPT$("0",18)&"800400
     0000000020000000000"):: REM
     EXPLOSION 2
1040 CALL CHAR(128,"000000000000001F7
     FFFE6E67F1F"&RPT$("0",18)&"F8F
     EFF6767FEF8000000"):: REM UFO
1050 CALL MAGNIFY(4):: CALL COLOR(1
     1,13,1)
1060 FOR I=1 TO 14 :: READ NOTE(I),
     DUR(I):: NEXT I :: RETURN
1070 DATA 392,100,392,100,392,100,5
     23,900,392,100,392,100,392,100
1080 DATA 523,100,466,100,440,100,3
     92,700,40000,100,392,360,523,6
     00
1090 RETURN
```

Program 6: Apple Missile Math

```
0   HOME : VTAB 10: HTAB 14: PRINT "LOAD
    ING DATA"
1   GOSUB 1000
5   HOME
10  VTAB 4:A$ = "MISSILE MATH": GOSUB 9
    00
11  PRINT
15  VTAB 6: PRINT "            USE PAD
    DLE 0"
20  VTAB 8
25  PRINT : PRINT "          PRESS SPACE
    TO SELECT"
```

```
26   PRINT : PRINT "       PRESS RETURN
       TO START"
30   VTAB 15: PRINT "       ADDITION &
       MULTIPLICATION"
40   PRINT "    SUBTRACTION       & DIVI
       SION"
45   G = 0
49   VTAB 18
50   PRINT : PRINT "SLOW
       "
60   PRINT : PRINT "FAST
       "
70   IF G = 0 THEN  VTAB 19: HTAB 9
71   IF G = 1 THEN  VTAB 21: HTAB 9
72   IF G = 2 THEN  VTAB 19: HTAB 29
73   IF G = 3 THEN  VTAB 21: HTAB 29
74   PRINT "*"
80   POKE  - 16368,0
90   IF  PEEK ( - 16384) < 128 THEN 90
100  GET A$: IF A$ =  CHR$ (13) THEN 20
       0
110  IF A$ < > " " THEN 80
120  G = G + 1: IF G = 4 THEN G = 0
130  GOTO 49
200  L = 1:SC = 0:S = 3
205  IF G = 0 OR G = 2 THEN  POKE 768,1
       40: GOTO 210
206  POKE 768,80
210  TEXT : HOME : VTAB 7: PRINT "    P
       OSITION GUN OVER CORRECT ANSWER"
220  BN = 0:BO = 0:B = 0:SH = 0
230  PRINT : PRINT : PRINT "
       AND SHOOT UFO"
240  PRINT : PRINT : PRINT "     BE CAR
       EFUL NOT TO WASTE SHOTS"
250  PRINT : PRINT : PRINT "
       ENTERING LEVEL ";L
260  FOR I = 1 TO 5000: NEXT
270  GR : HOME : VTAB 23: HTAB 25: PRINT
       "SHIPS: ";S
280  HTAB 25: PRINT "SCORE: ";SC
290  A =  INT ( RND (1) * 4 + 1) + 4 * (
       L - 1)
291  B =  INT ( RND (1) * 4 * L + 1)
292  IF G < 2 THEN C = A + B: GOTO 294
293  C = A * B
294  A$ =  STR$ (A):B$ =  STR$ (B):C$ =
       STR$ (C)
295  RP =  INT ( RND (1) * 4) + 1
300  IF RP = 1 THEN ANS = A:A$ = "?"
310  IF RP = 2 THEN ANS = B:B$ = "?"
320  IF RP > 2 THEN ANS = C:C$ = "?"
330  VTAB 23: IF G < 2 THEN  PRINT A$+
       "B$"="C$: GOTO 340
335  PRINT A$"X"B$"="C$
340  IF ANS > 90 THEN DL = 10: GOTO 370
350  IF ANS > 20 THEN DL = 5: GOTO 370
360  DL =  INT ((ANS / 10) + 1)
370  DT =  INT ( RND (1) * 5) + 1
375  VTAB 21
380  FOR I = 1 TO 5:CH(I) = ANS + (I -
       DT) * DL: HTAB I * 4 + 4: PRINT CH
       (I);: NEXT
385  FOR I = 1 TO 5: IF CH(I) = ANS THEN
       POKE 800,I - 1
386  NEXT
390  CALL 24576
395  SH = SH + 1
400  X =  PEEK (769): IF X = 35 THEN 600
```

```
500  IF X = 25 THEN B = 5
510  IF X = 15 THEN B = 10
520  IF X = 5 THEN B = 25
521  HOME
525  BN = BN + B
526  SC = SC + B: VTAB 23:A$ =  STR$ (B)
       + " POINTS": GOSUB 900
530  IF BN > = 50 THEN L = L + 1: GOTO
       700
550  GOTO 260
600  S = S - 1
601  HTAB 1: VTAB 21: PRINT "
       ";:A$ =
       "KEEP TRYING": GOSUB 900
602  PRINT :A$ = "I KNOW YOU CAN DO IT"
       : GOSUB 900
603  PRINT :A$ = "THE ANSWER WAS:": GOSUB
       900: PRINT :A$ =  STR$ (ANS): GOSUB
       900
610  FOR I = 1 TO 1000: NEXT
620  IF S = 0 THEN 800
630  GOTO 260
700  BO =  INT (BN / SH + .5)
710  IF BO > = 25 THEN BO = 50: GOTO 7
       50
720  IF BO > = 10 THEN BO = 25: GOTO 7
       50
730  GOTO 760
750  VTAB 21
755  A$ = "BONUS: " +  STR$ (BO): GOSUB
       900:SC = SC + BO
760  FOR I = 1 TO 5000: NEXT : GOTO 210
800  TEXT : HOME : VTAB 5: PRINT "
       SCORE: "SC
810  PRINT : PRINT : HTAB 18: PRINT "GA
       ME": PRINT : PRINT : HTAB 18: PRINT
       "OVER"
820  VTAB 20: HTAB 5: PRINT "PRESS ANY
       KEY TO PLAY AGAIN"
830  POKE  - 16368,0
840  IF  PEEK ( - 16384) < 128 THEN 840
850  GET A$: GOTO 5
900  X =  LEN (A$):A$ = A$ + " "
910  FOR I = 1 TO X + 19 -  INT (X / 2 +
       .5)
920  HTAB 39 - I: PRINT  MID$ (A$,1,I);
930  NEXT I
940  RETURN
1000 CK = 0
1010 FOR I = 24576 TO 25329: READ A:CK
       = CK + A: POKE I,A: NEXT
1020 IF CK < > 73926 THEN  PRINT "ERR
       OR IN DATA": END
1030 RETURN
1100 DATA  76,22,96,255,160,160,232,16
       9
1110 DATA  201,197,160,160,176,165,162
       ,160
1120 DATA  128,6,10,14,18,22,32,46
1130 DATA  96,32,86,96,32,216,97,32
1140 DATA  111,98,32,175,98,173,16,96
1150 DATA  208,239,32,212,98,96,169,1
1160 DATA  141,16,96,169,4,96,169,2
1170 DATA  141,5,96,169,33,141,7,96
1180 DATA  141,8,96,169,5,141,12,96
1190 DATA  141,1,3,169,0,141,13,96
1200 DATA  169,9,141,14,96,96,173,4
1210 DATA  96,205,5,96,240,34,169,0
1220 DATA  32,100,248,173,4,96,141,3
```

```
1230  DATA  96,32,43,97,169,6,32,100
1240  DATA  248,173,5,96,141,3,96,32
1250  DATA  43,97,173,5,96,141,4,96
1260  DATA  169,0,32,100,248,173,7,96
1270  DATA  141,6,96,173,12,96,141,9
1280  DATA  96,32,126,97,173,14,96,32
1290  DATA  100,248,173,8,96,141,6,96
1300  DATA  173,1,3,141,9,96,32,126
1310  DATA  97,173,8,96,141,7,96,173
1320  DATA  1,3,141,12,96,173,13,96
1330  DATA  201,1,208,68,169,0,32,100
1340  DATA  248,172,2,3,185,17,96,24
1350  DATA  105,2,168,173,10,96,32,0
1360  DATA  248,172,2,3,185,17,96,24
1370  DATA  105,2,168,173,11,96,32,113
1380  DATA  248,201,0,208,28,169,15,32
1390  DATA  100,248,172,2,3,185,17,96
1400  DATA  24,105,2,168,173,11,96,32
1410  DATA  0,248,173,11,96,141,10,96
1420  DATA  96,173,32,3,205,2,3,240
1430  DATA  6,169,15,141,14,96,96,169
1440  DATA  0,141,16,96,169,15,141,14
1450  DATA  96,173,14,96,32,100,248,32
1460  DATA  126,97,32,187,98,206,14,96
1470  DATA  208,239,96,174,3,96,189,17
1480  DATA  96,24,105,2,168,169,36,133
1490  DATA  45,169,34,32,40,248,174,3
1500  DATA  96,189,17,96,168,200,24,105
1510  DATA  3,133,44,169,37,32,25,248
1520  DATA  174,3,96,189,17,96,168,24
1530  DATA  105,4,133,44,169,38,32,25
1540  DATA  248,174,3,96,189,17,96,168
1550  DATA  200,169,39,32,0,248,174,3
1560  DATA  96,189,17,96,168,200,200,20
      0
1570  DATA  169,39,32,0,248,96,173,6
1580  DATA  96,168,24,105,5,133,44,173
1590  DATA  9,96,200,32,25,248,173,6
1600  DATA  96,168,24,105,6,133,44,173
1610  DATA  9,96,24,105,1,32,25,248
1620  DATA  173,6,96,168,24,105,6,133
1630  DATA  44,173,9,96,24,105,2,32
1640  DATA  25,248,173,6,96,168,24,105
1650  DATA  6,133,44,173,9,96,24,105
1660  DATA  3,32,25,248,173,6,96,168
1670  DATA  24,105,5,133,44,200,173,9
1680  DATA  96,24,105,4,32,25,248,96
1690  DATA  162,0,32,30,251,152,201,50
1700  DATA  144,17,201,100,144,18,201,1
      50
1710  DATA  144,19,201,200,144,20,169,4
1720  DATA  76,7,98,169,0,76,7,98
1730  DATA  169,1,76,7,98,169,2,76
1740  DATA  7,98,169,3,76,7,98,141
1750  DATA  5,96,173,8,96,240,6,206
1760  DATA  8,96,76,55,98,169,33,141
1770  DATA  8,96,32,84,98,169,9,141
1780  DATA  14,96,169,0,141,13,96,173
1790  DATA  1,3,24,105,10,141,1,3
1800  DATA  201,35,208,3,76,147,98,173
1810  DATA  13,96,201,1,208,48,173,11
1820  DATA  96,201,2,144,15,206,11,96
1830  DATA  206,11,96,206,11,96,206,11
1840  DATA  96,76,110,98,169,2,141,13
1850  DATA  96,169,0,32,100,248,172,2
1860  DATA  3,185,17,96,24,105,2,168
1870  DATA  173,10,96,32,0,248,96,173
1880  DATA  97,192,48,1,96,173,13,96
1890  DATA  201,0,208,22,169,33,141,10
1900  DATA  96,141,11,96,169,1,141,13
1910  DATA  96,173,4,96,141,2,3,32
1920  DATA  198,98,96,169,0,141,16,96
1930  DATA  169,15,141,14,96,173,14,96
1940  DATA  32,100,248,32,43,97,32,187
1950  DATA  98,206,14,96,208,239,96,174
1960  DATA  0,3,160,0,200,208,253,202
1970  DATA  208,250,96,162,10,160,0,200
1980  DATA  208,253,202,208,250,96,162,
      15
1990  DATA  173,48,192,169,4,32,168,252
2000  DATA  202,208,245,96,169,16,141,1
      5
2010  DATA  96,160,1,162,1,169,80,32
2020  DATA  168,252,173,48,192,232,208,
      253
2030  DATA  136,208,240,206,15,96,208,2
      33
2040  DATA  96,255,0,0,255,255,0,0
```

Program 7: PC/PCjr Missile Math

```
10 WIDTH 40:KEY OFF:SCREEN 0,1:DEF SEG=0
   :POKE 1047,64:DEFINT A-Z
20 GOSUB 610:GOSUB 450:GOSUB 480:GOSUB 5
   10:GOTO 110
30 IF Y2=N21 THEN RETURN ELSE C$=INKEY$:
   IF C$="." AND H<N28 THEN LOCATE N21,H:PR
   INT "  ":H=H+N5
40 IF C$="," AND H>N9 THEN LOCATE N21,H:
   PRINT "  ":H=H-N5:Z=RND(1)
50 IF C$=CHR$(N32) AND FIRE =N0 AND FI=N
   0 THEN FIRE=N1:FI=N1:Y=N20:X=H+N1:PANS=C
   H((H-N3)/N5):NS=NS+N1:Z=RND(1):RETURN
60 LOCATE N21,H:COLOR N3:PRINT BASE$:RET
   URN
70 X2=X2-N1:IF X2=N1 THEN LOCATE Y2,X2+N
   1:PRINT "    ":Y2=Y2+N5:X2=N35:FI=N0:R=N
   4
80 SP=SCREEN(Y2,X2):IF SP=21 THEN FL=2:K
   =S:RETURN:ELSE IF SP=202 THEN FL=3:K=S:R
   ETURN
90 LOCATE  Y2,X2:COLOR R:PRINT SPCE$:RET
   URN
100 COL=COL+1:IF COL=8   THEN COL =2
110 COLOR COL:CLS:W$="ENTERING LEVEL"+ST
    R$(LVL):YY=12:GOSUB 590:GOSUB 630
120 IF NS THEN IF NS<3 THEN W$="50 POINT
     BONUS":YY=14:GOSUB 600:ELSE IF NS<6 THE
    N W$="25 POINT BONUS":YY=14:GOSUB 600
130 FOR TD=1 TO 1000:NEXT:SCL=0:NS=0
140 CLS:LOCATE 2,1:FOR J=1 TO NM:COLOR 1
    3:PRINT CHR$(6)" ";:NEXT
150 LOCATE 2,29:PRINT"SCORE "SC
160 X2=36:Y2=6:T=INT(RND(1)*20)+40:FL=1:
    FI=0:FIRE=0:R=4
170 A=INT(RND(1)*4+1)+4*(LVL-1)
180 B=INT(RND(1)*4*LVL+1)
190 IF P$="A" THEN C=A+B:GOTO 210
200 C=A*B
210 A$=STR$(A):B$=STR$(B):C$=STR$(C)
220 RP=INT(RND(1)*4)+1
230 IF RP=1 THEN ANS=A:A$=CHR$(63)
240 IF RP=2 THEN ANS=B:B$=CHR$(32)+CHR$(
    63)
250 IF RP=3 OR RP=4 THEN ANS=C:C$=CHR$(3
    2)+CHR$(63)
260 IF P$="A" THEN LOCATE 4,14:COLOR 7:P
    RINT A$" +"B$" ="C$:GOTO 280
```

www.commodore.ca

```
270 COLOR 7:LOCATE 4,14:PRINT A$" X"B$"
    ="C$
280 IF ANS>90 THEN DEL=10:GOTO 320
290 IF ANS>20 THEN DEL=5:GOTO 320
300 DEL=INT((ANS/10)+1)
310 DT=INT(RND(0)*5)+1
320 FOR I=1 TO 5:CH(I)=ANS+(I-DT)*DEL:LO
CATE 22,5*I+3:PRINT CH(I):NEXT I
330 GOSUB 30:GOSUB 410:T=T-N1:IF T<N0 TH
EN FOR K=N1 TO S:GOSUB 410:GOSUB 70:NEXT
:GOSUB 410
340 ON FL GOTO 330,350,370
350 R=N7:FL=N1:IF ANS<>PANS THEN 330 ELS
E FOR TI=N1 TO N3:COLOR TI:LOCATE Y2-N1,
X2:PRINT E1$:LOCATE Y2,X2:PRINT E2$:LOCA
TE Y2+N1,X2:PRINT E3$:LOCATE Y2+N2,X2:PR
INT P(Y2)" POINTS":FOR TD=37 TO 45:SOUND
 TD,.5:NEXT TD,TI:FIRE=0:FOR TD=1 TO 900
:NEXT
360 SC=SC+P(Y2):SCL=SCL+P(Y2):IF SCL>=50
 THEN LVL=LVL+1:SCL=0:GOTO 100:ELSE 140
370 NM=NM-1:SOUND 100,20:FOR TD=1 TO 600
:NEXT:SOUND 37,20:FOR TI=1 TO 5:FOR TD=1
 TO 600:NEXT:LOCATE 22,5*DT+3:PRINT "
 ":FOR TD=1 TO 60:NEXT:LOCATE 22,5*DT+3:
COLOR 4:PRINT CH(DT):FOR TD=1 TO 60:NEXT
:NEXT
380 IF NM>0 THEN W$="KEEP TRYING":YY=10:
GOSUB 590:GOSUB 630:GOTO 140
390 LOCATE 2,1:PRINT " ":W$="PLAY AGAIN
?(Y/N)":YY=10:GOSUB 590
400 I$=INKEY$:IF I$="Y" THEN RUN ELSE IF
 I$="N" THEN CLS:END:ELSE 400
410 IF FIRE=N0  THEN FOR TD=1 TO 10:NEXT
:RETURN

420 Y=Y-1:IF Y=5 THEN FIRE=0:LOCATE Y+1,
X:PRINT" ":RETURN
430 SP=SCREEN(Y,X):IF SP<>32 THEN FL=2
440 SOUND 200-Y*3,.1:LOCATE Y,X:COLOR 14
:PRINT MISS$:RETURN
450 CLS:W$="MISSILE MATH !":YY=10:COLOR
2:GOSUB 590:FOR TD=1 TO 800:NEXT
460 COLOR 3:W$="< MOVES YOU LEFT":YY=12:
GOSUB 590:COLOR 4:W$="> MOVES YOU RIGHT"
:YY=14:GOSUB 600:LOCATE 16,11:COLOR 7:PR
INT "SPACE BAR TO FIRE":GOSUB 630:RETURN
470 FOR TD=1 TO 2000:NEXT:CLS:RETURN
480 SPCE$=CHR$(17)+STRING$(2,219)+CHR$(1
74)+CHR$(32)+CHR$(32):BASE$=CHR$(32)+CHR
$(202)+CHR$(32):MISS$=CHR$(21)+CHR$(31)+
CHR$(29)+CHR$(32):E1$=CHR$(92)+STRING$(2
,179)+CHR$(47):E2$=CHR$(196)+STRING$(2,3
2)+CHR$(196)+CHR$(32)
490 E3$=CHR$(47)+STRING$(2,179)+CHR$(92)
500 H=28:LVL=1:NM=3:N1=1:N21=21:N32=32:N
28=28:N0=0:N2=2:N3=3:N202=202:N5=5:N9=9:
N32=32:N20=20:N35=35:N4=4:N7=7:DIM P(16)
:P(6)=25:P(11)=10:P(16)=5:COL=2:RETURN
510 CLS:LOCATE 8,8,0:COLOR 7:PRINT "(A)D
DITION AND SUBTRACTION":LOCATE 10,19:PRI
NT"OR"
520 LOCATE 12,7 :PRINT"(M)ULTIPLICATION
AND DIVISION"
530 P$=INKEY$:IF P$<>"A"AND P$<>"M" THEN
 530
540 LOCATE 15,12:PRINT "(F)AST OR (S)LOW
"
550 S$=INKEY$:IF S$="S" THEN S=1 ELSE IF
 S$="F" THEN S=2 ELSE 550
560 LOCATE 21,11:PRINT "HIT ANY KEY TO S
TART"
570 Z=RND(1):I$=INKEY$:IF I$=""THEN 570
580 RETURN
590 W=LEN(W$):NW$=W$+STRING$(20-W/2,32):
FOR K=1 TO LEN(NW$):LOCATE YY,39-K,0:PRI
NT LEFT$(NW$,K):NEXT:RETURN
600 W=LEN(W$):NW$=STRING$(18-W/2,32)+W$:
FOR K=1 TO LEN(NW$):LOCATE YY,1:PRINT RI
GHT$(NW$,K):NEXT:RETURN
610 DIM NO(14,2):FOR A=1 TO 14:FOR B=1 T
O 2:READ NO(A,B):NEXT B,A:RETURN
620 DATA 196,1,196,1,196,1,261,9,196,1,1
96,1,196,1,261,1,232,1,220,1,196,7,32000
,1,196,3,261,6
630 FOR R1=1 TO 14:SOUND NO(R1,1),NO(R1,
2)*1.5:FOR TD=1 TO NO(R1,2)*90 :NEXT:NEX
T:RETURN                              ©
```

www.commodore.ca

Lightsaver

Jon Rhees

The world is depending on you to save its supply of light bulbs. A quick eye and an even faster hand are the only two things that will help you. This arcade-style action game uses machine language for fast action. Versions are included for the Commodore 64 and VIC-20, the Atari home computers, and the IBM PC/PCjr.

Technology has backfired again! A dangerous power surge has transformed the light fixture in your room into a devilish lamp, which is trying to rid the world of light bulbs by tossing them to the ground. Only you can stop the lamp from carrying out its horrid plan. You must catch the bulbs in your baskets before they hit the ground and break. The more bulbs you save, the more angry the lamp will become, and the faster it will drop bulbs. How long can you survive the wrath of the lamp?

"Lightsaver" is a fast-action, arcade-style game which utilizes many of the capabilities of the Commodore 64. Most of the action in this game is written in machine language for speed, while the scoring and setup routines are written in BASIC. The game requires a set of paddle controllers, which should be plugged into port 1.

The object of the game is simple: catch as many light bulbs as you can without missing or dropping any. You must catch each bulb in your baskets. At the beginning of a game you start out with three baskets, stacked atop each other. You position the baskets by turning the paddle controller. Each time you miss a bulb, you lose a basket, and are set back one level. Each consecutive level has more bulbs to catch and faster action. Bonus baskets are awarded every 2000 points, but you can have only three baskets at a time. Also, the higher the level you are on, the more points each bulb you catch is worth.

There are 16 skill levels in Lightsaver and two basket sizes. The higher the skill level you choose, the faster the game. Bulbs are easier to catch with a large basket than with a small one. A large basket is twice as wide as a small one. You can change the basket size during the game by pressing the f1 key. Press the paddle fire button to start a new round.

Lightsaver takes advantage of the sprite, redefined character, color, and sound capabilities of the 64. It contains two separate machine language routines—one for the game action, and another to quickly copy the character ROM into RAM for programmable characters. A demo mode is also included. The computer plays a game against itself if left alone. All eight sprites are used, as well as a redefined character set. The first two voices of the SID (Sound Interface Device) chip are used for sound effects, and the third voice is used as a random number generator for the machine language routine.

If you'd rather not type in the program, you can obtain tape or disk copies (64 version only). Send an SASE, a disk or tape, and $3 to:

Jon Rhees
1660 S. Duneville
Las Vegas, NV 89102

Program 1: 64 Lightsaver

Refer to the "Automatic Proofreader" article before typing this program in.

```
1 GOSUB550                                    :rem 75
2 Z=646:POKE53280,0:POKE53281,0:POKEZ,1:X
  =58692:SYSX:V=53248                         :rem 89
3 G=54272:POKEG+6,0:POKEG+5,0:POKEG+4,0:P
  OKEG+6,240:POKEG+4,17:POKEG+24,143
                                              :rem 122
4 POKEG+6,240:NS$="000000":GOSUB400
                                              :rem 187
10 T=49152:H=54273:B=832:C=1022:E=49550:F
   ORA=BTOC:READM:POKEA,M:POKEH,M:NEXT
                                              :rem 131
20 B=V+39:FORA=TTOE:READM:POKEA,M:POKEH,M
   :POKEB,M:NEXT:POKEG+5,186:POKEG+6,.
                                              :rem 124
26 POKE252,208:POKE251,.:POKE253,.:BL=872
   :POKE254,48                                :rem 240
28 POKE56334,.:POKE1,51:SYS49490:POKE4951
   3,56:POKE49500,145:POKE49501,253
                                              :rem 208
29 POKE49502,200:POKE49503,234:SYS49490:P
   OKE1,55:POKE56334,1:GOTO600   :rem 214
60 POKE53280,3:SYSX:PRINT"{5 DOWN}
   {7 SPACES}SELECT SKILL LEVEL (A-P) ";
                                              :rem 232
61 GETA$:SL=ASC(A$+" ")-59:IFSL<6 OR SL>2
   1 THEN 61                                  :rem 170
62 PRINTA$:POKE53280,14:PRINT:PRINT"
   {3 DOWN}{7 SPACES}SELECT BASKET SIZE (
   S/L) ";:P=.                                :rem 95
```

Notes For VIC-20, Atari, And IBM PC/PCjr Versions

In "Lightsaver," you must catch light bulbs that fall from a lamp at the top of the screen.

The VIC Version

Program 2, for the unexpanded VIC-20, is written entirely in machine language and offers five levels of play. The level selected affects the speed of the falling bulbs at the beginning of play. The speed increases when a light bulb is caught in your basket and decreases when light bulbs crash to the bottom of the screen. The game ends when three bulbs have been missed.

You move your baskets under the falling bulbs by using a paddle plugged into the control port.

Since the VIC version is written in machine language, you must use a special technique to enter the program. You must first type in the "Tiny MLX" program found elsewhere in this issue, and then use this program to type VIC Lightsaver. Be sure you read and understand the Tiny MLX article before you begin typing Lightsaver. Also, remove or disable any memory expanders. Once you have typed in Lightsaver, and saved it to tape or disk using MLX, you can load and run the program just as you would a BASIC program.

The Atari Version

To catch bulbs in the Atari version of Lightsaver, you must move your basket with the paddle controller and touch the bulbs before they hit the bottom of the screen. If a light bulb reaches the bottom of the screen, it will break and you will lose one of your three baskets. The number of points awarded for catching a bulb is equal to the current level. For instance, on the fourth level, every bulb that you catch is worth four points. You are given an additional basket every 2000 points. Each time you complete five levels, the speed of the light bulbs increases.

Atari Lightsaver has been split into two programs so it will work on 16K computers with DOS. Load and run Program 3. This program loads two machine language subroutines and the redefined character set. When finished, this program will load and run Program 4. Program 4 (the main program) contains the player missile graphics and the main game loop. Since Program 3 loads and runs Program 4, disk users must save both programs on the same disk. Program 4 must be saved with filename LIGHTSAVER.

Cassette users must change line 1046 in Program 3 to:

1046 READ A: IF A = −1 THEN PRINT "HIT RETURN TO LOAD LIGHTSAVER": RUN "C:"

Next, rewind a tape to a blank section, press the PLAY and RECORD buttons, and enter SAVE "C:" to store Program 3. When the cassette is finished, type in Program 4. Save Program 4 immediately following Program 3 by once again entering SAVE "C:". To load and run the two programs, rewind the tape to the start of Program 3 and type RUN "C:".

The IBM Version

In the IBM version of Lightsaver (Program 5), you must catch bulbs before they fall below basket level. The IBM version has only one difficulty level and a constant basket size. Game play consists of several rounds, each consisting of a barrage of dropping bulbs. Between 10 and 20 bulbs are dropped before the round ends. Each additional round increases the motion of the lamp which drops the bulbs and increases the number of points awarded. The formula is simple: The number of points awarded equals the number of bulbs caught, multiplied by the difficulty level.

You can catch a bulb only if it drops into the basket from above. It is possible to break a bulb by hitting it with the rim of your basket. If you let a bulb break, the round ends and the difficulty level decreases by one.

IBM Lightsaver will run on PCs with Advanced BASIC (BASICA) and PCjrs with DOS and Cartridge BASIC. A Color/Graphics Monitor Adapter card is needed on PCs. A joystick is required to control the left-right movement of the baskets.

Since the video display processor of the IBM PC may refresh the screen while we are using the PUT statement to place graphics on the screen, a short machine language routine has been included to synchronize PUT with the vertical retrace interval.
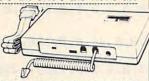
```
           MOV DX,3DAh
CHECKPORT  IN AX,DX
           AND AX,8
JZ CHECKPORT
RETF
```

This machine language routine reduces flicker when the baskets are moved around the screen.

```
63 GETS$:IFS$<>"S"ANDS$<>"L"THEN63
                                    :rem 253
64 PRINTS$"{5 DOWN}":POKE53280,6:LV=.:POK
   E209,88:POKE210,6:IFBW=141THENGOSUB500
                                    :rem 146
65 PRINT"PRESS <F1> TO CHANGE BASKET SIZE
    IN PLAY":GOSUB700             :rem 22
67 POKE63,.:POKE65,.:POKE67,.:SYSX:NS$="0
   00000":GOSUB202                :rem 60
68 POKEV+29,-(S$="L"):GOSUB340:GOTO71
                                    :rem 163
70 GOSUB200                       :rem 121
71 PRINT"{HOME}{17 DOWN}{2 SPACES}{GRN}PR
   ESS FIRE BUTTON WHEN READY!"    :rem 226
72 N=244-PEEK(54297):N=N*-(N>=24)-24*(N<2
   4):POKEV,N                      :rem 93
73 POKEV+29,(PEEK(V+29)-(PEEK(197)=4))AND
   1:IF(PEEK(56321)AND4)THEN72     :rem 158
74 PRINT"{UP}{31 SPACES}":POKE254,.:SYST:
   IFPEEK(68)THEN80                :rem 252
76 M=54273:POKEF+11,.:N=5:I=125:J=53280:F
   ORA=NTOISTEP3:POKEJ,A           :rem 15
77 POKEM,AAND(LV+5)*2:NEXT:POKEF+5,186:PO
   KEF+6,.:POKEF+12,186:POKEF+13,.:rem 54
79 POKEJ,.:GOTO70                  :rem 29
80 POKEF+11,.:POKEF+4,.:POKEF+1,186:POKEF
   +4,129:POKEF+5,188:POKEF+6,.    :rem 30
81 POKEV+4,PEEK((PEEK(781)+4)+V):POKEV+5,
   232:B=50:D=-.9:E=V+3:F=10264    :rem 175
82 FORA=BTO.STEPD:POKEE,A:N=RND(.)*21:M=N
   *1.4:POKEN+BL,PEEK(N+BL)ANDRND(.)*256
                                    :rem 23
83 POKEM+F,PEEK(M+F)ORRND(.)*256:POKE780,
   M:SYS49540:NEXT:POKEV+41,0:POKE53281,0
                                    :rem 47
84 LV=LV-2:P=P+1:POKEV+2,.:LV=LV*-(LV>-1)
   :FORA=BLTOBL+20:POKEA,.         :rem 197
85 NEXT:BL=BL-20:IFBW=141THENGOTO600
                                    :rem 1
86 GOSUB200:IFP<3THENGOSUB340:GOTO71
                                    :rem 41
88 IFVAL(NS$)=VAL(HS$)ANDVAL(NS$)>0THENGO
   TO900                           :rem 99
90 POKEV+21,.:PRINT"{HOME}{7 DOWN}{CYN}"S
   PC(9)"** GAME OVER **":BL=872:P=.
                                    :rem 87
96 GOSUB800:GOTO600                :rem 144
100 DATA 255,255,255,109,182,218,27
                                    :rem 192
101 DATA 109,180,13,182,216,11,109,176
                                    :rem 69
102 DATA 7,255,224,0,0,0,255,255,255
                                    :rem 220
103 DATA 91,109,182,45,182,216,27,109
                                    :rem 33
104 DATA 176,13,182,208,7,255,224,0,0,0
                                    :rem 109
105 DATA 255,255,255,109,182,218,27,109
                                    :rem 139
106 DATA 180,13,182,216,11,109,176,7
                                    :rem 231
107 DATA 255,224,0,0,0,0,0,60,0,0,60,0
                                    :rem 22
108 DATA 0,60,0,0,60,0,0,60,0,0,60,0,0
                                    :rem 11
109 DATA 60,0,0,60,0,0,60,0,0,60,0,0,60
                                    :rem 66
110 DATA 0,0,60,0,0,60,0,0,60,0,0,60,0
                                    :rem 4
111 DATA 0,20,0,0,85,0,1,85,64,5,85,80
                                    :rem 50
112 DATA 21,85,84,170,170,170,0,0,0,0,0
                                    :rem 90
113 DATA 0,0,0,0,0,0,0,0,0,24,0,0,60,0
                                    :rem 247
114 DATA 0,60,0,0,36,0,0,126,0,0,126,0
                                    :rem 21
115 DATA 0,255,0,1,255,128,1,255,128,3
                                    :rem 60
116 DATA 253,192,1,253,128,1,251,128,0
                                    :rem 66
117 DATA 255,0,0,60,0,0,0,0,0,0,0,0,0,0
                                    :rem 49
118 DATA 120,173,2,220,72,169,192,141,2
                                    :rem 117
119 DATA 220,169,64,141,0,220,160,128
                                    :rem 19
120 DATA 234,136,16,252,56,169,220,237
                                    :rem 79
121 DATA 25,212,176,2,169,0,24,105
                                    :rem 122
122 DATA 24,141,0,208,104,141,2,220,88
                                    :rem 54
123 DATA 172,30,208,169,8,133,253,162,2
                                    :rem 123
124 DATA 254,5,208,189,4,208,240,33,169
                                    :rem 135
125 DATA 188,56,253,5,208,176,13,152,37
                                    :rem 139
126 DATA 253,240,8,169,0,157,4,208,76
                                    :rem 35
127 DATA 14,193,189,5,208,201,232,208,5
                                    :rem 128
128 DATA 169,1,133,68,96,232,232,6,253
                                    :rem 88
129 DATA 224,12,240,3,76,51,192,234,164
                                    :rem 125
130 DATA 197,192,4,208,17,165,255,201
                                    :rem 32
131 DATA 64,208,11,169,1,24,109,29,208
                                    :rem 75
132 DATA 41,1,141,29,208,132,255,164,64
                                    :rem 120
133 DATA 240,99,132,64,173,2,208,24,101
                                    :rem 120
134 DATA 254,141,2,208,198,2,16,84,169
                                    :rem 86
135 DATA 31,160,0,133,2,173,27,212,133
                                    :rem 55
136 DATA 254,56,237,2,208,176,8,173,2
                                    :rem 37
137 DATA 208,56,229,254,160,1,74,74,74
                                    :rem 91
138 DATA 74,74,133,254,56,233,4,176,225
                                    :rem 140
139 DATA 192,1,208,9,165,254,169,0,56
                                    :rem 41
140 DATA 229,254,133,254,230,66,230,66
                                    :rem 79
141 DATA 166,66,224,10,208,4,162,0,134
                                    :rem 65
142 DATA 66,169,80,157,7,208,173,2,208
                                    :rem 90
143 DATA 157,6,208,234,234,206,64,0
                                    :rem 183
144 DATA 169,146,133,150,198,150,165
                                    :rem 247
145 DATA 150,141,1,212,230,9,165,9,141
                                    :rem 63
146 DATA 8,212,201,30,208,3,141,11,212
                                    :rem 50
```

C www.commodore.ca

```
147 DATA 172,52,3,32,204,255,32,204,255
                                    :rem 120
148 DATA 136,192,0,208,248,76,0,192,165
                                    :rem 138
149 DATA 251,248,24,101,67,133,67,169,0
                                    :rem 135
150 DATA 101,65,133,65,169,0,101,63,133
                                    :rem 113
151 DATA 63,216,162,4,181,63,72,41,15
                                    :rem 23
152 DATA 24,105,48,157,19,5,104,74,74
                                    :rem 33
153 DATA 74,74,24,105,48,157,18,5,202
                                    :rem 32
154 DATA 202,16,230,169,129,141,11,212
                                    :rem 65
155 DATA 169,0,133,9,198,252,208,3,133
                                    :rem 84
156 DATA 68,96,76,92,192,160     :rem 122
157 DATA 0,177,251,17,253,145,253,177,251
    ,74,200,145,253,208,242        :rem 86
158 DATA 230,252,230,254,165,254,201,52
                                    :rem 123
159 DATA 208,230,96,234,238,0,201,174,0
                                    :rem 124
160 DATA 201,189,0,200,168,138,105,109
                                    :rem 71
161 DATA 170,173,2,208,157,0,200,234,140,
    141,41,208,141,32,208,141,33,208,96,3
    2                               :rem 22
162 DATA 48,126,103,103,103,103,103,127
                                    :rem 106
163 DATA 63,49,28,60,124,28,28,28,127
                                    :rem 41
164 DATA 63,50,126,103,7,127,112,115,127
                                    :rem 166
165 DATA 63,51,126,103,7,63,7,103 :rem 81
166 DATA 127,63,52,6,15,27,51,127,127
                                    :rem 32
167 DATA 7,15,53,126,96,126,103,7,103
                                    :rem 29
168 DATA 127,63,54,126,103,96,126,103
                                    :rem 34
169 DATA 103,127,63,55,126,103,7,14,28
                                    :rem 78
170 DATA 28,28,28,56,126,103,103,127
                                    :rem 234
171 DATA 103,103,127,63,57,126,103,103
                                    :rem 63
172 DATA 127,7,103,127,63,104,0,0,0,0
                                    :rem 253
173 DATA 239,239,85,254           :rem 125
200 NS$="":FORA=1298TO1303:NS$=NS$+CHR$(P
    EEK(A)):NEXT                    :rem 99
201 AA=INT(VAL(NS$)/2000)-INT(Q/2000):IFA
    ATHENGOSUB300                  :rem 176
202 Q=VAL(NS$):IFQ>=VAL(HS$)THENHS$=NS$
                                    :rem 147
204 POKE53280,.:LV=LV+1:LV$=STR$(LV)+"
    {2 SPACES}":IFLV>1THEN208       :rem 151
206 FORA=1024TO1057STEP2:POKEA,95:POKEA+1
    ,105:POKE54272+A,4:POKE54273+A,10:NEX
    T                               :rem 199
207 FORA=1984TO2023:POKEA,104:POKEA+54272
    ,12:NEXT                        :rem 138
208 PRINTCHR$(19)CHR$(159)CHR$(18)SPC(34)
    CHR$(169)"LIGHT"CHR$(146);:POKEZ,13
                                    :rem 86
209 PRINTSPC(34)CHR$(127)CHR$(18)"SAVER":
    POKEZ,7:PRINTSPC(74)"SCORE:"SPC(74);
                                    :rem 152
210 PRINTCHR$(5)NS$:POKEZ,8:PRINTSPC(76)"
    HI"SPC(76)"SCORE:"SPC(74)CHR$(5)HS$
                                    :rem 89
211 PRINTSPC(34);:POKEZ,6:PRINT"*";:POKEZ
    ,10:PRINTHU$;:POKEZ,6:PRINT"*"
                                    :rem 252
212 POKEZ,14:PRINTSPC(74)"LEVEL:"SPC(75)C
    HR$(5)LV$:F=54272:POKEF+20,128
                                    :rem 194
216 POKE2040,13:POKE2041,14:POKEF+19,.
                                    :rem 125
220 POKEV+1,205:POKEV+3,50:POKEV+23,3:POK
    EV+28,2:POKEV+39,7:POKEF+18,.:rem 156
222 FORA=4TO14STEP2:POKEV+A,.:POKEV+A+1,A
    *8+140:POKE2040+A/2,15:NEXT     :rem 71
223 FORA=10240TO10303:POKEA,.:NEXT:POKE20
    42,160:POKE66,.:POKE2,.         :rem 34
226 POKEV+37,12:POKEV+38,14:FORA=41TO46:P
    OKEV+A,1:NEXT:POKEV+21,255:POKEF+19,.
                                    :rem 238
228 POKEF+11,0:POKEF+7,0:POKEF+13,128:POK
    EF+12,.:K=260-SL*LV:POKEF+20,128
                                    :rem 248
230 POKE64,LV*2+9:POKE252,LV*2+9:LN=LV:PO
    KEF+15,255:IFLN>100THENLN=100:rem 208
232 POKE251,INT(LN/10)*16+LN-INT(LN/10)*1
    0:POKEF+14,.:POKEF+18,129:IFK<1THENK=
    1                               :rem 113
233 POKEF+1,.:POKEF+4,.:POKEF+5,.:POKEF+6
    ,128:POKEF,.:POKEF+24,143:POKEF+4,33
                                    :rem 24
234 POKE820,K:POKEV+41,.:RETURN    :rem 156
300 F=54272:POKEF+4,.:POKEF+5,128:POKEF+6
    ,128:POKEF+4,17:REM EXTRA BASKET
                                    :rem 45
301 FORA=1TO4:ST(A)=PEEK(62+A):NEXT:BL=BL
    +20*AA:IFBL>872THENBL=872       :rem 17
303 P=P-AA:P=P-(P>.):RESTORE       :rem 251
310 RESTORE:FORA=832TO832+20*(3-P):READM:
    POKEA+1,.:POKEA,M               :rem 18
330 POKEF+1,INT(RND(.)*256):NEXT:FORA=1TO
    4:POKE62+A,ST(A):NEXT:POKEF+1,.:RETUR
    N                               :rem 179
340 POKEF+4,.:POKEF+5,128:POKEF+6,128:POK
    EF+4,33:B=28+INT(RND(.)*8)*32:C=.
                                    :rem 216
345 FORA=CTOB:POKEV+2,A:POKEF+1,AAND10:NE
    XT:POKEF+4,.                    :rem 63
350 POKEF+6,128:POKEF+5,.:POKEF+4,33:POKE
    F+1,.:RETURN                    :rem 189
400 PRINT"{CLR}{6 DOWN}{2}{RVS}{*}
    {8 RIGHT}{CYN} ":PRINT"{2}{RVS}
    {8 RIGHT}{CYN} {4 RIGHT}{6} ":POKE204
    0,15                            :rem 114
410 PRINT"{2}{RVS} {8 RIGHT}{CYN}
    {4 RIGHT}{6} ":POKEV+1,101:POKEV+39,1
    :POKEV,32:POKEV+21,1            :rem 39
420 PRINT"{2}{RVS} {RIGHT} {YEL} {RIGHT}
    {5}£{3 SPACES}{RIGHT} {CYN} £ {*}{6}
    {OFF}{*}{RVS}{2 SPACES}{RIGHT}{PUR}£
    {2 SPACES}{OFF}£{RVS}{RIGHT} {RED}£
    {2 SPACES}{*}{RIGHT}{BLU} {2 RIGHT}
    {RIGHT}{GRN}£ {*}{RIGHT}{3} £{*}":P
    OKEV+29,1                       :rem 26
430 PRINT"{2}{RVS} {RIGHT} {YEL} {RIGHT}
    {5} {2 RIGHT} {RIGHT} {CYN} {OFF}£
    {RVS}{RIGHT} {RIGHT}{6} {2 RIGHT}
    {PUR}{OFF}{*}{@}{2 RIGHT}{RVS}{RIGHT}
    {RED} {2 RIGHT} {RIGHT}{BLU}
    {2 RIGHT} {RIGHT}{GRN} {OFF}{P}{RVS}
```

September 1984 COMPUTE! 95

```
                {SPACE}{RIGHT}£3┐ {OFF}£┌*┐":POKEV+2
        3,.                               :rem 185
440 PRINT"£2┐{RVS} {RIGHT}{YEL} {RIGHT}
        £5┐ {2 RIGHT} {RIGHT}{CYN} {2 RIGHT}
        {SPACE}{RIGHT}£6┐ {RIGHT}£{2 RIGHT}
        {PUR}{OFF}£T┐{RVS}£*┐{RIGHT}{RED}
        {2 RIGHT} {RIGHT}{BLU}{OFF}£*┐{RVS}
        £*┐£{OFF}£{RIGHT}{GRN}
        {3 RIGHT}£3┐ "            :rem 87
450 PRINT"£2┐{RVS} {RIGHT}{YEL}{OFF}£*┐
        {RIGHT}£5┐£*┐{RVS}{3 SPACES}{RIGHT}
        {CYN} {2 RIGHT} {RIGHT}£6┐{OFF}£*┐
        {RVS} {OFF}£{RVS}{PUR}£{2 SPACES}
        {OFF}£{RIGHT}{RED}£*┐{RVS} {OFF}£
        {RVS} {2 RIGHT}{BLU}{OFF}£*┐£
        {2 RIGHT} {RIGHT}{GRN}£*┐{RVS} {OFF}£{RIGHT}
        {RVS}£3┐ "                :rem 18
460 PRINT"£2┐{RVS} {6 RIGHT}£5┐ ":PRINT"
        £2┐{RVS} {3 RIGHT}£5┐{OFF}£*┐{RVS}
        {2 SPACES}{OFF}£":PRINT"£2┐{RVS}
        {4 SPACES}£*┐{3 DOWN}"        :rem 123
470 PRINT"{WHT}{4 SPACES}USE PADDLES IN P
        ORT ONE TO PLAY.":RETURN      :rem 58
500 QW=49164:RW=49517               :rem 172
510 FORAW=.TO22:BW=PEEK(AW+QW):POKEAW+QW,
        PEEK(AW+RW):POKEAW+RW,BW:NEXT:rem 109
520 IFBW=141THENPOKE49262,96:POKE49230,63
        :POKE49378,52:POKE49379,3:GOTO540
                                     :rem 129
530 POKE49262,165:POKE49230,14:POKE49378,
        64:POKE49379,0:POKE49481,252:RETURN
                                     :rem 199
540 POKE49481,255:RETURN             :rem 180
550 IFPEEK(51457)=114THENGOTO560     :rem 123
552 HS$="000000":HU$="C-64":GOSUB590
                                     :rem 170
560 HT$="":FORA=51458TO51467:HT$=HT$+CHR$
        (PEEK(A)):NEXT               :rem 208
562 HS$=LEFT$(HT$,6):HU$=RIGHT$(HT$,4)
                                     :rem 166
564 RETURN                           :rem 127
590 HT$=HS$+HU$+" ":FORA=1TOLEN(HT$):POKE
        51457+A,ASC(MID$(HT$,A,1)):NEXT:RETUR
        N                            :rem 179
600 POKE53269,.:IFPEEK(49164)=141THENGOSU
        B500:IFSL=0THENSYSX:NS$="000000"
                                     :rem 129
605 POKE53272,28:PRINT"{HOME}{12 DOWN}
        {YEL}{6 SPACES}PRESS <F1> TO START...
        "                            :rem 36
607 IFSL=0THENSL=6:GOSUB700          :rem 202
610 GOSUB202:GOSUB340:POKE254,0:POKE820,2
        55:POKE68,0:SYST              :rem 231
620 IFP=2THENP=1:BL=BL+20             :rem 243
630 IFPEEK(68)=1THENGOTO80            :rem 74
640 BL=872:SYSX:POKE54296,0:POKE53269,0:P
        OKE53280,0:GOSUB800:GOTO60     :rem 38
700 RESTORE:B=589:FORA=.TOB:READM:NEXT:FO
        RA=.TO10:READM:J=M*8+12288     :rem 252
710 FORB=.TO7:READM:POKEB+J,M:NEXTB,A:RET
        URN                           :rem 132
800 RESTORE:FORA=832TO895:READM:POKEA,M:N
        EXT:POKE54273,.:RETURN         :rem 98
900 F=54272:FORA=0TO24:POKEF+A,.:NEXT:A$=
        "* HIGH SCORER *":POKE646,6:GOSUB990
                                     :rem 15
910 POKE646,7:POKEF+6,240:POKEF+4,17:POKE
        F+24,143                      :rem 68
920 FORA=1TOLEN(A$):FORB=.TO200STEP10:POK
        EF+1,B:NEXT:PRINTMID$(A$,A,1)"{RIGHT}
```

```
        ";:NEXT                       :rem 85
930 POKEF+24,140:A=1397                :rem 239
940 PRINT,"{13 DOWN}{CYN} TURN PADDLE TO
        {SPACE}CHOOSE LETTERS..{DOWN}"
                                     :rem 253
941 PRINT" PUSH BUTTON TO MOVE CURSOR"
                                     :rem 67
942 PRINT"{HOME}{9 DOWN}£6┐{15 SPACES}£@┐
        £@┐ £@┐":HU$=""               :rem 29
943 POKEA+40,30:POKEA+F+40,10:POKEF+A,1
                                     :rem 221
945 B=29-PEEK(54297)/9:POKEA,B:POKE646,(P
        EEK(646)+1)AND255:GOSUB990     :rem 125
950 POKEF+1,B*8+15:IFPEEK(56321)AND4THENG
        OTO945                        :rem 33
955 POKEF+4,129:FORC=0TO90STEP1.5:POKEF+1
        ,C:NEXT:POKEF+4,17:POKEA+40,32:rem 75
960 A=A+2:HU$=HU$+CHR$(B+64):IFA<1405THEN
        943                           :rem 231
962 PRINT"{4 DOWN}":POKEF+4,0:POKE51457,1
        14                            :rem 81
965 GOSUB590:PRINT"{HOME}":FORA=1TO13
                                     :rem 183
970 PRINT"{32 SPACES}":NEXT:GOTO90
                                     :rem 198
990 PRINT"{HOME}{2 DOWN}{RIGHT}U*********
        ********************I"SPC(9)"B"SPC(29
        )"B"                          :rem 65
995 PRINT"{RIGHT}J*********************
        ******K{HOME}{3 DOWN}{2 RIGHT}";:RETU
        RN                            :rem 191
```


The player just missed a light bulb in the 64 version of "Lightsaver."

6430,7677

Program 2: VIC Lightsaver/MLX

Be sure to read the "Tiny MLX" article before typing.

Version by Gregg Peele, Assistant Programming
Supervisor

Refer to the "Automatic Proofreader" article before typing this
program in.

```
6430 :011,016,000,000,158,052,011
6436 :049,048,057,000,000,000,190
6442 :169,000,141,092,003,141,076
6448 :093,003,169,024,141,015,237
6454 :144,032,078,020,032,100,204
6460 :016,032,236,017,032,042,179
6466 :017,169,007,141,074,003,221
6472 :169,003,141,064,003,173,113
```

www.commodore.ca

```
6478 :076,003,016,024,032,203,176
6484 :017,172,074,003,196,251,029
6490 :208,003,076,077,016,032,246
6496 :015,019,032,246,018,169,083
6502 :010,141,076,003,206,076,102
6508 :003,032,108,017,032,125,169
6514 :016,032,038,019,032,179,174
6520 :019,173,072,003,208,207,034
6526 :076,205,018,169,000,141,223
6532 :060,003,141,061,003,141,029
6538 :074,003,032,042,017,169,219
6544 :003,141,072,003,169,010,030
6550 :141,076,003,096,169,000,123
6556 :141,019,145,169,127,141,130
6562 :034,145,173,008,144,074,228
6568 :074,074,074,073,015,056,022
6574 :233,003,016,002,169,000,085
6580 :141,064,003,168,162,002,208
6586 :185,162,031,208,008,169,181
6592 :032,153,162,031,032,023,113
6598 :017,185,184,031,208,008,063
6604 :169,032,153,184,031,032,037
6610 :023,017,185,206,031,208,112
6616 :008,169,032,153,206,031,047
6622 :032,023,017,200,202,016,200
6628 :213,172,064,003,196,252,104
6634 :240,071,164,252,169,032,138
6640 :162,003,153,184,031,153,158
6646 :206,031,200,202,016,246,123
6652 :172,064,003,169,027,153,072
6658 :184,031,153,206,031,132,227
6664 :252,169,000,153,184,151,149
6670 :153,206,151,200,169,028,153
6676 :153,184,031,153,206,031,010
6682 :169,000,153,184,151,153,068
6688 :206,151,200,169,029,153,172
6694 :184,031,153,206,031,169,044
6700 :000,153,184,151,153,206,123
6706 :151,096,173,082,003,240,027
6712 :003,206,082,003,238,060,136
6718 :003,173,060,003,208,003,000
6724 :238,061,003,162,004,160,184
6730 :016,024,032,240,255,174,047
6736 :060,003,173,061,003,032,156
6742 :205,221,056,173,060,003,036
6748 :237,092,003,133,170,173,132
6754 :061,003,237,093,003,005,244
6760 :170,144,012,173,060,003,154
6766 :141,092,003,173,061,003,071
6772 :141,093,003,162,008,160,171
6778 :016,024,032,240,255,174,095
6784 :092,003,173,093,003,032,012
6790 :205,221,096,169,206,133,140
6796 :167,169,031,133,168,173,213
6802 :141,002,208,251,169,228,121
6808 :133,170,169,031,133,171,191
6814 :169,228,133,180,169,151,164
6820 :133,181,162,021,160,021,074
6826 :177,167,208,012,072,169,207
6832 :032,145,167,104,145,170,171
6838 :169,000,145,180,136,016,060
6844 :237,056,165,167,233,022,044
6850 :133,167,165,168,233,000,036
6856 :133,168,056,165,170,233,101
6862 :022,133,170,165,171,233,076
6868 :000,133,171,056,165,180,149
6874 :233,022,133,180,165,181,108
6880 :233,000,133,181,202,016,221
6886 :193,096,032,148,224,165,064
6892 :142,162,003,010,202,208,195
6898 :252,024,101,142,010,101,104

6904 :142,105,003,133,142,074,079
6910 :074,056,233,012,176,252,033
6916 :105,015,133,251,096,169,005
6922 :128,141,019,145,169,255,099
6928 :141,034,145,169,147,032,172
6934 :210,255,162,011,160,005,057
6940 :024,032,240,255,160,000,227
6946 :185,193,019,032,210,255,160
6952 :200,192,012,208,245,169,042
6958 :014,032,210,255,162,013,220
6964 :160,005,024,032,240,255,000
6970 :160,000,185,205,019,032,147
6976 :210,255,200,192,012,208,117
6982 :245,032,228,255,240,251,041
6988 :201,049,144,247,201,054,204
6994 :176,243,056,233,048,073,143
7000 :255,141,063,003,169,147,098
7006 :032,210,255,174,063,003,063
7012 :024,169,000,105,048,202,136
7018 :208,251,141,082,003,162,185
7024 :003,160,016,024,032,240,075
7030 :255,160,000,185,010,020,236
7036 :032,210,255,200,192,007,252
7042 :208,245,162,006,160,016,159
7048 :024,032,240,255,160,000,079
7054 :185,017,020,032,210,255,093
7060 :200,192,012,208,245,162,143
7066 :010,160,016,024,032,240,124
7072 :255,160,000,185,029,020,041
7078 :032,210,255,200,192,007,038
7084 :208,245,162,013,160,016,208
7090 :024,032,240,255,160,000,121
7096 :185,036,020,032,210,255,154
7102 :200,192,015,208,245,162,188
7108 :022,160,001,024,032,240,163
7114 :255,160,000,185,059,020,113
7120 :032,210,255,200,192,019,092
7126 :208,245,173,017,145,041,019
7132 :016,208,249,160,021,169,019
7138 :032,153,228,031,136,208,246
7144 :248,096,160,002,162,022,154
7150 :024,032,240,255,160,000,181
7156 :185,217,019,032,210,255,138
7162 :200,192,018,208,245,032,121
7168 :228,255,240,251,201,089,240
7174 :240,007,201,078,240,006,010
7180 :076,226,018,076,029,016,197
7186 :000,164,251,140,074,003,138
7192 :162,001,024,032,240,255,226
7198 :160,000,185,235,019,032,149
7204 :210,255,200,192,016,208,093
7210 :245,096,172,074,003,162,026
7216 :001,024,032,240,255,160,248
7222 :000,185,251,019,032,210,239
7228 :255,200,192,014,208,245,150
7234 :096,160,021,185,228,031,019
7240 :208,028,169,036,153,228,126
7246 :031,169,000,153,228,151,042
7252 :072,152,072,032,142,019,061
7258 :032,179,019,104,168,104,184
7264 :206,072,003,076,076,019,036
7270 :136,208,220,169,032,153,252
7276 :228,031,162,011,160,016,204
7282 :024,032,240,255,173,082,152
7288 :003,205,084,003,240,013,156
7294 :160,000,185,051,020,032,062
7300 :210,255,200,192,008,208,181
7306 :245,173,082,003,141,084,098
7312 :003,056,073,255,170,169,102
7318 :000,032,205,221,162,015,017
```