COMPUTE! Interviews Gerrard O'Neill

COMPUTE!

$2.95
August
1984
Issue 51
Vol. 6, No. 8
£2.25 UK $3.75 Canada
02193
ISSN0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

The Consumer Electronics Show: New Excitement In Home Computing

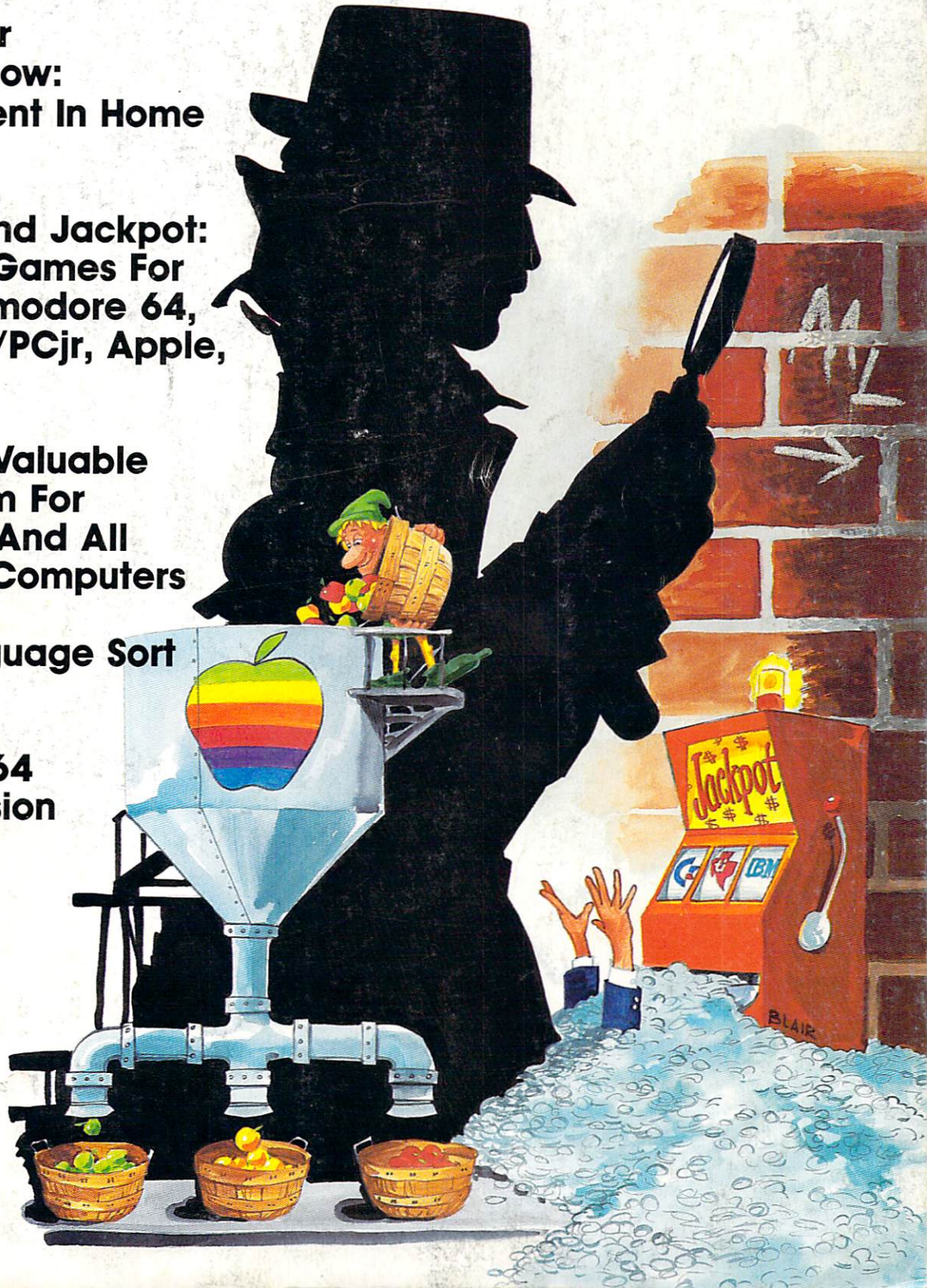
Devastator And Jackpot: Two Exciting Games For VIC-20, Commodore 64, Atari, IBM PC/PCjr, Apple, And TI-99/4A

ML Tracer: A Valuable Utility Program For Atari, Apple, And All Commodore Computers

Machine Language Sort For Apple

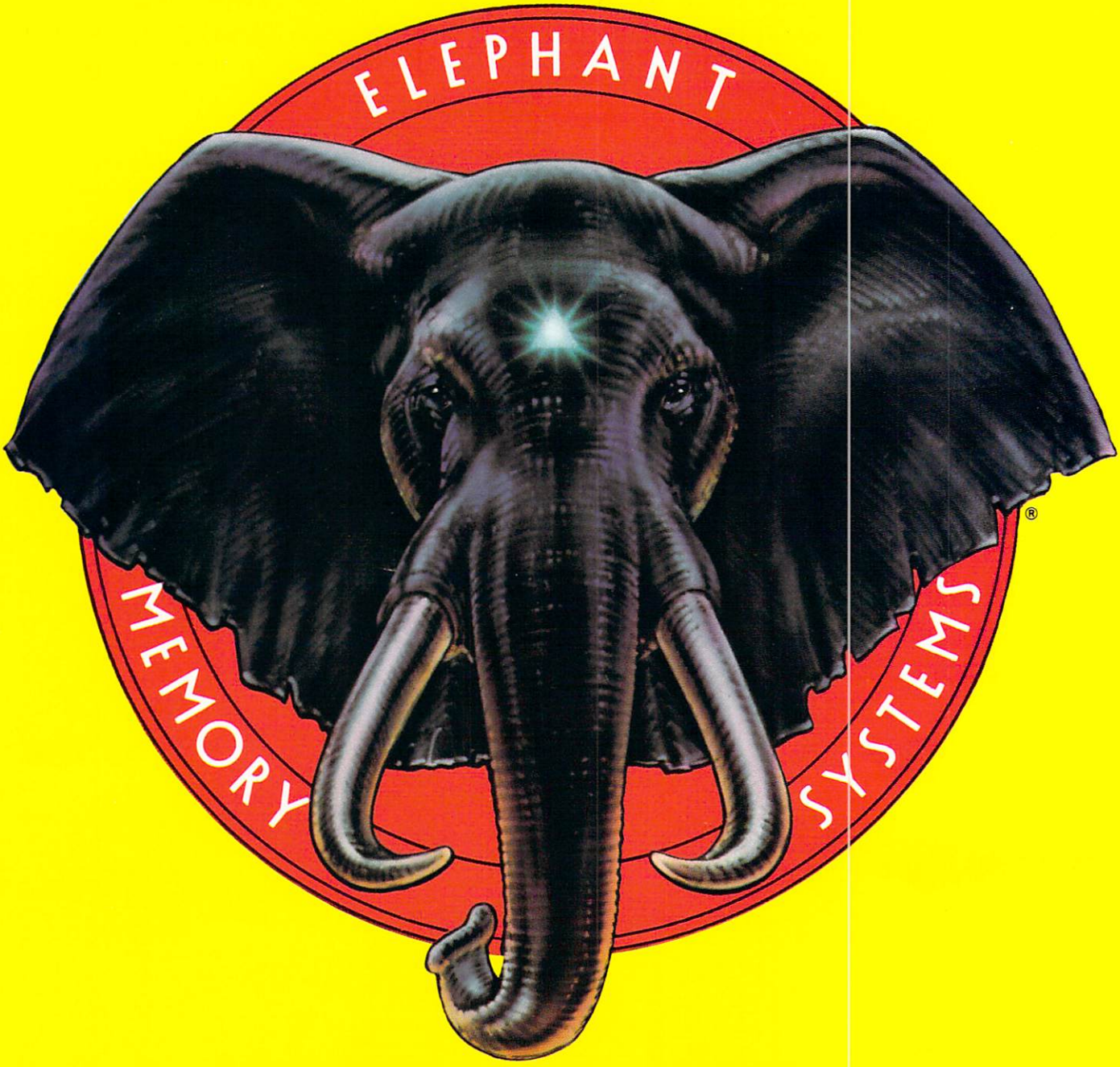
Commodore 64 Error Suppression

And More



08
71486 02193

REMEMBER.



ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a lifetime of heavy-duty use).

Contact Dennison Computer Supplies, Inc., 55 Providence Highway, Norwood, MA 02062 or call toll-free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150. Telex 951-624.

Dennison

WordPro 3 Plus™/64

The #1 Selling Word Processor for the Commodore 64™

Now Includes SpellRight Plus! Spell Checking Program

WordPro 3 Plus™/64 and SpellRight Plus™ provide a total word processing solution for the Commodore 64™ which gives you:

- ★ Sophisticated Word Processing
- ★ Built-in Mail Merging for Form Letters
- ★ Math Functions for Column Totals
- ★ Fast and Complete Spell Checking via SpellRight Plus
- ★ A Super Value (two programs) for Only $99.95!

WordPro and SpellRight are both specifically designed for the novice user with no computer or word processing experience whatsoever. And with over 40,000 WordPro versions sold, you can be sure that WordPro is a very sophisticated word processor loaded with powerful features including: Transfer, Insert, Delete, and Rearrange Text, Auto Page Numbering, Math Functions, Headers, Footers, Global Search and Replace, the Ability to Create Multiple Personalized Letters and Documents, and much more. WordPro can create documents of virtually any length and will print up to 165 columns wide. You get all of this PLUS fast and complete spell checking using SpellRight Plus!

SpellRight Plus locates and highlights misspelled words and then allows you to quickly correct the misspellings — improving the quality of your letters and reports.

And, best of all, WordPro and SpellRight's powerful arsenal of features can be put to use almost immediately — by even the novice user. So whether you're a student, professional writer, in business, education or a hobbyist, you'll quickly become a WordPro Pro!

Both WordPro and SpellRight Plus are also available separately at popular computer outlets nationwide.

Invest in the best . . .WordPro Plus. In a class by itself.

Professional Software Inc.

51 Fremont Street
Needham, MA 02194

(617) 444-5224
Telex: 951579

Dealer and Distributor inquiries are invited.

IBM PC Software: the value of choosing

Shoes.
If they don't fit, they're not worth wearing.
Software programs.
If they don't fit, they're not worth using.
That's why it's altogether fitting that IBM Personal Computer Software offers you a choice.

Size up the selection.

You'll find many types of programs in the IBM software library. They'll help keep you on your toes in the office, at home or in school.

There are, in fact, seven different categories of IBM programs called "families." A family of software for business, productivity, education, entertainment, lifestyle, communications or programming.

Of course, every program in every family is tested and approved by IBM. And IBM Personal Computer Software is made to be compatible with IBM Personal Computer hardware.

programs that fit.

Putting your best foot forward.

Although every person isn't on equal footing when it comes to using personal computer software, there's something for almost everyone in the IBM software library.

For example, you may be on a shoestring budget and want a big selection of programs with small price tags.

You may be introducing students to computing and want programs that are simple to use and simple to learn.

You may run a business requiring sophisticated inventory and payroll programs. Or you may run a business requiring a single accounting program.

You may write interoffice memos and want a streamlined word processing program. Or you may be a novelist looking for a program with features worth writing home about.

Now you can find IBM Personal Computer Software that fits — to help you accomplish specific tasks and reach individual goals.

Stroll into a store today.

What's the next step?

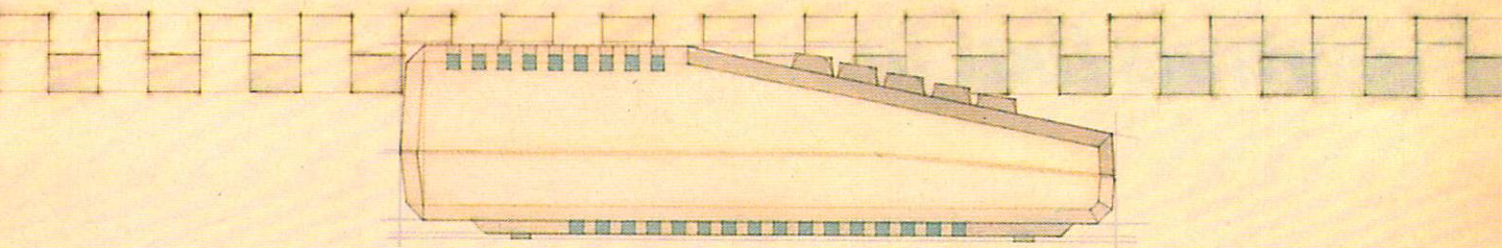
Visit an authorized IBM Personal Computer dealer or IBM Product Center near you. To find out exactly where, call 800-447-4700. In Alaska or Hawaii, 800-447-0890.

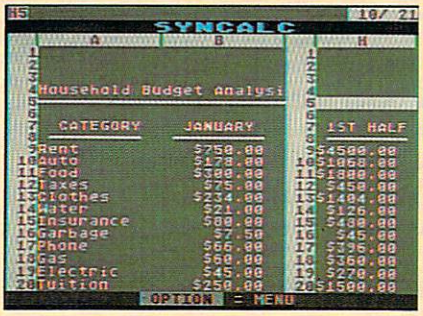
Ask your dealer to demonstrate your choice of programs. Then get comfortable. Sit down at the keyboard and try IBM software on for size.

IBM®

Personal Computer Software

Little Tramp character licensed by Bubbles Inc., s.a.



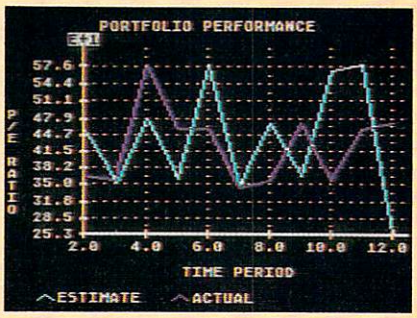


Advanced spreadsheet

SynCalc™

Database management

SynFile+™



Graphing and statistics

SynTrend™

Now your Atari computer gets down to business.

If you're a serious home manager, a student, or run a small business at home, now you can get sophisticated, integrated software for your ATARI computer with the same features as the more expensive IBM and Apple packages.

SynCalc makes a spreadsheet more manageable.

First, there's SynCalc, the most advanced spreadsheet program ever created for ATARI Home Computers. Not only does SynCalc help you get all your numbers down easily, it also comes with a sorting feature, and the ability to label and name your formulas like "beginning inventory + goods purchased−goods sold = inventory on hand," as well as standard entries. And SynCalc also comes with "expert" commands, to use once you've become more familiar with its procedures. Plus many other features found in the more expensive programs.

SynFile+ keeps information more organized.

SynFile+ can function as your database, your filing system. With SynFile+, you can reorganize and sort parts or whole files instantly. Not only can you enter text, you can calculate and update data as well. And files from both SynCalc and SynFile+ can also be used by the ATARI word processor, AtariWriter,™ for uses such as mail merge.

SynTrend gives you a more graphic way to look at data.

Next, there's SynTrend, which can be the graphing and statistical arm of your operation. SynTrend allows you to visualize your data from SynCalc or SynFile+ with either bar graphs, pie charts, line graphs or scatter plots. To do statistical analysis, you can quickly calculate means and variances, standard deviations, or even linear and multiple regressions. It's pretty easy to understand, eh? And also pretty easy to operate because all three programs come replete with easy-to-understand "pop-up" menus, to take you through their paces step by step. And remember, all three programs can share data, which helps you get the job done even faster.

So get down to business with SynCalc, SynFile+, SynTrend, developed exclusively for ATARI by Synapse. And see for yourself why the cost of taking care of business doesn't have to put you out of it.

DISCOVER WHAT YOU AND ATARI® CAN DO.

August 1984 Vol. 6, No. 8

FEATURES

EDUCATION AND RECREATION

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 57 before typing in programs.

GUIDE TO ARTICLES AND PROGRAMS

AP Apple **AT** Atari, **P** PET/CBM, **V** VIC-20, **C** Radio Shack Color Computer, **64** Commodore 64, **TS** Timex/Sinclair, **TI** Texas Instruments, **PCjr** IBM PCjr, **PC** IBM PC, **AD** Coleco Adam, *All or several of the above.

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

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies

One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. Send subscription orders or change of address (P.O. form 3579) to **COMPUTE!** Magazine, P.O. Box 914, Farmingdale, NY 11737. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X

EDITOR'S NOTES

I was unable to attend this summer's Consumer Electronics Show, and in deference to its importance, asked Selby Bateman, our Features Editor, to contribute a guest editorial.

Robert C. Lock
Editor In Chief

The old Chinese curse "May you live in interesting times" often seems to have been aimed directly at our present high-tech, microprocessor-based era.

At least that may have been the feeling for many of the 98,271 attendees who shuffled and stared their way through June's four-day Consumer Electronics Show in Chicago. More than 50,000 electronics retailers and over 2000 members of the press were among that number, each of them trying to comprehend the overwhelming quantity of new products being offered to the American and world markets.

Almost 1400 different exhibitors filled 811,000 square feet of space, displaying the latest stereo TV receivers, new-generation digital audio disc players, cellular telephones, color televisions that fit in the palm of your hand, videocassettes, car stereos, and—of course—computers, software, and hardware peripherals.

One of the clearest trends evident at CES was that computers are becoming linked more closely with almost every other consumer electronics product exhibited. In the not too distant future, the fairly clear-cut lines between computers, stereos, telephones, video systems, and many other products will disappear. This will become even more apparent by the beginning of 1985, with the arrival in quantity here of new MSX operating system micros from Japan.

One example of this trend: Atari chairman James Morgan, in his efforts to bring his company into the future with brighter prospects, emphasizes that Atari's goal isn't just to produce computers, but to "enhance consumers' lives through interactive electronics." That sentiment is being echoed in different words by many other electronics' manufacturers. They see their products getting "smarter," as everything from washing machines to automobiles begins to carry microprocessors.

Interesting changes in microcomputer hardware and software were everywhere at CES. While the great majority of the public attempts to understand microcomputer developments that are essentially several years old, the industry charges forward at a gallop. Even for those who stay abreast of the latest news from the high-tech front lines, the power and the pace of change in this industry are often bewildering.

How can an individual learn about and digest all of the innovations, new products, changing technologies, and scattered trends that take place in the computer and electronics field on a daily basis? More importantly, how can those changes be understood, wisely interpreted, and selectively used?

Although we're biased on the subject, it seems obvious that those who have found an interest in—sometimes a passion for—our remarkable computer revolution may be in a better position to understand and take advantage of what Eric Hoffer called the wrenching "ordeal of change."

One model for us is the subject of this month's COMPUTE! Interview, physicist Gerard O'Neill of Princeton. Throughout his career as a scientist, writer, lecturer, and entrepreneur, O'Neill has consistently blended an ability to understand society's changes with a clear vision of how things can and should work. His books and his interests reflect a mix of the hard sciences, human values, visionary ideas, and an unquenchable, optimistic curiosity.

His interests are eclectic—from developing colonies in space to piloting glider planes to researching high-energy physics to working with his Apple II+ computer. Perhaps it is O'Neill's curiosity and his practical optimism which are fundamental to his highly successful approach to the whirlwind of technological change. Importantly, those seem to be characteristics which our readers and many of those who are intrigued by computing appear to have in abundance.

Is it really a curse or a blessing to live in interesting times? Samuel Clemens once remarked that anyone who has held a bull by the tail knows five or six things more than someone who hasn't. So enjoy the mixed blessings of the microcomputer revolution, and the fact that you know five or six things more than you did before.

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

Coming In Future Issues

Choosing The Best Educational Software

Two Exciting Games For Several Computers: Lightsaver And Missile Math

Commodore 64 SYSound

Atari Speed-Reading

TI Screen Dump

Update On Commodore COMAL

And Much More....

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

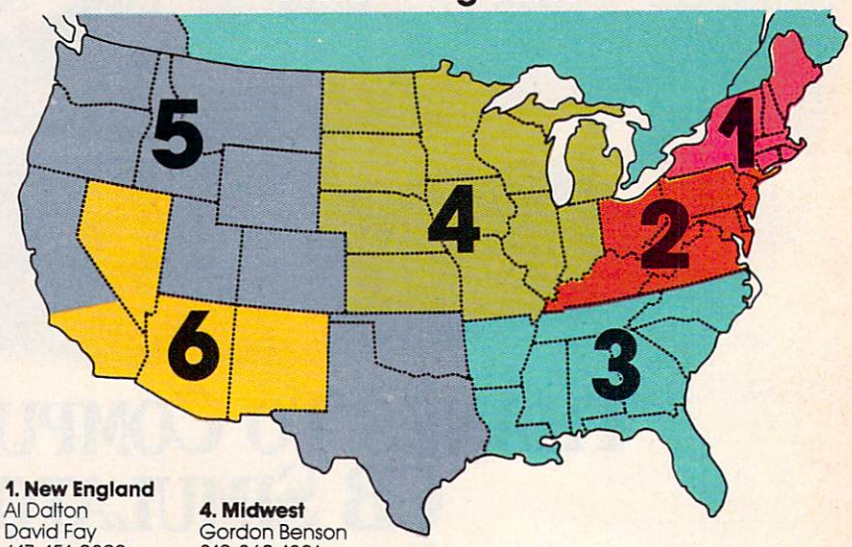
COMPUTE! Circulation Dept.
P.O. Box 914
Farmingdale, NY 11737

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

COMPUTE! Subscription Rates (12 Issue Year):

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

Flight Simulator II

For Apple, Atari, & Commodore 64

Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.

See your dealer . . .

or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800/637-4983

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

How Much Commodore 64 Memory?

I have a Commodore 64. How do I determine how much memory a program occupies? I cannot find this information in either the 64 *User's Manual* or the *Programmer's Reference Guide.*

Donald E. Lassiter, Jr.

The amount of BASIC memory available on the 64 is 38,911 bytes. You will see this message when you first turn the 64 on. To determine how much memory is left free (unused), type and enter:

```
PRINT FRE(0)-65536*(FRE(0)<0)
```

or

```
PRINT FRE(0)+2↑16
```

To determine how much memory a program is using, subtract the value received using the formula above from 38911, or type and enter:

```
PRINT 38911-(FRE(0)-65536*(FRE(0)<0))
```

or

```
PRINT 38911-(FRE(0)+2↑16)
```

Apple Pascal

I am interested in learning more about Apple's Pascal operating system. Is it software that needs to be loaded from disk or on a card that needs to be installed or what?

Mirim Lew

Apple's Pascal operating system is a version of the UCSD Pascal system, written at the University of California at San Diego. It is supplied on disk and loaded into RAM, where it is used in place of the normal Applesoft ROM. This disk is all that is needed on newer Apples (IIe's and IIc's). Apple II's and II+'s don't contain the extra RAM needed to hold Pascal and require the Apple Language Card—a special 16K memory card which plugs into interface slot zero.

Atari Checksum Errors

I have had my Atari computer for a year now, but I still have a few unanswered questions you might be able to help me with. Sometimes when I load a program off my cassette, I get something called a "serial bus data frame checksum error." What does this mean, and how can I remedy it? Also, when I get an error in the middle of the loading process, is there any way I can retrieve the portion that did load correctly? And is there a way to verify Atari SAVEs?

Jeff McCain

The "serial bus data frame checksum error" and its cousin, "serial data frame overrun," are just Atari's way of telling you that the computer encountered a tape error. The tape drive is very sensitive to errors in timing—if a tape is stretched in the middle, it will throw off the bit timing. You can also get this error if you didn't allow enough leader when you positioned the tape for CLOAD.

An incomplete program can be a major problem. Due to the way Atari programs are stored, BASIC must know how to find the exact end of a program. A partial program is often cut off in the middle of a line, and when BASIC scans to find the end of the program, it locks up, not finding it. So to prevent this problem, a faulty LOAD causes the partial program to be NEWed. If you store programs with LIST "C:", you can then ENTER "C:" to retrieve it. If there is an error, you will still be left with a partial program. You can continue with ENTER "C:", and you may pick up more and more of the listing.

ENTER can also be used to verify a LISTed program. If you ENTER a program that you have just LISTed, the program in memory will not be lost. If it ENTERs without an error, you've verified that the data is stored correctly. Otherwise, you'll still be left with the program in memory, so you can try again. LIST and ENTER, though, are slower and use more tape space than CSAVE and CLOAD.

Pushing And Pulling The Stack

When programming in BASIC or machine language, how does pushing and pulling things on the stack affect the return jump?

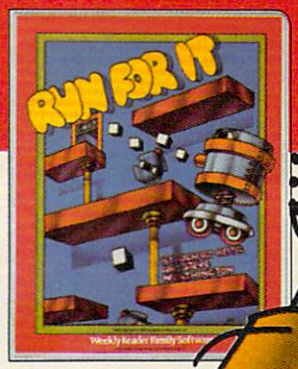
Thomas McCrossin

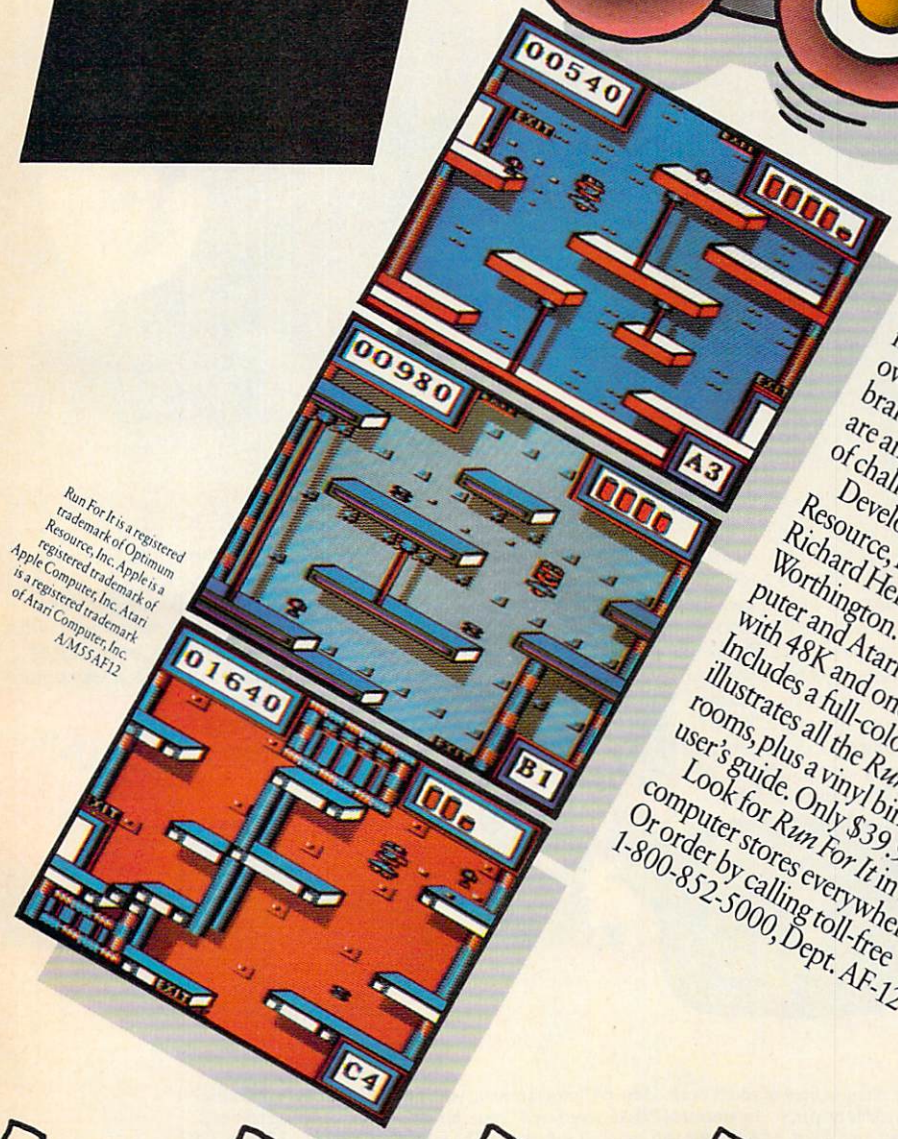
The stack is an area of 256 RAM memory bytes that is used to hold return addresses for BASIC GOSUBs and machine language JSRs.

When a GOSUB is encountered while running a BASIC program, the following happens:

1. In the simplest terms, the memory address of the next executable statement following the

Run For It™



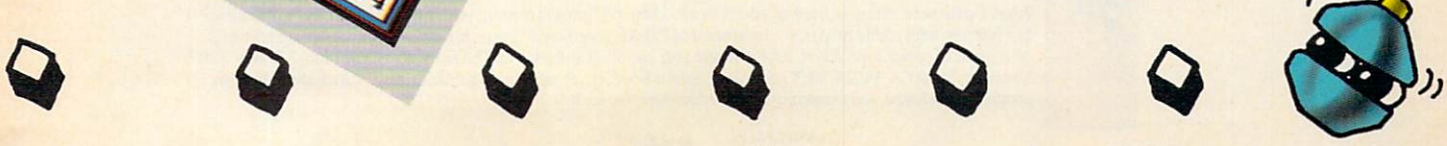


Can you help ORBIT the Robot flee from room to mazelike room in his daring, thrill-a-second effort to escape the bad guys? With *Run For It*'s amazing 3-D effect, you'll want to step right inside the screen to make sure ORBIT reaches the end of his dangerous journey! You can make ORBIT bounce up or scrunch down as he climbs from level to level in order to avoid his enemies—and fires powerful missiles of his own! The special effects on this brand-new arcade-like game are amazing—and a new kind of challenge for game players.

Developed by Optimum Resource, Inc. Designed by Richard Hefter and Steve Worthington. For Apple® Computer and Atari® Computers with 48K and one disk drive. Includes a full-color poster that illustrates all the *Run For It* rooms, plus a vinyl binder and user's guide. Only $39.95 each.

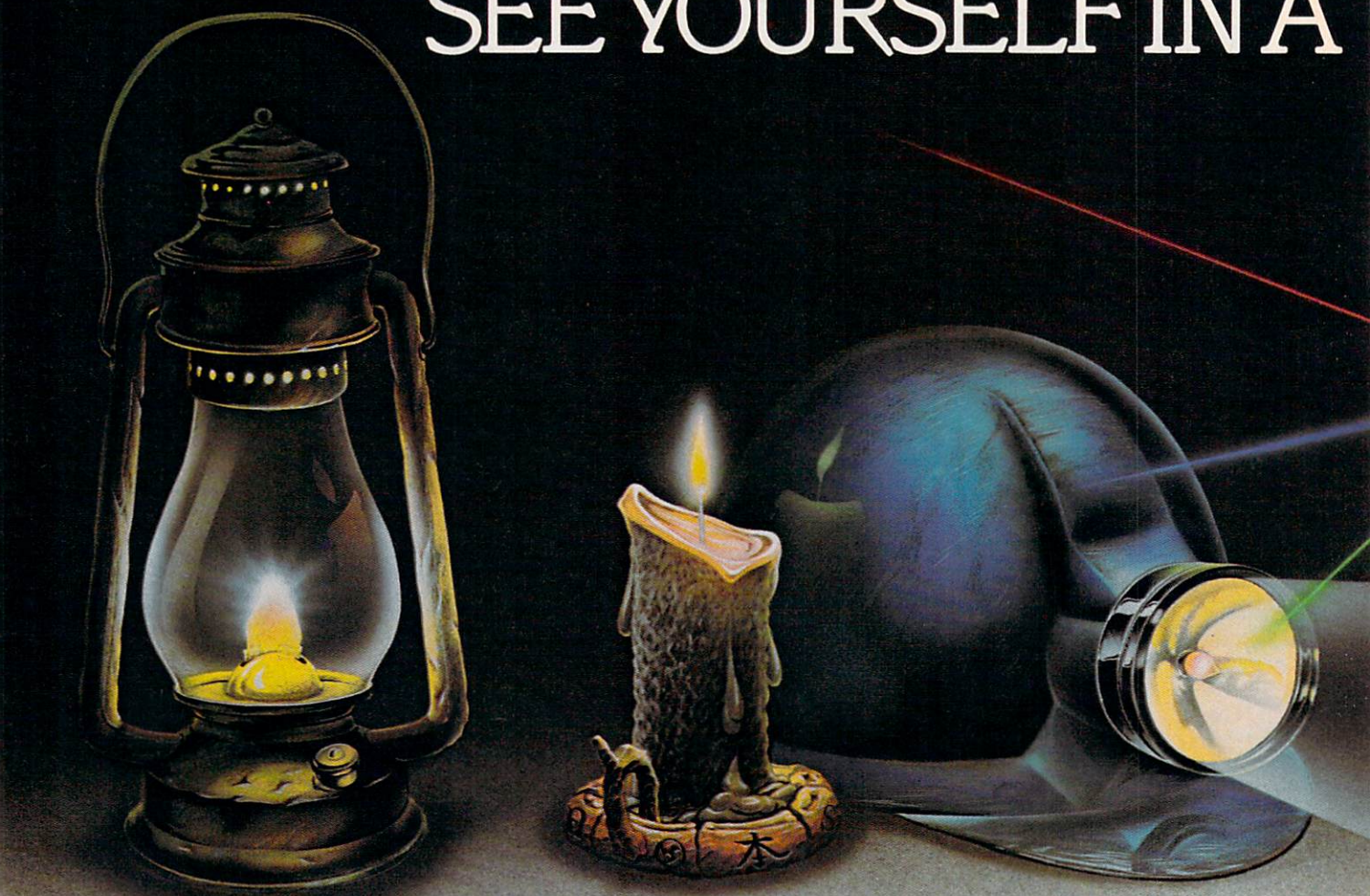
Look for *Run For It* in finer computer stores everywhere. Or order by calling toll-free 1-800-852-5000, Dept. AF-12

Run For It™
Weekly Reader Family Software
A division of Xerox Education Publications
Middletown, CT 06457



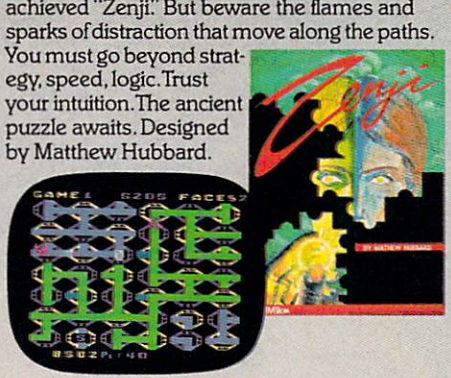
INTRODUCING ACTIVISION

SEE YOURSELF IN A



You leave the sun behind as you lower yourself down into the unexplored caverns beneath the Peruvian jungle. Deeper and deeper you go. Past Amazon frogs, condors, and attacking bats. Across eel-infested underground rivers. From cavern to cavern, level to level. Swimming, running, dodging, stumbling, you search for the gold, the Raj diamond and the thing you really treasure ... adventure. Head for it. Designed by David Crane.

You have heard the elder speak of one central source and a maze of unconnected grey paths. As you connect each grey path to the central source, what was grey becomes the green of life. When all are connected, then you have achieved "Zenji." But beware the flames and sparks of distraction that move along the paths. You must go beyond strategy, speed, logic. Trust your intuition. The ancient puzzle awaits. Designed by Matthew Hubbard.



You strap on your helicopter prop-pack, check your laser helmet and dynamite. There's no predicting what you'll have to go through to get to the trapped miners. Blocked shafts, molten lava, animals, insects, who knows what lies below. But you'll go, you're in charge of the Helicopter Emergency Rescue Operation. The miners have only one chance. You. The opening shaft is cleared now, it's time to go. Designed by John Van Ryzin.

What if you were sitting in front of your Commodore 64™ programming your own Pitfall Harry™ adventure? It can happen with a little help from the creator of Pitfall Harry: David Crane. Just write your name and address on a piece of paper, tape 25¢ to it for postage and handling and mail to: The Activision C-64 Club, P.O. Box 7287, Mountain View, CA 94039. We'll send you David's Booklet, "Programming Pitfall Harry." It includes a written program that helps you create your own adventure. Go for it.

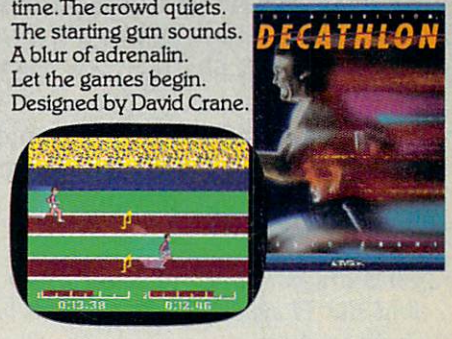
FOR YOUR COMMODORE 64.

DIFFERENT LIGHT.

As you suit up you see the webbed forcefield surrounding your planet. Holding it. Trapped with no escape. No hope. Except you: The Beamrider. The freedom of millions depends on you. Alone you speed along the grid of beams that strangle your planet. You must destroy the grid sector by sector. Your skills and your reflexes alone will determine the future of your people. Take their future in your hands. Designed by Dave Rolfe.

You can almost hear the quiet. And it's your job to keep it that way. A toy factory at midnight. Did you hear something? Guess not. Wrong! Suddenly balloon valves open, conveyor belts move and a whole factory full of toys goes wild. Even the robot, their latest development, is on the loose and after you. Capture the runaway toys. Restore order. Restore peace. Restore quiet. Do something! Hurry! Designed by Mark Turmell.

You made it. The Olympics. You hear languages you've never heard. And the universal roar of the crowd. You will run. Hurl. Vault. Jump. Ten events. One chance. You will push yourself this time. Further than ever. Harder than ever. But then... so will everyone. The competition increases, now two can compete at the same time. The crowd quiets. The starting gun sounds. A blur of adrenalin. Let the games begin. Designed by David Crane.



ACTIVISION.

We put you in the game.

GOSUB is pushed onto the stack. (Specifically, it's the address minus one byte.)

2. The branch to the subroutine is taken, and the subroutine is executed.

3. When the RETURN is encountered in the subroutine, the return address information is pulled off the stack, program control is returned to that point, and processing continues.

Basically, the same sequence of events is taken when using machine language. When a JSR is encountered, the return information is pushed onto the stack, the branch is taken, and when the RTS (ReTurn from Subroutine) is encountered, the information is pulled from the stack, and control is returned to that place in the program.

When using GOSUBs and JSRs, this stack activity is automatically performed by the computer.

However, you can push and pull stack information yourself. This can be done with the use of the PHA machine language instruction, which pushes the number in the Accumulator onto the stack, and PLA, which pulls a byte off the stack and places it in the Accumulator. Other stack commands available are PHP, which pushes the processor status onto the stack, and PLP, which pulls a byte from the stack and puts it into the status register.

Manipulating the stack can be tricky. However, if, after jumping to a subroutine, you wish to return somewhere else, you can pull the return information off the stack (placed there by the operating system), and replace it with your data using the PHA command.

The stack can also be used as a temporary storage place for data in machine language programming. Instead of storing information in zero page, or some other area, push it onto the stack. When it's needed again, pull it back off. But be careful, because the stack can hold only 256 bytes of information. Also if you RTS before PLAing the byte or bytes off the stack, the return address will be wrong.

TI Programs Vs. Data Files

I read somewhere that if a TI-99/4A program sets up a data file, the data file should be stored on a separate disk or cassette from the program. Why is that? It seems to me that the logical place for the data would be on the same disk or cassette as the program using it.

Florence Fischer

Files are not saved or loaded by name on a cassette, and the TI makes no distinction between data and program files. As a result, if you place a data file on a tape following a program file, you may have difficulty locating the data file (especially if your recorder lacks a moderately accurate counter). Also, if you place the data file prior to the program file on the tape, and later expand your data file, you may end up writing over the program file.

For these reasons, it is wise to keep your program and data files on separate cassettes (or on opposite sides of a single cassette). No such problems exist for disk files since programs are stored by name and are labeled as program, data, etc., on the disk.

Slowing Things Down On VIC, 64, Or PET/CBM

I found something very interesting while experimenting with my 64. While listing a program, I noticed that if you press the CTRL key the listing will slow down. Does this work on all Commodore computers? Is it supposed to do this?

Mike Merriman

Yes, it is. Pressing the CTRL key on the Commodore 64 or the VIC-20 will slow the listings, and some BASIC programs. On the older CBM (Commodore Business Machines) computers like the 8032, the PET, etc., pressing the 1 key will do the same thing. This is to allow you to read the listings more easily as they scroll by.

To see how this affects a BASIC program, type, enter, and RUN the following program. While it is running, press the CTRL key and see what happens:

```
10 PRINT"A":GOTO10
```

Z80 Atari XL?

I have an Atari 800 and I am thinking about moving on to a more sophisticated system like the Atari 1450XLD. I have heard that the 600XL and 800XL are much like the older 400/800 models, but how about the 1400XL and 1450XLD? Is the BASIC language different? I heard it has a Z80 microprocessor. Is all this true?

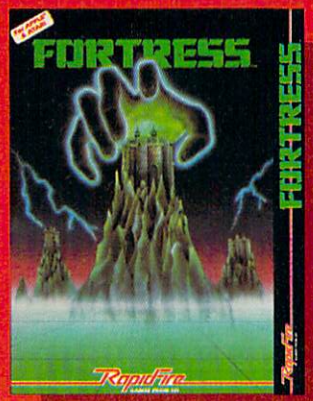
Alekos Couloumbis

The 600XL and 800XL computers are very much like the 400 and 800. The 600XL and 800XL are almost identical, except that the 600XL has 16K while the 800XL has 64K. There have been some enhancements to the operating system of the XL computers, making it different enough so that some 400/800 programs will not run on the XL computers. However, Atari has a Translator *disk available through its Customer Service that allows you to run 400/800 programs on your XL computer. The BASIC in all XL computers (except the late 1200XL) is built-in, and almost identical to the earlier Atari BASIC, except that the infamous keyboard lockup has been fixed and the exponentiation function has been improved.*

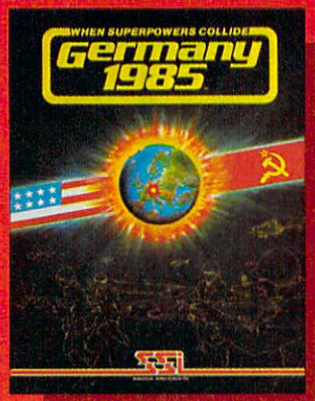
The Atari 600XL and 800XL are now in full

FROM SSI, MORE OF THE BEST STRATEGY GAMES FOR YOUR

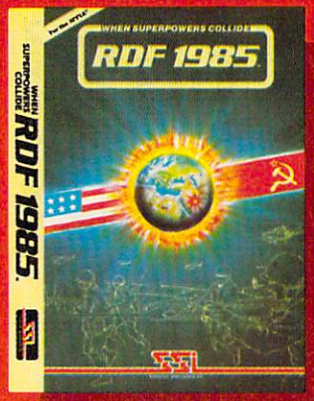
C O M M O D O R E 6 4™

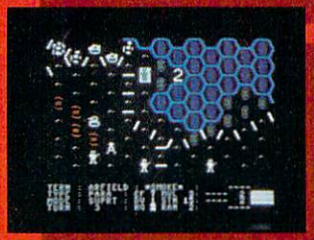


A simple and fast strategy game of power and conquest that's right up there with the classics such as chess and *go*. When you play the computer, it actually learns from you and improves with time to become the ultimate opponent. On disk. $34.95.

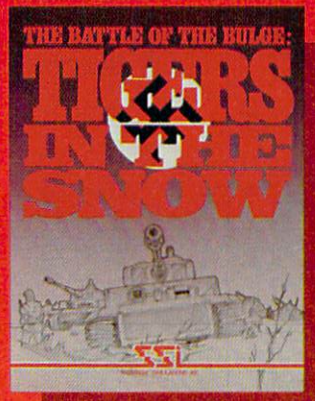


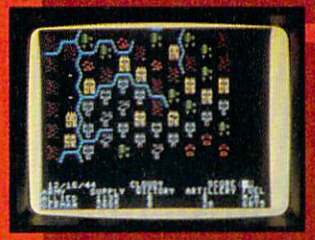
The first game in the series. When Superpowers Collide, our hypothetical projection of possible U.S.-Russia confrontations. GERMANY 1985™ simulates an invasion of West Germany by Soviet battalions. NATO forces must repel this onslaught. On disk. $59.95.



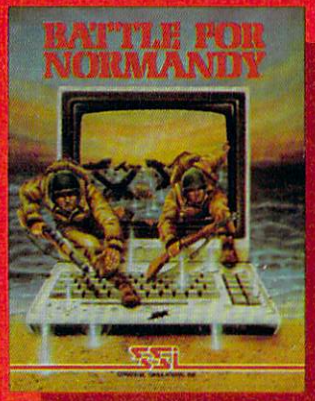


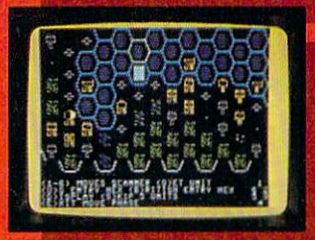
This second game in the series "When Superpowers Collide" takes you to the desert oilfields of Saudi Arabia. Soviet troops have captured these vital points, and it is up to the U.S. Rapid Deployment Force to regain them. On disk. $34.95.





As the German Commander in this re-creation of the Battle of the Bulge, you must use your deadly Tiger tanks to break through the Allied lines in the snow-covered Ardennes Forest. As the Allied leader, you must stop this German blitzkrieg. On disk & cassette. $39.95.





This game takes you to the northern coast of France for the start of *D-Day*! As the Allied Commander, you must see to it that the invasion succeeds. As commander of the German forces, you must not let history repeat itself! On disk & cassette. $39.95.

Commodore 64 is a trademark of Commodore Electronics, Ltd.

Screen displays are not necessarily from the Commodore 64™.

SSI

STRATEGIC SIMULATIONS INC

If there are no convenient stores near you, VISA & Mastercard holders can order direct by calling **800-227-1617, ext. 335 (toll free)**. In California, call 800-772-3545, ext. 335.

To order by mail, send your check to: SSI, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. Include $2.00 for shipping & handling. (California residents, add 6.5% sales tax.) All our games carry a "14-day satisfaction or your money back" guarantee.

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

production. The 1400XL will not be marketed, and the future of the 1450XLD is still somewhat uncertain.

The 1450XLD has a bigger keyboard than the 600XL and 800XL, with extra function keys. It also boasts a built-in speech synthesizer and direct-connect modem. The 1450XLD has a built-in, double-sided, double-density, high-speed disk drive.

The 1450XLD does not have a Z80 coprocessor. At the June 1983 CES, Atari showed a CP/M Z-80 interface. Then at the January 1984 CES, Atari said that they would not market it, but that a third-party company might. Rumors continue to swirl around the 1450XLD, and Atari may well market a slightly different 1450XLD than was shown at recent shows.

Confusing POKEs

When typing in your programs, I have sometimes come upon statements involving POKEing in the PEEK of the same POKE. I would like to know how this works. I also don't understand POKEing the PEEK while ANDing/ORing another number such as:

```
10 POKE 12345, PEEK(12345) OR 67
```

To add to my confusion would be an IF PEEK statement THEN command, that is:

```
20 IF PEEK(12345) THEN GOTO 67
```

or:

```
30 IF 9 AND PEEK(12345) THEN GOTO 67
```

Can you explain this for me?

Dwight Weese

Usually, when you see a POKE and a PEEK to and from the same byte, it's done in conjunction with an AND or OR command.

Each byte is composed of eight bits. Each of the eight bits is like a light switch: It can be either on (1) or off (0). Each of the eight bits has its own value (see illustration below).

Bit	7	6	5	4	3	2	1	0
Value	128	64	32	16	8	4	2	1

Any byte can hold a value from 0 to 255. The value of the byte is determined by which bits are on or off, and is derived by adding the values of each of the on bits. This is called a binary number. *For example, a byte containing a value of 1 would look like:*

00000001 (only the 1 bit is on)

A decimal eleven would look like 00001011 (the 8 plus 2 plus 1 bits are on, so: 8+2+1 = 11), and a byte whose value is 255 would be 11111111 (128+64+32+16+8+4+2+1 = 255).

When you use the AND and OR commands, it's like placing a mask over the eight bits in the byte. The mask is, in effect, placed over the byte, and the bits in the byte are turned on or off according to the rules governing AND and OR. The mask can be thought of as an imaginary byte with bits set or not depending on the mask number.

The bit pattern of the masking byte follows the same rules of binary numbers described above. For example, if you're ANDing or ORing with a value of 21, the bit pattern of the mask would look like 00010101 (16+4+1 = 21).

When you AND a byte, you compare each bit of the byte with each bit of the mask. The result will be an "on bit" only if both *bits (of the byte and the mask) are "on" in that position.*

For example, ANDing a byte with a value of eleven (00001011) with a three (00000011) would result in 00000011 (three). This is because the 1 and 2 bits of both bytes were 1's, but the 8 bit in the mask was a 0.

ORing a byte compares each bit in the same manner as the AND. But when you OR, if the bit in either *the original or the mask byte is a one, then the result is a one ("on").*

For example, ORing a byte containing a value of 15 (00001111) with 240 (11110000) would result in 11111111 (255). This is because in all cases of the compare, at least one of the bits was a 1.

In your other examples, IF PEEK (12345) THEN GOTO 67 *is a standard IF-THEN compare. However, in this case there is no comparing expression as in:* IF PEEK (12345) =1 THEN GOTO 67. *When compares are done in this manner, it is a special kind of test. In this example, the branch to 67 would be taken if the PEEKed number is anything other than zero. If the number is a zero, the test fails and there will be no GOTO.*

The third example, IF 9 AND PEEK (12345) THEN GOTO 67, *is the same. In such cases, all numbers will answer "yes" to the IF, except zero, which will answer "no." The* IF 9 AND PEEK(12345) *expression will be zero in those cases when the 1 and 8 bits of location 12345 are both zero.*

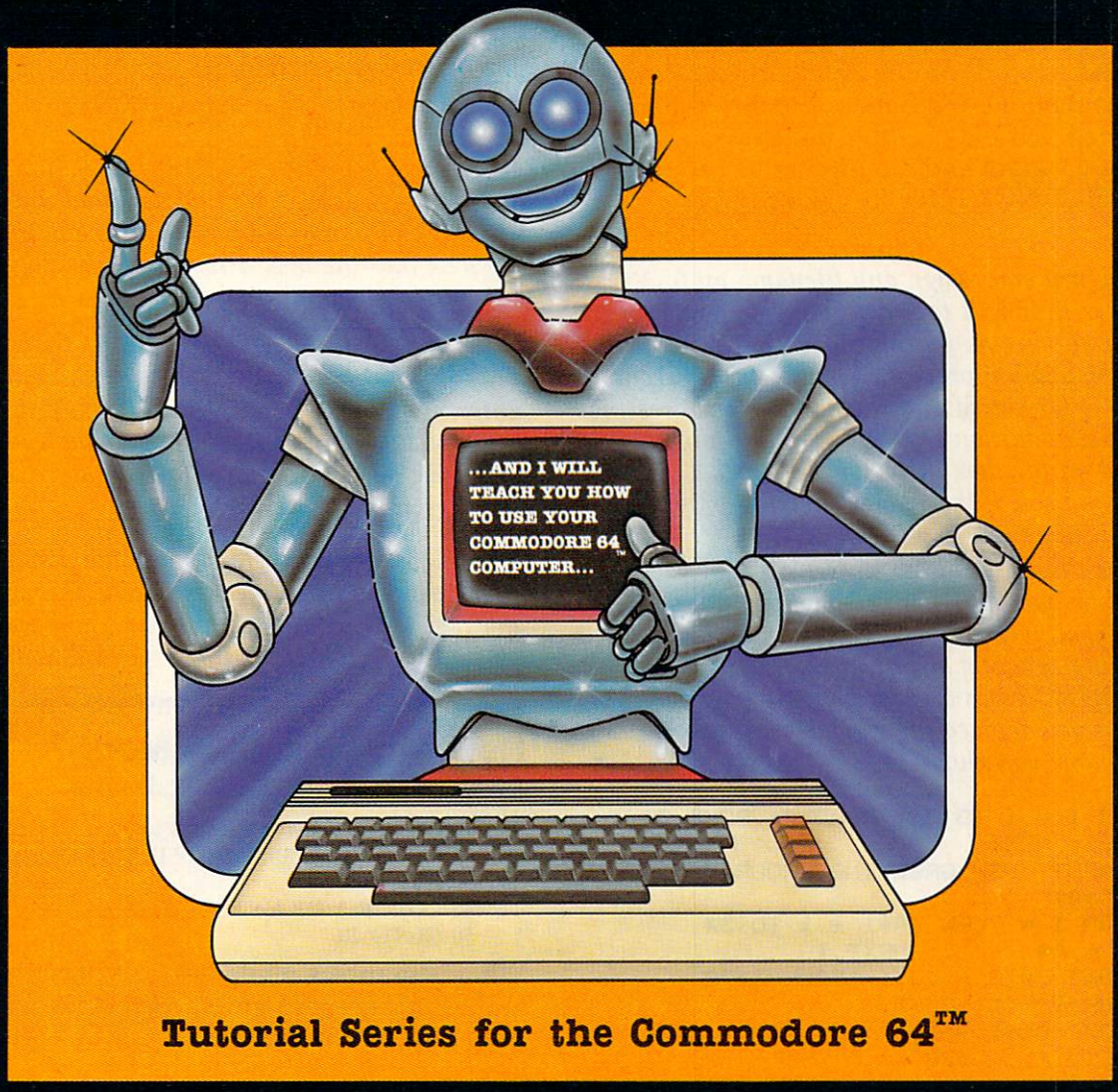
Rainbows For Atari

I own an Atari 800 and I am perplexed about how to access all the colors in graphics mode 8. The only graphics color I can get is white. How do you access the other colors?

Rod MacPherson

GRAPHICS 8, like GRAPHICS 0, is a two-color mode. You can display a character or a pixel in either the background color or a different shade (luminance) of the background color. So the only nonbackground color you get in GRAPHICs 8 is COLOR 1. Due to the way TV pictures are drawn, the tiny pixels can appear in different colors, depending on what column they are in (this is often called artifacting). *You can extend your color options*

I AM THE C-64™



Tutorial Series for the Commodore 64™

Get the most out of your Commodore 64 with the I AM THE C-64 tutorial series from Creative Software. Each practical operation you can perform is explained in simple terms right on the screen.

Step by step, the introductory series gives you an overall introduction to the Commodore 64, an introduction to the keyboard, and an introduction to BASIC programming language. The advanced series guides you through advanced programming techniques, sprite graphics, music and sound effects.

I AM THE C-64 provides you with a friendly and patient private tutor. For the Commodore 64 owner, this tutorial package is an unbeatable combination for learning all the power your computer has to offer.

CREATIVE SOFTWARE

this way, but we can't go into the details here. There is plenty of information on artifacting and other graphics modes in COMPUTE!'s First Book of Atari Graphics.

Apple Greetings

In my experience it seems that for the PR#6 command to work on the Apple, the greeting program that one INITializes the disk with must be entitled "HELLO". After RENAMEing the greeting program and attempting to boot the disk, a FILE NOT FOUND error resulted.

Jeff Walsh

Actually, you can assign any filename up to 30 characters in length to the greeting program (or to any other program). Ordinarily, this is done during the INITialization process. When INIT is executed, the disk is formatted and the name of the greeting program (in addition to being stored in the catalog) is placed on the disk in sector 9 of track 1. So when the disk is booted, the named greeting program is automatically run.

If you later decide to RENAME your greeting program (as you did) or have another program you wish to boot, you must also replace the filename for the booting program stored in track 1 or a FILE NOT FOUND error occurs. The following short program lets you replace the filename in track 1 with any filename you choose:

```
50  FOR I = 1 TO 36: READ J: POKE 3071 +
    I,J: NEXT
100  INPUT "NEW GREETING PROGRAM NAME -
    >";A$
120  FOR I =  LEN (A$) + 1 TO 30
130 A$ = A$ + " ": NEXT I
140  POKE 3072 + 22,1: CALL 3072
150  FOR I = 1 TO 30
160  POKE 8192 + 116 + I, ASC ( MID$ (A
    $,I,1)) + 128
170  NEXT
180  POKE 3072 + 22,2: CALL 3072
190  DATA  169,12,160,10,32,217
200  DATA  3,96,0,0,1,96
210  DATA  1,0,1,9,32,12
220  DATA  0,32,0,0,2,3
230  DATA  254,96,1,0,0,0
240  DATA  0,0,0,1,239,216
```

After you INPUT your new filename, the program writes it to track 1, sector 9. The short ML routine in this program enables you to replace the filename on the disk.

Cleaning Commodore Disk Drives

I own a Commodore 64 and a 1541 disk drive. I have cleaned the disk head with two different cleaning kits. However, neither kit had instructions on how to turn on the motor and lower the head for the 30–60 seconds that is recommended.

The best I have been able to do is use LOAD "$",8 for the directory many times. Even this does not work after a few attempts.

What is the best command or series of commands to clean the disk drives using these kits?

Douglas Gwost

The easiest way to spin the disk drive motor and engage the head is with the use of the DOS utility commands (SCRATCH, NEW, VERIFY, etc.).

To clean the disk drive, first remove any diskettes that may be in the drive. Follow the instructions for the cleaning disk and prepare it for use (with cleaning fluid, etc.). Now place the cleaning disk into the disk drive, close the door, and enter the following commands:

```
OPEN 15,8,15
PRINT #15,"I"
```

Entering the PRINT#15,"I" *command three or four times should spin the disk long enough to fully clean the head. After cleaning the drive, enter:*

```
CLOSE 15
```

After cleaning, and before using the disk drive, you might want to wait a few minutes. This will allow any residual cleaning fluid to dry, and reduce the possibility of contaminating a good diskette.

New Atari Graphics Mode?

I wrote a program that puzzles me:

```
10 GRAPHICS 9016:POKE 710,0
20 X=INT(RND(0)*318):Y=INT(RND(0)*191)
30 PLOT 159,95
40 COLOR Y:DRAWTO X,Y
50 GOTO 20
```

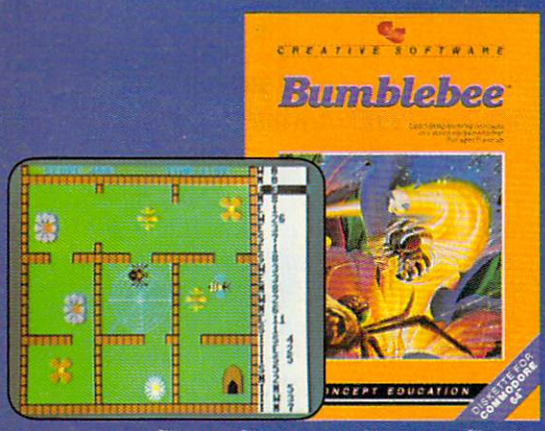
What is graphics mode 9016? When I break in and rerun the program, the previous picture is not erased! Also, sometimes it would suddenly clear the screen. Why? Is there something wrong with my computer?

Gordon E. Gizowski II

No, your computer is fine. You've just revealed some of the peculiarities of BASIC and the operating system. First, graphics modes are specified with a number from 0 to 15. If you add 16 to the number, the split screen will be disabled. If you add 32 to the number, the screen will not be cleared when the graphics mode is entered. But anything above 15+16+32 is just chopped off. In binary terms, only the lower six bits of the mode number are used. So GRAPHICS 71 is the same as GRAPHICS 7+64. Since 64 (bit 6) is not used, GRAPHICS 71 is the same as GRAPHICS 7. GRAPHICS 9016 is the same as 8+32+16+8960. Since 8960 is outside the range, it is ignored, and you get 8+32+16 (56), which is GRAPHICS 8 with no split screen. The 32, as mentioned, prevents the screen from being

Bumblebee™

Dancing a Fine Line Between Innovative Game and Educational Tool



For Commodore 64™, IBM PC™ and IBM PCjr™, and Apple™

Bumblebee is a highly interactive game which provides learning in a fun environment. That's what we call Concept Education.

Bart the Bee will demystify the programming process and teach your kids basic concepts without complex computer language. The player controls Bart by giving him instructions on how to move from flower to flower, picking up "pollen points." Bart's flight pattern must be carefully designed to avoid bumping into walls or becoming an unfortunate meal for Olga the Spider or Phineas the Frog.

Bumblebee requires logical "if-then" thinking. Your child is rewarded for accuracy and expediency and challenged by increasing levels of difficulty.

We call it concept education. Your kids will call it fun.

C R E A T I V E S O F T W A R E

cleared when GRAPHICS 8 is set up. A portion of the previous picture may have been destroyed, though, if you have changed modes (such as from GRAPHICS 8 to GRAPHICS 0).

The reason the screen was sometimes cleared is in line 40. You PLOT and DRAWTO random X,Y coordinates, but also use the Y coordinate for the color number. COLOR also chops off the part of a number that is not used. In GRAPHICS 8, only COLOR 1 and COLOR 0 are valid, so that odd numbers count as COLOR 1, and even numbers work as COLOR 0. PLOT is the same as PRINTing the CHR$ value of the color at the screen X,Y position (try using COLOR and PLOT with GRAPHICS 0, 1, and 2 to see the effect). If, however, the color number is 125, it is interpreted as CHR$(125), which is the same as the code for clear screen (CTRL-CLEAR). So COLOR 125:PLOT x,y will clear the screen. Your program is interesting, but to get the intended effect, you should use a different variable for the color. For example:

```
25 C=INT(2*RND(0)).
40 COLOR C:DRAWTO X,Y
```

TI Synthesizer Update

In the March 1984 issue of COMPUTE!, reader Jim Pate suggested using CALL PEEK(−28672,SP) on the TI-99/4A to check if the Speech Synthesizer is attached. He said that if it were attached, SP would be 96. This was correct to an extent. Because the address −28672 is part of the speech read/write buffer, sometimes (like after a CALL SPGET or CALL SAY) a value of 96 will not be placed into SP. To avoid this problem, instead of:

```
IF SP=96 THEN CALL SAY("UHOH")
```

use this:

```
IF SP THEN CALL SAY("UHOH")
```

This way, the CALL SAY statement will execute as long as SP is not 0.

Mark Chance

Thank you for the clarification on this.

Hidden 64 RAM

I have been dabbling in machine language a bit, and have a question. I would like to know if it is possible to load machine language programs into the RAM that is underneath BASIC ROM. If it is, how do I go about switching out BASIC ROM to use the ML routines, and then switching BASIC back in?

Kenneth Cox

There is 16K (16,384 bytes) of hidden RAM in the 64. 8K can be found underneath BASIC ROM at 40960 to 49151, hex $A000–$BFFF, 8192 bytes, and 8K is under the Kernal at 57344 to 65535, hex $E000–$FFFF.

Switching either BASIC or Kernal ROM in or out to expose the available RAM underneath is done via memory location 1. Normally, there's a 55 in that location. Setting bit 0 here to a zero will switch out BASIC and expose the 8K block of RAM underneath. Setting bit 1 of memory location one to a 0 will switch out both BASIC and Kernal ROM, exposing a total of 16K of RAM.

Use this BASIC line to switch out BASIC ROM:

```
POKE 1,PEEK(1)AND254
```

To switch out both BASIC and the Kernal, use:

```
POKE 1,PEEK(1)AND253
```

When memory location 1 is set at its normal value of 55 (BASIC and Kernal ROM switched in), POKEing and PEEKing to this memory follows special rules. When you PEEK this memory, you will get the values of the BASIC or Kernal ROM, that is, PEEK (40960). However, POKEing this memory (POKE 40960,255) will automatically POKE the RAM underneath.

This makes placing programs into the hidden RAM easy. You can POKE in your machine language routines via a BASIC poker program, or simply load the programs from tape or disk.

File Structure On Atari

I have an Atari 800 and am trying to write a BASIC program to access records in a file. If I open a file with a 9 to append the file, it will use the entire sector to store the data. If I open the file with a 12, I can write to the entire sector, but eventually I will come up with an EOF (End Of File) error. Is there any way to get around this problem? Also, are there any good books (besides the *DOS manual*) on file and record structure for the Atari disk?

Charles Bentivegna

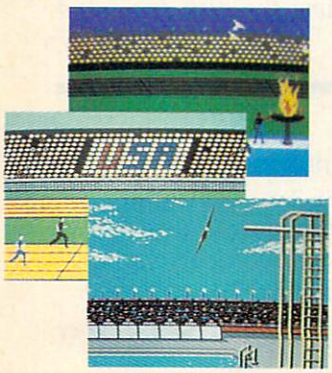
The OPEN command has four parameters:

```
OPEN IOCB#,access,aux,"filename"
```

IOCB# is a number from 1 to 7. There are eight Input/Output Control Blocks on the Atari. Each IOCB keeps track of an individual file. IOCB #0 is reserved for use by the screen editor (INPUT and PRINT). IOCB #7 is used for LPRINT, CSAVE, SAVE, LOAD, and CLOAD. When you OPEN a file to a particular IOCB, you use the same number when accessing the file with PRINT#IOCB; data or INPUT#IOCB, variable.

The second parameter, access, *is either 4 (OPEN for read), 8 (OPEN for write), 12 (OPEN for read and write), or 9 (OPEN to append). The* aux *byte is usually just 0. Access numbers 4 and 8 are straightforward. OPEN for read lets you GET or INPUT from that file, but not PRINT or PUT to it.*

SUMMER GAMES.™
WHY WATCH THE OLYMPICS WHEN YOU CAN BE IN THEM?



You're an Olympic athlete competing in eight key events at the Summer Games. How well can you score in track, swimming, diving, shooting, gymnastics and more? So realistic, there's even an opening ceremony and awards presentation after each event.

Unlike other "Olympics-Like" games, Summer Games has incredible realism, superb state-of-the-art graphics and sound effects (including national anthems from 18 countries), and it is a true action-strategy game. In each event you must plan and execute your game strategy in order to maximize your score. It is not just a matter of how fast you can move the joystick.

So change into your running shoes, grab your joystick and GO FOR THE GOLD!

One or more players; joystick controlled.

EPYX™
COMPUTER SOFTWARE

Strategy Games for the Action-Game Player

Access number 8 lets you create a file, or send data out to a device like a printer with PUT and PRINT. With a disk drive, using access number 8 will either create a new file or replace a previous one. Access number 12 lets you read and write to an existing file.

It's a little strange. You can keep reading the file until you get to the place you want to change, then start writing. Once you start writing, however, you can no longer read, since you have started to replace a portion of the file. With access number 9, you can only write to the end of a file. To keep things simple, the data you append to an existing file starts on a new sector, rather than filling up the remainder of the last sector used by the file. If you add short items to files with access number 9, you can waste a lot of disk space over the long run.

The only way you can both read and write independently of the disk is to use random access files. With NOTE, you can store the relative sector number of each block as you write the file to the disk. You can then refer to the information you stored with NOTE, and use POINT to jump directly to any sector in that file. You can write a single sector (record) independently of the rest of the file, and instantly skip to any sector without having to sequentially read through all the previous data. We cannot go into detail on the use of NOTE and POINT here, but the DOS 2.0S Manual *has most of the information you need. Another source for details on the working of DOS is Bill Wilkinson's* Inside Atari DOS, *available from COMPUTE! Books.*

Machine Language Decimal Mode

Can anyone explain the SED command? I used the SYS command to go to a machine language subroutine with the intent of returning to BASIC. The program ran fine until it hit the last statement—which was the RTS—and then it crashed and displayed an OVERFLOW ERROR message.

The machine language subroutine contained several JSR commands, and each one was covered with the RTS. The SED command was used before an addition. When the SED was deleted, everything worked fine. What did (and does) SED do?

E. H. Giles

When programming in machine language, SED stands for set decimal mode. *This command sets the* decimal flag *on the status register and tells the 6502 chip that all addition and subtraction is to be done in "decimal mode" (as opposed to "binary"). In this mode, the* carry flag *is set when addition exceeds 99.*

Setting the processor to the decimal mode has its drawbacks. For example, all additions and subtractions are then done in decimal, but the INC (increment) command still uses the binary mode.

Failing to clear the decimal flag (CLD) before returning to BASIC could cause catastrophic results. This is the reason your computer is freezing up when you exit your machine language subroutine. Try using the CLD (clear decimal) after the addition and before the RTS.

You always need a CLC before an addition, but SED is useful only in highly specialized applications. See Jim Butterfield's column "Machine Language" elsewhere in this issue for more on this topic.

Atari Disk And DOS

I own an Atari 800 and a Rana 1000 disk drive. Recently I purchased a game disk. I loaded in DOS, and typed A for a disk directory. All that was printed was how many sectors left. Is there any way to print the names of the files?

T. C. Birgler

At the lowest level of disk access, there are no filenames or directories. Data is stored in 128-byte blocks called sectors. A single-density Atari disk is divided into 720 of these sectors. DOS is a control program that makes this level of the disk invisible to you, and lets you create named files which can be accessed through a directory.

Most game disks don't need DOS, since there is no need for reading or creating named files. These boot disks load directly from the sectors into your computer memory without needing to load DOS. (DOS itself starts to load directly from a boot sector.) Since there are no named files, and no directory on most game disks, there is nothing for DOS to list when it looks on the disk where it expects to find the directory.

All this is similar to the fact that you can't use BASIC to LIST a machine language program. BASIC insulates you from machine language just as DOS insulates you from a disk system that inherently works only with sectors. You can use a disassembler to decode and list machine language. Likewise, there are programs that can directly read and display sector data. But just as you can't make much sense of a disassembly without any knowledge of machine language, the sector data can also be hard to follow without some background on how the disk drive and DOS work.

Colorful 64 Sprites

In looking over the various informational sources on sprites, I have come across a subject unanswered by all of them. The question is this: How does one tell the computer what color to make a certain part of a multicolored sprite?

Michael O'Day

Multicolored sprites are composed of four different colors. The four colors are 1) background color, 2) multicolor 1, 3) multicolor 2, and 4) sprite color. The locations to POKE to set the colors are as follows:

Background color:		**POKE 53281**
Multicolor 1	:	**POKE 53285**
Multicolor 2	:	**POKE 53286**
Sprite color	:	**POKE 53287 through 53294**

The eight sprite color locations correspond to the eight different sprites. For more information on programming with sprites, see the 64 Programmer's Reference Guide, *"Programming Graphics" section.*

Atari Peripheral Adequacy

I have an Atari 400 computer and an Atari 410 cassette recorder. I'm planning to buy an Atari 800XL computer, and I was wondering if the cassette recorder would work on the Atari 800XL computer.

Isaac Thornton Scott

As long as your recorder is still working fine with your 400, there should be no problem using it with an 800XL. You may want to have the tape heads cleaned and demagnetized, even realigned to give you a fresh start with your new computer. All 400/800 peripherals we know of will work just fine with the 600XL and 800XL computers.

Atari Graphics 2 Vs. 0

I have an Atari 1200XL. I have trouble seeing, so I prefer to use the largest size text mode, GRAPHICS 2. Is it possible to use this mode in place of GRAPHICS 0 for entering, editing, and running programs?

Wanda Ellis

First, you should be aware that GRAPHICS 2 interprets text characters differently than GRAPHICS 0. As set up, only uppercase characters are permitted. Lowercase characters, inverse characters, and inverse lowercase characters all appear in distinctly different colors than uppercase text, but still appear as uppercase. The screen editor is set up to work with GRAPHICS 0, which has 40 columns and 24 lines. GRAPHICS 2 has 20 columns and 12 lines. It is possible to use GRAPHICS 2 in place of GRAPHICS 0, in a limited way. Enter this line to see the technique:

```
GR.2+16:POKE 87,0
```

The +16 disables the text window, and POKE 87,0 fools the Atari into thinking it is in GRAPHICS 0. In this mode you will be able to type lines and even cursor around and make changes. The bottom half of the screen will be invisible, so scrolling will be tricky. Also, the cursor will only be visible when resting on a character. It wouldn't be too hard to write a machine language editor for using large size characters (perhaps with GRAPHICS 7).

COMPUTE! welcomes questions, comments, or solutions to issues raised in this column. Write to: Readers' Feedback, COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC 27403. COMPUTE! reserves the right to edit or abridge published letters. ©

David D. Thornburg, Associate Editor

Computer-Assisted Explorations With Music

I was having lunch with a friend one day when the conversation drifted to the subject of music, specifically the different aspects of music education. It was suggested that we spend a lot of time teaching skills in the reading and performance of musical compositions (especially with youngsters), but we spend little or no time teaching children (or adults, for that matter) how to create their own musical compositions.

As I thought about the similarities between this traditional approach to music and traditional approaches to, for example, math education, I was struck by an interesting idea. One of Logo's appropriate claims to fame is that it helps children to think mathematically—to explore mathematics as an experimental science, and to make math discoveries without outside "help" (or intervention). Professor Papert wanted to provide an environment in which children were free to explore mathematics on their own terms—to secure their own "ownership" of mathematical ideas.

While these are appropriate goals for mathematics education, they are no less appropriate when applied to other fields of endeavor, including music. In fact, I would guess that the general public might find music discovery to be every bit as exciting as math discovery.

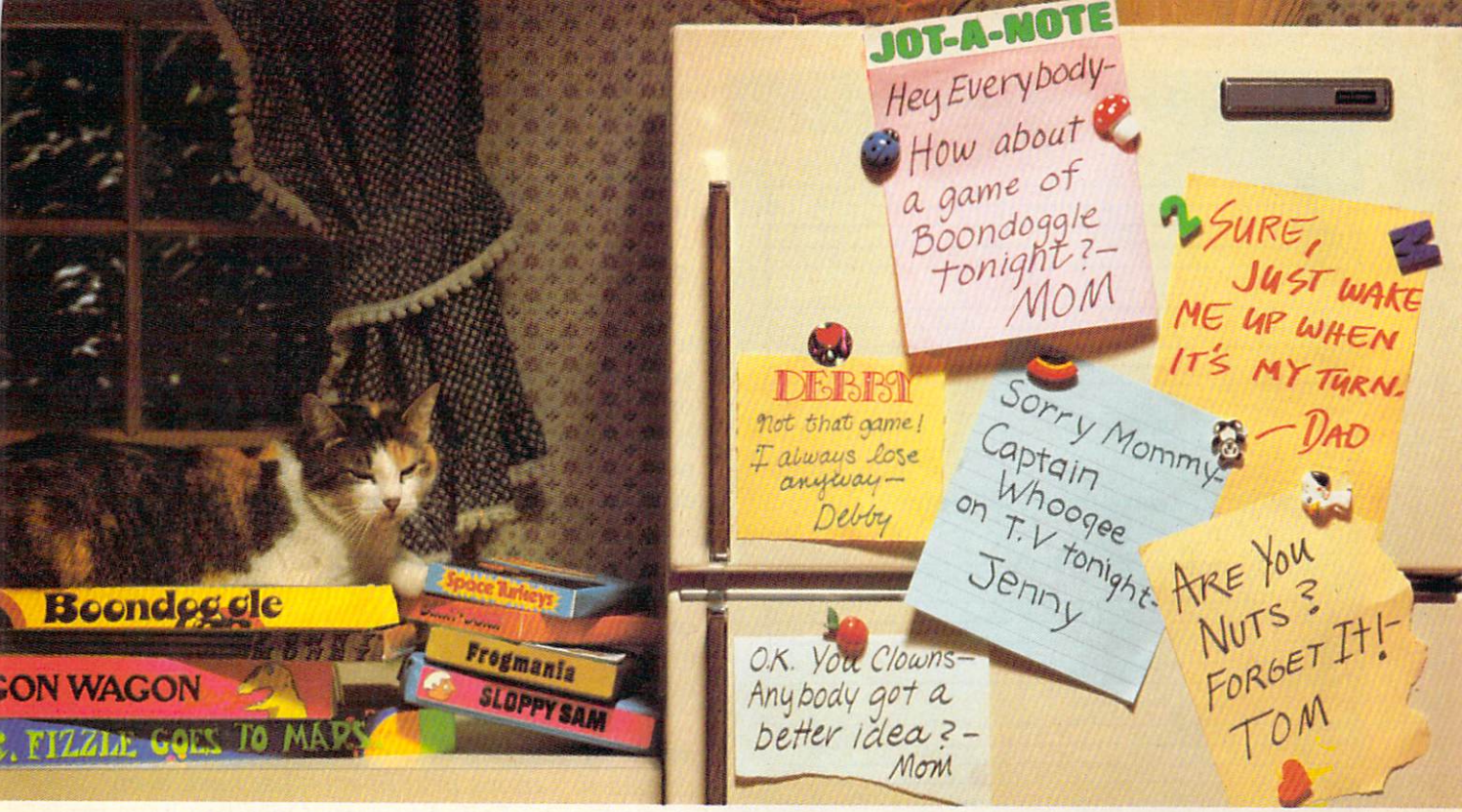
It turns out that the "discovery-based-learning" approach to music is not a new idea. Carl Orff and his colleagues created an exceptional program in this area that is still used in some schools. But just as Logo lets an individual make discoveries on his or her own, I think that the music discovery environment should operate in the same way. The computer is a perfect tool for this, and the idea of a musical equivalent to Logo is quite exciting.

A music program philosphically attuned to the Logo experience is already on the market, and it will soon be joined by add-on programs that preserve the spirit of discovery as the user explores musical ideas on his or her own. The product to which I am referring is *Musicland* from Syntauri.

Musicland is presently available for the Apple II family and requires the MusicSystem cards from Mountain Hardware. The user can interact with the system through the joystick, or KoalaPad. To give a feel for the areas that can be explored with this product, I will describe it in some detail. *Musicland* is divided into four types of activities: Music Doodles, Timbre Painting, Music Blocks, and Sound Factory. While these activities have cute names, and can be used by small children, *Musicland* is no more a kid's product than is Logo. Professional musicians have enjoyed it every bit as much as children.

Music Doodles lets you create motifs by "drawing them" on a grand staff appearing on the screen. For example, if you wanted to hear some music that looks like the letter A, all you would have to do is draw an A on the screen (see Figure 1).

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. David Thornburg's recent books include Computer Art and Animation: A User's Guide to Atari Logo, The KoalaPad Book *(in which* Musicland *is also described), and* Exploring Logo Without a Computer *(a book for teachers). All three of these books are published by Addison-Wesley. His Macintosh book (*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.*



If getting the whole family together is a real challenge, maybe you need games that really challenge the whole family.

Introducing a new generation of computer games. Family Learning Games from Spinnaker.

Ever notice how a little fun with the family can be a little hard to arrange?

Well, now there's a solution – Spinnaker's Family Learning Games. A whole family of great games that make getting the family together seem like child's play. And make "family fun" really seem like fun again. What's more, they'll even help your kids develop some very important skills.

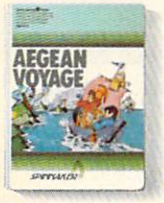
What makes our Family Learning Games so special? Well, for one thing they're designed to challenge and excite everyone in the family, from grade schoolers to grownups. Their unique combination of chance and strategy makes them perfect for young players, yet challenging enough that everyone will want to play them again and again.

But what makes our Family Learning Games even more unique is how they help kids learn – about problem solving, strategizing, spelling, even Greek mythology. That's quite a bit more than they'd learn from a typical board game (if you could even get them to play a typical board game).

So next time you want to get everybody together, don't get discouraged – get Spinnaker's Family Learning Games.

You'll find the biggest challenge in family fun won't be on the refrigerator. It'll be on the computer.

Spinnaker games are available for ColecoVision® and for Coleco Adam,™ Commodore 64™ and Atari® home computers.



It's New!

AEGEAN VOYAGE.™

Where do monsters lurk? And which islands have treasures to behold? Heed the oracle's words, for only his clues can lead you to riches and a safe return. **Ages 8 - Adult.**

ADVENTURE CREATOR.™

Design a challenging adventure game that everyone can play – or let the computer design one for you. It's exciting, creative – and utterly addictive! **Ages 12 - Adult.**

UP FOR GRABS.™

It's a wildly exciting crossword game where everyone has to think fast. More words will help you win – but don't get caught with leftover letters! **Ages 8 - Adult.**

SPINNAKER™

We make learning fun.

Cartridges for: ColecoVision, Coleco Adam, Atari and Commodore 64.

Figure 1: Drawing The Sound Of A

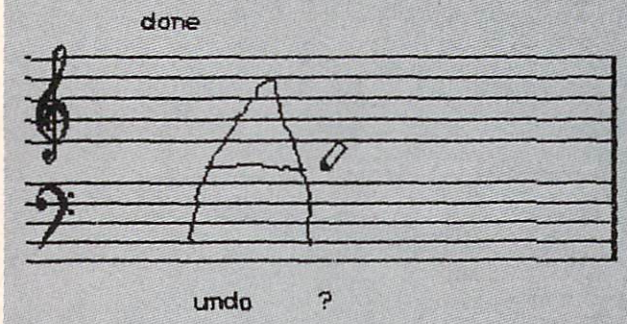


Figure 2: From Picture To Notes

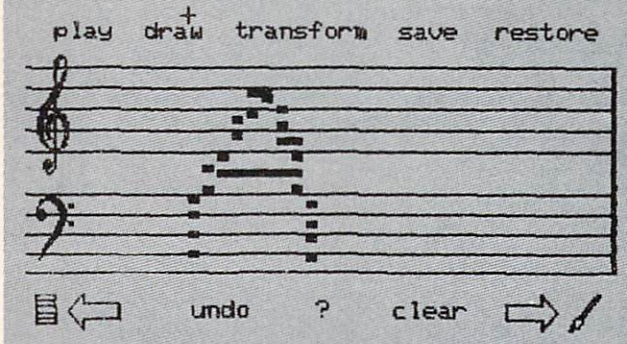


Figure 3:
Experimental Transformation Of Motif

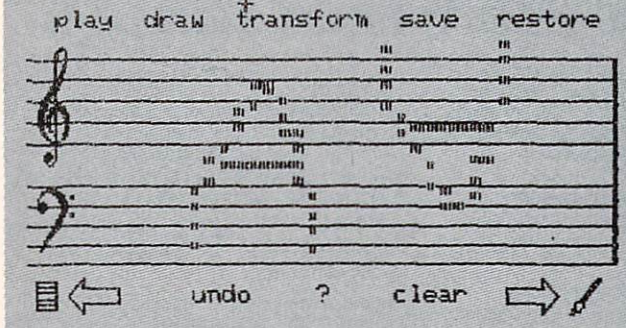
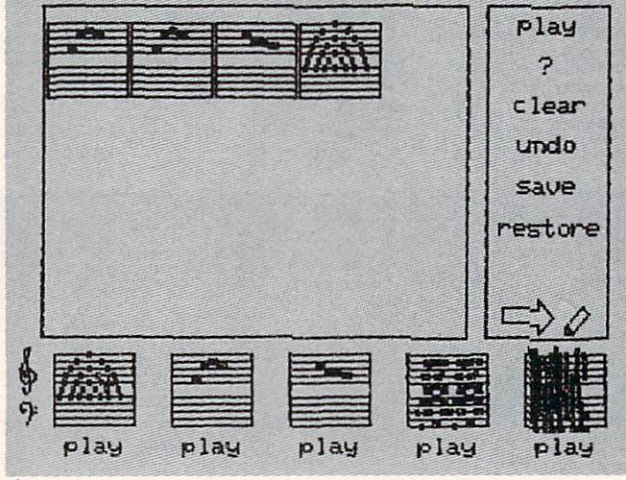


Figure 4:
Creating Melodies With Music Blocks



When this drawing is finished, *Musicland* translates the picture into notes. Note duration is represented by length, and note value is represented by vertical position. Sharps and flats are indicated by partial positioning on the staff. The result resembles a player piano roll more than it does a traditional score. (See Figure 2.)

Once a motif has been drawn, you are free to experiment with it and to transform it anyway you want. For example, a motif can be stretched or compressed in time (including making it play backwards), and it can be stretched or compressed vertically (including turning it upside down). A motif can be transposed to any position on the staff, and as a result, one musical idea can be transformed into the structures used in music from Baroque fugues to the 12-tone-scale music of Schönberg. For example, I have transposed and inverted our motif before placing it next to the original in Figure 3.

Timbre Painting lets you add color to your music by painting over notes with colors that pertain to various instruments that have been created in the Sound Factory. Once you have created a motif using Music Doodles, you can create entire melodies by bringing several doodles to Music Blocks. This tool lets you assemble a complete piece by building the composition from an assembly of any of five music blocks you have created with Music Doodles. By pointing to a block, you can automatically place it at the next available location in the score. (See Figure 4.) Once the blocks are in place, they can be switched around or deleted until the final composition meets your goals.

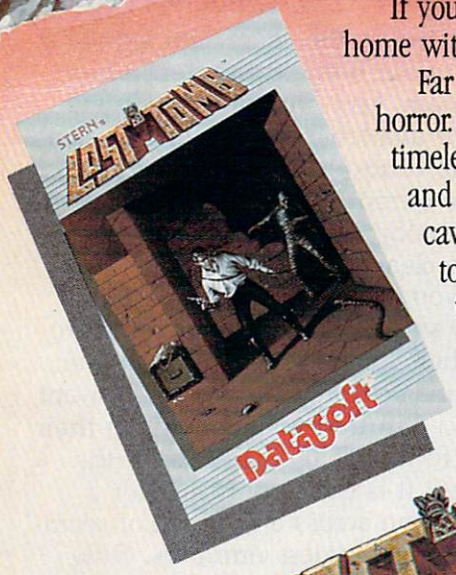
The Sound Factory gives you the complete freedom to create your own musical sounds by selecting harmonic content and a time envelope. Because of the ease with which various sounds can be made, you can experiment with many types of sounds. *Musicland* even lets you work with the sound waveform instead of the harmonic content, if you wish.

The documentation includes project cards and shows that *Musicland* can be used as a tool for experimenting with musical ideas, and as a tool for exploring the physics and aesthetics of sound as well.

Musicland was designed by Dr. Martin Lamb and his colleagues at the University of Toronto. I expect further product developments in this area in the near future.

As we see more discovery-based-learning emphasis in our educational system, we will continue to see the computer being used in ways that show the unique strengths of this medium. The computer is not just a teaching tool—it is a tool to help us learn. The distinction is important.

THE GAME YOU CAN GET WRAPPED UP IN.



LOST TOMB

If you thought Egypt was the home of the Pyramids, wait until you get home with Lost Tomb.™

Far more than mere chambers of wonder, these chambers are filled with horror. Poisonous scorpions, screeching bats and terrifying mummies. And in the timeless tradition of the most daring expeditions, you'll pack a pistol, plenty of ammo and a whip to crack the curse of the pyramids. Earthquakes rumble along cavernous passageways. Walls crumble and crackle with gunfire. Your mission is to make it through all 91 chambers and 13 levels. And then make off with the loot. The only things we can't give you are the things you'll need most. Cool reflexes, uncanny instincts and the courage to use them.

Lost Tomb.™ Can you unravel the mystery?

Available now for Atari, Commodore 64, Apple II series and IBM PC and PC/JR. Suggested retail price $29.95. Check with your local home computer software retailer for Lost Tomb,™ and to learn of other great programs from Datasoft® send for a free consumer catalog.

By Datasoft®

19808 Nordhoff Place, Chatsworth, CA 91311 Phone (818) 701-5161

Robert Alonso, Assistant Editor

Printing And Asking

In most versions of BASIC there are usually many ways to accomplish a task. For example, to clear the screen Atari users have the option of printing either a control character or CHR$(125). On any Commodore computer, the user has a similar option, but the CHR$ would have to be (147). Beginners usually decide that they prefer PRINTing the control codes between quotes. The reason for this choice is that the CHR$ command and the associated ASC command are often misunderstood.

The CHR$ function is really a very easy function to use and to understand. Let's say that you wanted to print the letter A on your computer's screen. You could simply type PRINT "A" and hit the RETURN key. However, you could also PRINT CHR$(65). The result would be the same. Although it is always nice to know that you have different ways to do things, you probably are better off just printing the A on the screen.

The real value of the CHR$ command is that it allows you to print special control codes such as cursor movement and screen clearing commands. It is far more confusing, for example, to have embedded commands (such as the reverse-video heart in a Commodore listing) than to have the same commands as CHR$. There are usually CHR$ codes that allow you to change the color of the cursor and even ones that let you ring the internal bell of the computer. On the Apple and Adam home computers you can easily have the bell ring by just PRINTing a few CHR$(7)'s.

ASC The Computer

The reverse of the CHR$ command is ASC. ASC will let you find out what decimal number represents a given character. If, for example, you want to know the number that you would have to include with a CHR$ to PRINT a comma on the screen, you could just ask the computer via ASC. The correct syntax for the command is PRINT ASC(","). You would get 44 as the answer. If you then PRINTed CHR$(44), the computer would print out a comma. An easy way to remember what each function does is to remember that ASC is used for *ask*ing the computer for the correct number and the other is for giving the computer the right number.

Secret Coding

Program 1 demonstrates the characters immediately available on your home computer. When you run the program, keep in mind that it will print out all of the available characters as well as control codes. The control codes will affect the appearance of the output to the screen. For example, once the loop reaches 125 in the Atari, the screen will be cleared.

One reason you might want to use CHR$ within your programs is that you might want to conceal words when someone uses one of your programs. In a game of Hangman you might want all the words placed in DATA statements in their ASCII numerical format. This will prevent the user from cheating. It is quite confusing for a snooping user to come across a long list of seemingly random and meaningless numbers. Only you will know!

Program 2 is an example of how you might encode a sentence so that only you know what it means. If you take a close look at lines 50–70, you'll notice that it takes a lot more space to store a sentence in this method than it would to

THERE'S A COMPUTER BORN EVERY MINUTE...

GIVE IT A HOME.

For **$89.95** with the CS-1632 you can house your computer, peripherals, and accessories without spending a fortune.

For those with a large computer family the CS-2748 gives you all the room you need for your computer, monitor, printer, peripherals, software, etc. at a price that's hard to believe: **$299.95.**

The CS-1632 computer storage cabinets compact yet functional design fits almost anywhere while housing your computer monitor, joysticks, software, books and peripherals all for only $89.95.

The slide out shelf puts the computer at the right height and position for easy comfortable operation.

The fold up locking door keeps unwanted fingers off the key board when not in use.

To store joysticks just turn them upside down and slide them into the inverted storage rack.

Twist tabs on the back of center panel allow for neat concealed grouping of wires, while power packs rest hidden behind center panel on shelf.

The slide out software tray has room for 14 cartridges or cassettes and up to 30 diskettes. Most brands of software will fit between the adjustable partitions with a convenient hook for the spare key at rear.

Stand fits Atari 400 & 800, Commodore 64 & VIC 20, Ti 99/4A and TRS-80.

Cabinet dimensions overall 36" high x 33-7/8" wide x 16" deep.

To order CS-1632 send $89.95 to:

To order CS-2748 send $299.95 to:

HYTEC SYSTEMS

P. O. Box 446
West Lynn, OR 97068
For Fast Phone Orders Call Toll Free **1-800-547-3100**
Inside Oregon Call **(503) 635-6667**

Name _____

Address _____

City _____ State _____ Zip _____

Quantity_____ CS-1632 Quantity_____ CS-2748

☐ Golden Oak Finish ☐ Natural walnut finish

☐ My personal check, cashiers check or money order is enclosed.
☐ Bill my VISA # _____ Exp. Date ____
☐ Bill my MasterCard # _____ Exp. Date ____
☐ Please include freight charge on my VISA or MasterCard.

Card Holders Signature _____

Immediate shipment if in stock. If not, allow 3-4 weeks for delivery. If personal check is sent allow additional 2 weeks. CS-1632 ships UPS freight collect from Oregon. CS-2748 ships by truck freight collect from Oregon. Prices subject to change. Shipment subject to availability.

Both the CS-1632 and CS-2748 ship unassembled in two cartons. Assembly requires only a screwdriver, hammer, and a few minutes of your time.
Choice in simulated woodgrain of warm golden oak or rich natural walnut finish.

The two slide-out shelves put the keyboard at the proper operating height while allowing easy access to the disk drives.

The bronze tempered glass door protecting the keyboard and disk drives simply lifts up and slides back out of the way during use.

Twist tabs on the back of the center panel allow for neat concealed grouping of wires while a convenient storage shelf for books or other items lies below.

The printer sits behind a fold down door that provides a work surface for papers or books while using the keyboard. The lift up top allows easy access to the top and rear of the printer. A slot in the printer shelf allows for center as well as rear feed printers.

Behind the lower door are a top shelf for paper, feeding the printer, and a bottom shelf to receive printer copy as well as additional storage.

Stand fits same computers as the CS-1632 as well as the Apple I and II, IBM-PC, Franklin and many others.

The cabinet dimensions overall: 39-1/2" high x 49" wide x 27" deep.

Keyboard shelf 20" deep x 26" wide. Disk drive shelf 15-34" deep x 26" wide. Top shelf for monitor 17" deep x 27" wide. Printer shelf 22" deep x 19" wide.

HYTEC SYSTEMS

Table 1:
Atari Special CHR$ Codes

CHR$	Atari Effect
27	ESC
28	Cursor Up
29	Cursor Down
30	Cursor Left
31	Cursor Right
125	Clear Screen
155	RETURN
253	Buzzer

Table 2:
Commodore Special CHR$ Codes

CHR$	Commodore Effect
13	RETURN
14	Switch to Lowercase
17	Cursor Down
19	Cursor Home
29	Cursor Right
142	Switch to Uppercase
145	Cursor Up
147	Clear Screen
157	Cursor Left

simply type in the letters. Although it does take up extra memory, you will soon appreciate the potential that CHR$ codes have for creating fun and educational quiz programs.

As you can see, the program is straightforward. Line 10 starts everything by initializing a FOR–NEXT loop. The number 29 corresponds to the number of characters, including spaces and punctuation, that the sentence has. There are exactly 29 numbers in the DATA statements. The READ A command in line 20 gets one number in from the DATA statements for each pass through the FOR–NEXT loop. Line 30 is used for printing the characters on your screen. The CHR$ function is used here for converting the numbers to their appropriate letter representation. A semicolon is placed after the command PRINT CHR$(A) so that the characters are printed next to each other instead of down the left-hand side of the screen. The semicolon eliminates the carriage return that is executed at the end of each PRINT operation.

Line 35 is just a delay loop, and line 40 sends the computer back to line 10 to go through the next number in the loop. If you wanted to use this routine in one of your programs, you could easily make it a subroutine. Just add a line 45 with the instruction RETURN and you can then access the routine by GOSUBing to it from your main program. To modify the message length, just increase or decrease the 29 in line 10. The message can be changed by just typing the right numbers into the DATA statements. To find out which numbers to place in the DATA statements just type PRINT ASC("X"), where the X stands for the letter that you need to know about. If you typed PRINT ASC("X"), for example, you would get 88 as the answer.

There are usually tables of the letters and control codes that each computer can print in each computer's user's manual. You can use these tables to help you find the letters and symbols you need for your program. The following tables should get you started on the Atari and Commodore computers.

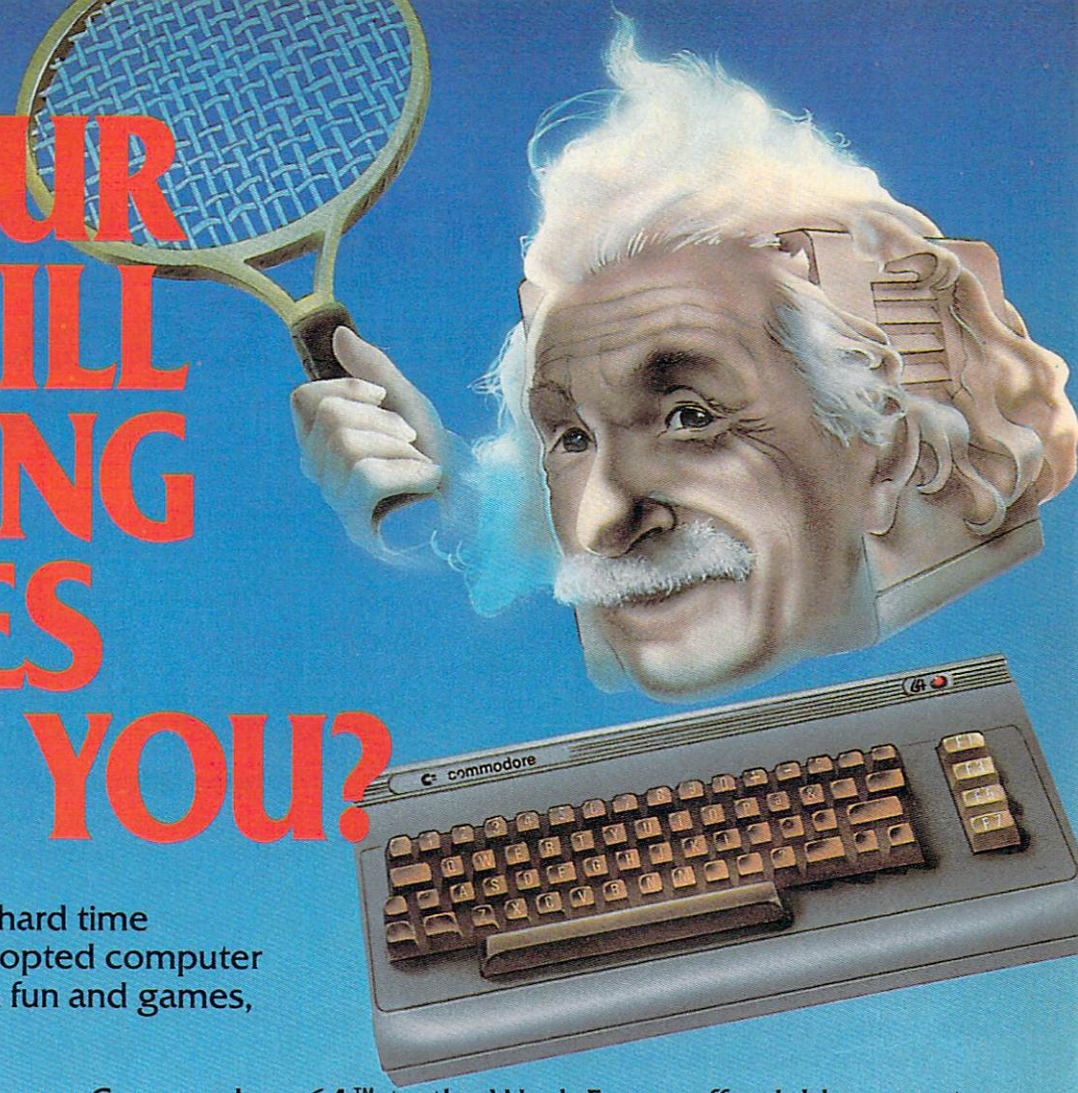
Program 1: CHR$ Display

```
10 FOR X=0 TO 255
20 PRINT CHR$(X);
30 FOR DE=1 TO 100:NEXT DE
40 NEXT X
```

Program 2: Secret Message

```
10 FOR X=1 TO 29:REM BEGIN LOOP
20 READ A: REM FETCH FIRST NUMBER
30 PRINT CHR$(A);:REM PRINT CORRESPONDING
   CHARACTER
35 FOR D=1 TO 50:NEXT D:REM DELAY LOOP
40 NEXT X
50 DATA 84,72,69,32,67,72,82,36,32
60 DATA 70,85,78,67,84,73,79,78,32
70 DATA 67,65,78,32,66,69,32,70,85,78,33
```

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

Software Power!

The Summer Consumer Electronics Show

Selby Bateman, Features Editor

Some hardware manufacturers have bailed out, but software is soaring. The introduction of several new personal computers at the Summer Consumer Electronics Show, held in Chicago in June, was not the only story. Just as important was the overwhelming amount of new software in almost every conceivable field of interest.

The gold-rush giddiness that brought 17 new computers to last year's Chicago's CES extravaganza is gone. This was the year of software. It's become a potential boom market in the highly competitive personal computer field.

Remember these names? Atari 1400 XL, Mattel Aquarius and Aquarius II, Spectravideo SV-318 and SV-328, Texas Instruments TI-99/4A, Timex/Sinclair 1000. Several of these companies have backed away from manufacturing personal computers over the past year. Some have withdrawn announced machines. It makes a long and revealing list.

Nonetheless, CES did show that there's plenty of life, and more than enough interest, in the growth potential of the personal computer field.

Miles Of Aisles

More than 90,000 exhibitors, journalists, dealers, and celebrities strolled along the miles of exhibits at CES—the world's biggest trade show—looking at virtually every kind of consumer electronic product in the world. And a good percentage of those attending spent much of their time just trying to get around in one building, McCormick West—three warehouse floors full of nothing but computer hardware and software.

More than 170,000 square feet of space was allotted in McCormick West at this year's show, a 25 percent jump over last year's floor area. And the sheer quantity of new software being introduced was enough to make even the most dedicated computerphile's eyes glaze over.

The good news is obvious: more software for virtually every home computer, especially Commodore, Atari, Apple, and IBM. And the bad news is equally plain: How can you learn about it all, let alone pick out the quality products?

Where Were IBM And Apple?

The software boom doesn't mean that computer hardware was unimportant at CES. The home computer market continues to evolve, and in some quite interesting ways.

For example, IBM and Apple, two of the biggest contenders in the personal computer field, didn't attend CES. Their dealer networks and their market strategies are not based around this trade show, as are some others.

Commodore, Coleco, and Atari, three other major contenders, did attend, however. And what they introduced, announced, or revealed says a lot about where home computers are heading this year. (See "Atari's CES Line-Up" in this issue.)

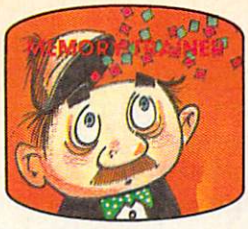
Possibly The Most Advanced Personal Computer Ever

Some of the most fascinating computers at this CES were under wraps, available for inspection only to a privileged few. For example, the new high-end Atari computer was shown only to software developers, and most of the Japanese MSX-standard home computers due in the U.S. next year were seen only at a private party thrown by Microsoft. Likewise, what might be the most advanced personal computer ever designed was shown behind closed doors inside the Amiga exhibit.

The computer is code-named the Amiga Lorraine, and right now it exists only in prototype form. But if it ever reaches production, and at a price even close to what is promised, it could signal the beginning of a completely new generation of personal computers.

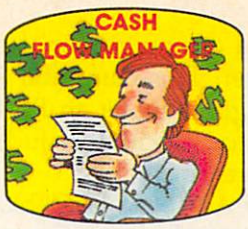
The Lorraine's graphics are a whole step

Get the jump on the weatherman by accurately forecasting the local weather yourself!



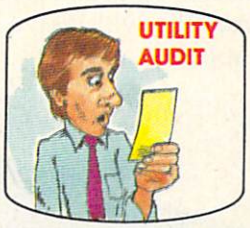
A scientifically proven way to develop an awesome memory.

You are trapped in a five-story, 125-room structure made entirely of ice. Find the exit before you freeze!



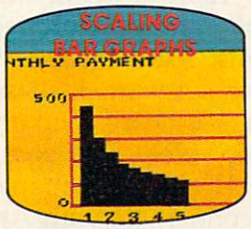
Take control of your personal finances in less than one hour a month.

The beautiful princess is held captive by deadly dragons. Only a knight in shining armor can save her now!



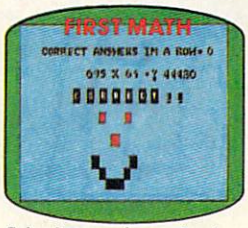
Cut your energy costs by monitoring your phone, electric and gas bills.

Computerize car maintenance to improve auto performance, economy and resale value.



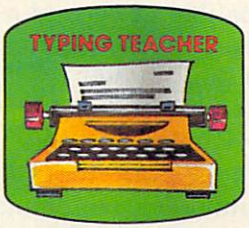
Create multi-colored bar graphs with a surprisingly small amount of memory.

A time-saving organizer for coupons, receipts and more.



School-age and pre-school children are rewarded for right answers, corrected on their wrong ones.

A real brainflexer. Deflect random balls into targets on a constantly changing playfield.



A fun way to dramatically increase typing speed and accuracy.

Get up to 30 new programs and games for less than 15 cents each— every month in COMPUTE!

Every month, COMPUTE! readers enjoy up to 30 brand new, ready-to-run computer programs, even arcade quality games.

And when you subscribe to COMPUTE! at up to 40% off the newsstand price, you'll get them all for less than 15 cents each!

You'll find programs to help you conserve time, energy and money. Programs like Cash Flow Manager. Retirement Planner. Coupon Filer. Dynamic Bookkeeping.

You'll enjoy games like Air Defense, Boggler, Slalom and High Speed Mazer.

Your children will find learning fast and fun with First Math, Guess That Animal and Mystery Spell.

Looking for a challenge? You can write your own games. Customize BASIC programs. Even make beautiful computer music and pictures.

It's all in COMPUTE! All ready to type in and run on your Atari, Apple, Commodore, PET/CBM, TI 99/4A, Radio Shack Color Computer, IBM PC or IBM PCjr.

What's more, you get information-packed articles, product reviews, ideas and advice that add power and excitement to all your home computing.

CALL TOLL-FREE
1-800-334-0868

And when it's time to shop for peripherals or hardware, check COMPUTE! first. Our product evaluations can save you money and costly mistakes. We'll even help you decide what to buy: Dot-matrix or daisy-wheel printer? Tape storage or disk drive? What about modems? Memory expansion kits? What's new in joysticks, paddles and track balls?

SAVE UP TO 40% OFF THE NEWSSTAND PRICE ON COMPUTE!

SUBSCRIBE NOW AND SAVE UP TO 40% ON COMPUTE!

Yes! Start my subscription to COMPUTE! for:

☐ 1 year $24—32% off!
☐ 2 years $45—36% off!
☐ 3 years $65—40% off!

☐ Payment enclosed ☐ Bill me

Charge my ☐ Visa ☐ MasterCard ☐ Am.Ex.

Account No. _____ / _____
EXP. DATE

Name _____

Address _____

City _____ State _____ Zip _____

Return the coupon or postpaid card today!

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

ahead of any personal computer now on the market. This computer is potentially powerful enough to make an IBM-PC look like a four-function calculator. Judge for yourself. Standard features include:

- A Motorola 68000 microprocessor chip for the central processing unit. This is the same 16/32-bit chip found in the Apple Macintosh.
- 128K of RAM, expandable to 512K internally and several megabytes (1000K) externally. 64K of ROM, with built-in BASIC and speech software, including a text-to-speech program. We heard the Lorraine talk in its male and female voices, and both were quite understandable. The BASIC language is said to be very fast and compatible with Applesoft, though with extra commands for graphics and other capabilities.
- Built-in 320K double-sided disk drive, IBM-compatible. A second external drive can be powered by the internal power supply.
- Built-in 300-bps (bits per second) modem, replaceable with a 1200-bps modem.
- Parallel and serial interface ports; a top "chimney port" for individually powered expansion modules, including more RAM and a hard disk drive; and a front cartridge slot for ROM software or coprocessors, such as an 8088 module for IBM/MS-DOS compatibility.
- Four sound channels, with music capabilities comparable to the Commodore 64's SID chip. We heard this demonstrated with a plug-in organ keyboard; the sound was very impressive. (One sound channel is used by the speech software.)
- Medium-resolution graphics of 320 × 200 pixels (screen dots), and hi-res graphics of 640 × 200 pixels, with a total of 4096 colors. (That's not a typo. We saw a dramatic rainbow demo which supposedly displayed all 4096 colors on the screen simultaneously, though we didn't have time to count them.)
- Eight sprites (up to 16 colors each) with collision detection and display priorities. Plus another feature called "frame-buffer animation," which lets you pick up any piece of the screen and move it anywhere else. Plus built-in commands for line-drawing, fills, and both horizontal and vertical fine scrolling. Plus split-screen graphics, with each screen "window" capable of displaying different graphics modes while operating independently, even with fine scrolling.
- Outputs for TV, composite video, and two types of RGB (red-green-blue) direct-drive monitors. The TV output shown to us was so pure that 80-column text (also standard) was almost readable from across the room.
- On top of all this, Amiga claims the Lorraine will come bundled with software, including a disk operating system, word processor, and spreadsheet.

So how much will this wonderbox cost? According to Amiga, only $1500. Amiga also claims the Lorraine will be ready for shipment by Christmas.

However, there's a big difference between design engineering and production engineering. The Lorraine at the June CES was such a rough prototype that it was operated from a remote terminal. To gear up for production in less than six months will take a herculean effort, and lots of capital.

Commodore's new Plus/4 computer—a revised version of the 264 shown by Commodore last January—has four programmable function keys and four separate cursor keys.

Commodore's Plus/4 And 16

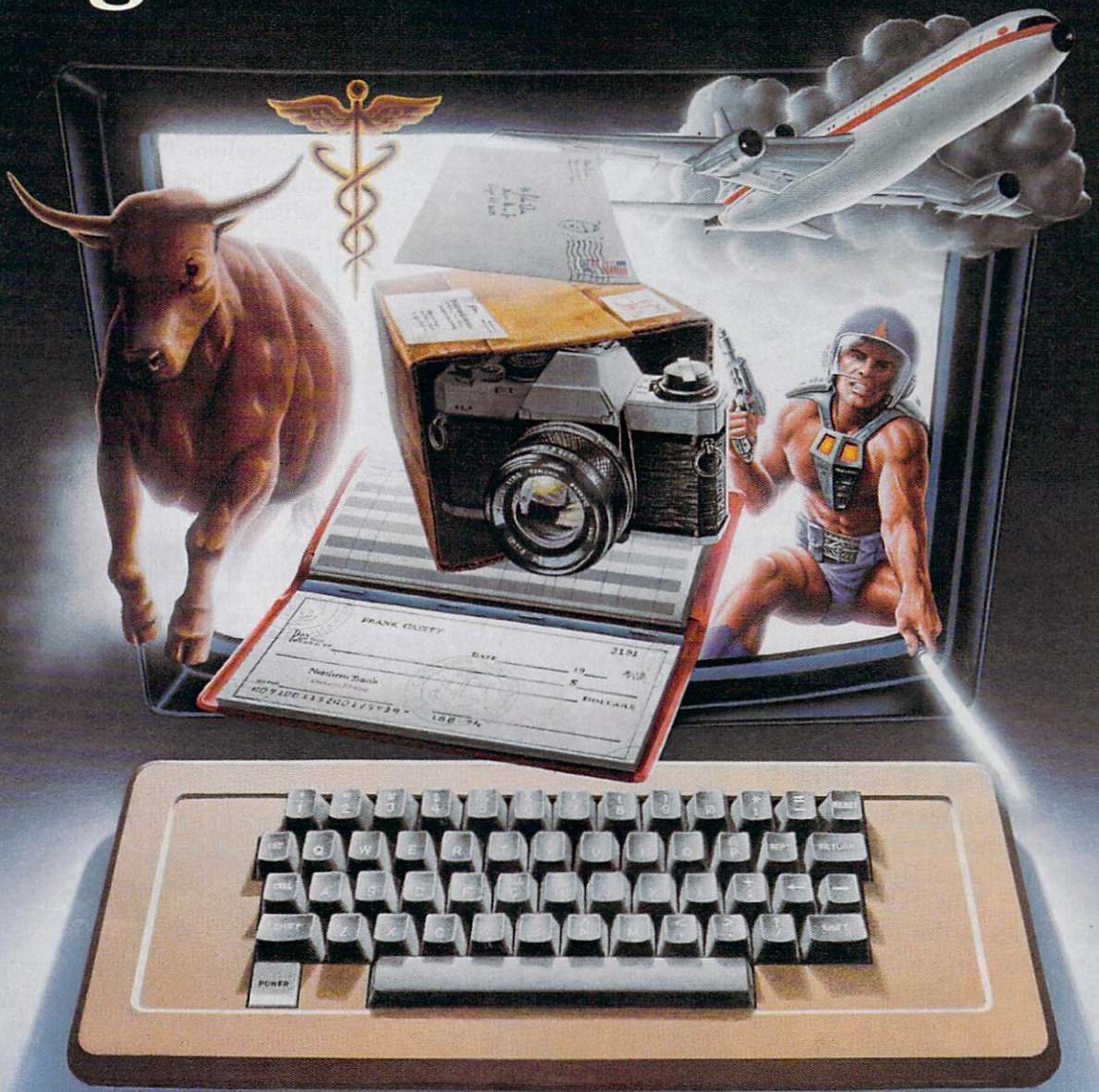
Commodore showcased two new computers. The Plus/4, which Commodore calls its "productivity machine," is based on the technology introduced (but never released) with the Commodore 264 at January's CES in Las Vegas. The Plus/4 comes with four built-in integrated programs: a word processor, data base, spreadsheet, and graphics.

The 64K RAM (random access memory) computer allots a full 60K for BASIC programming, and features an enhanced BASIC with over 75 commands, including 11 for graphics. The machine also has eight reprogrammable function keys, four separate cursor keys, 128 colors (16 primary colors and 8 luminance levels), a 320 × 200 pixel screen resolution, and a 12-command, built-in machine language monitor.

Commodore believes the Plus/4 offers a productivity-oriented alternative to the popular Commodore 64, which continues to sell very well. The Plus 4 does not have such attractive Commodore 64 features as the versatile SID (Sound Interface Device) chip, with its three independent voices, or the eight programmable, independently movable sprites.

Since the Plus/4 is significantly different internally from the 64, the two are largely incompatible when it comes to software. Commodore emphasizes that sales of the 64 continue to

We don't care which computer you own. We'll help you get the most out of it.



CompuServe puts a world of information, communications, and entertainment at your fingertips.

CompuServe is the easy to use videotex service designed for the personal computer user and managed by the communications professionals who provide business information services to over one fourth of the FORTUNE 500 companies.

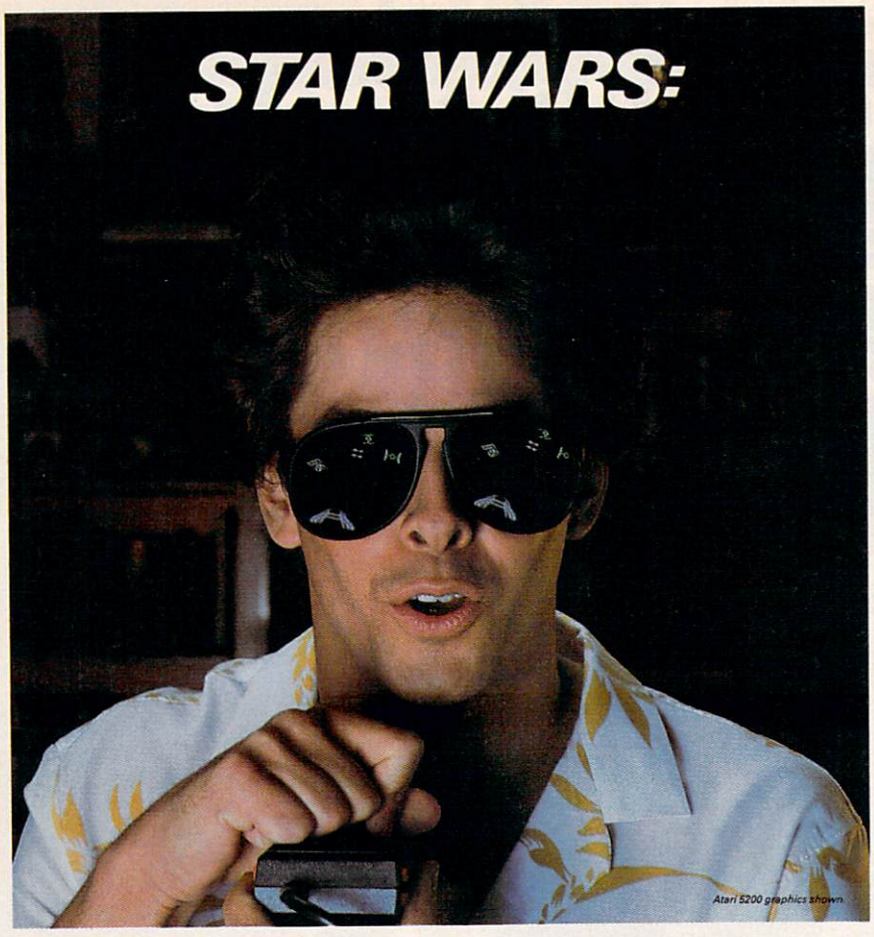
Subscribers get a wealth of useful, profitable, or just plain interesting information like national news wires, electronic banking and shop at home services, and sophisticated financial data. Plus, a communications network for electronic mail, a bulletin board for selling, swapping, and personal notices and a multichannel CB simulator.

You get games on CompuServe, too. Classic puzzlers, educational, sports and adventure games and fantastic space games featuring MegaWars, the "ultimate computer conflict."

To learn more about CompuServe, call toll-free, 800-848-8199, for an illustrated guide to the CompuServe Information Service. The videotex service for you, no matter which computer you own.

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



be excellent, and that the company's support of the machine will remain strong.

The Plus/4 will reportedly sell for about $299, and is expected to be on store shelves by this fall.

"The Learning Machine"

The Plus/4's younger brother is the new Commodore 16, which the company calls "The Learning Machine." This 16K machine is aimed at the first-time computer user, and will take the place of the still popular 4K VIC-20.

Commodore officials indicated at CES that while the company plans to continue software support for the VIC, production has stopped on the machine. The new Commodore 16 will reportedly retail for about $100, and is clearly aimed at the same market as the VIC.

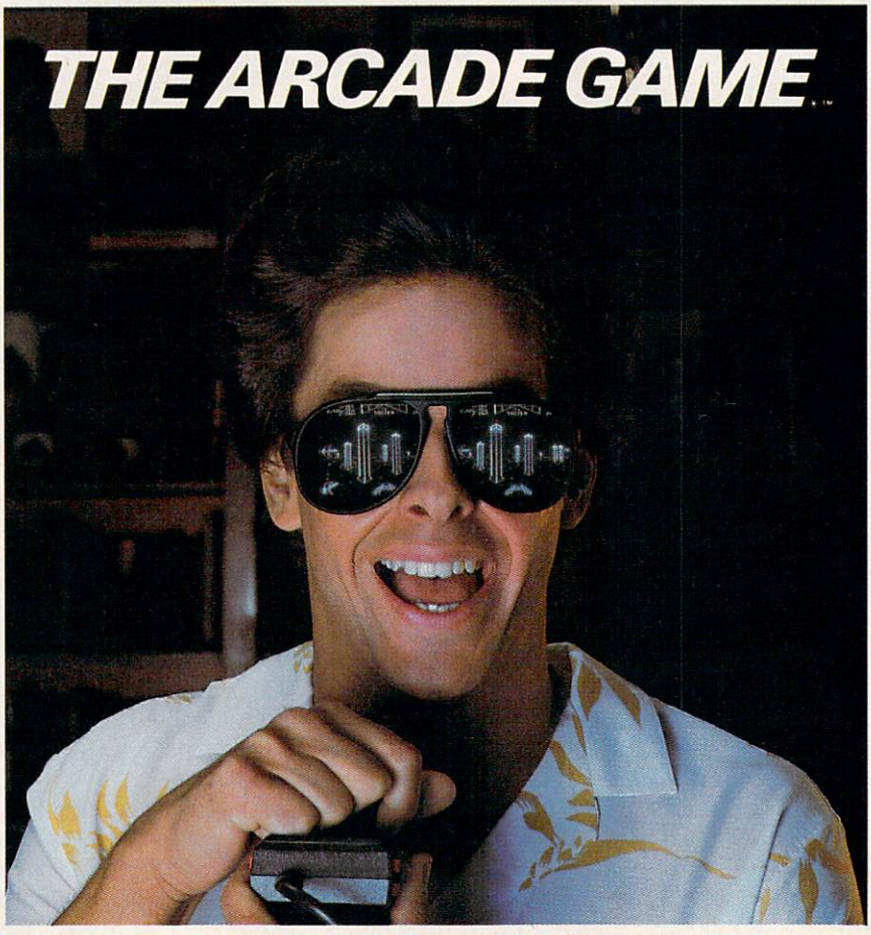
The Commodore 16 will use a Commodore Datassette recorder as the primary storage device, and can handle cartridge or cassette-based software. However, the 16 does have a disk drive port for those who wish to use it. Special features include a built-in machine language monitor, graphics and sound commands, BASIC 3.5, and screen window capability.

Commodore Peripherals

Commodore also introduced the following peripherals at CES, all of which are scheduled for fall release (prices have not been announced as of this writing):

• Commodore DPS 1101 Daisywheel Printer—A business-oriented letter-quality printer which features a bidirectional, logic-seeking print mechanism that prints at 18 characters per second (cps). It is compatible with the new Plus/4 computer.

• Commodore MPS 802 Dot Matrix Printer—A bi-directional impact dot matrix printer with a speed of 60 cps for correspondence-quality



print. It is compatible with all Commodore computers and prints numerics, symbols, and all PET graphics.

• Commodore MCS 801 Color Dot Matrix Printer—A unidirectional color impact dot matrix printer for users of the Commodore 64. It prints at 30 cps, with eight vertical dots and 640 columns, and produces color graphics.

• Commodore MPS 803 Dot Matrix Printer—The introductory dot matrix printer is for the Commodore 16 computer, and prints alphanumeric and graphic characters with a variety of styles and capabilities.

• Commodore 1531 Cassette—This cassette drive is aimed at users of the new Commodore 16 computer. It uses standard audio cassette tapes; digital tapes are not necessary.

• Commodore CM 141 Color Monitor—Designed to coordinate cosmetically with the Plus/4 computer, the CM 141 is also compatible with all of Commodore's computer equipment.

More Software For The 64

Commodore also had several new software announcements. Working with Marvel Comics and Adventure International, Commodore will release a series of six adventure games, called Questprobe, which will feature superheroes from Marvel Comics. The first game will be *The Hulk*, which will be available in August for both the Commodore 64 and the Plus/4 computers.

Videotex 64 is a new software package from Commodore for the 64, which will allow you to create business graphics or other pictures in high-resolution color and combine them with text for transmitting over regular telephone lines (using a VICMODEM) to other *Videotex 64* users. The package should be available by the time you read this (price to be announced).

Commodore also introduced a new educational software program, *Just Imagine*, for the Commodore 64. The package is designed to help children combine visual and verbal skills to create an animated story on the computer. *Just Imagine* is aimed at children from 4 to 14 years of age, and has increasing levels of complexity.

Finally, Commodore also introduced *B/Graph*, a charting and statistical analysis program for the Commodore 64, which is designed to analyze and convert any raw data into a graphic representation.

"The New Adam"

Coleco Industries, Inc., which a year ago made the biggest news at CES with its bundled Adam computer system, came to this year's show with the slogan, "The 1984 ADAM Is Ready."

The company says that any reliability problems that the Adam may have had have now been corrected, and a new six-month warranty program has been instituted to demonstrate Coleco's confidence in its machine.

Coleco announced the June shipment of a variety of new software programs for the Adam, including *SmartLogo*, a programming language; *SmartLetters & Forms*, a correspondence program; *SmartFiler* and *Recipe Filer*, which organize home data; and *SimpleCalc*, a spreadsheet. Educational and entertainment titles were also announced, including *Electronic Flashcard Maker, Brain Strainers, ExpeType, Zaxxon, Dukes of Hazard*, and others.

Among its hardware peripherals for the Adam, Coleco showed an additional digital data drive, which can store up to 512K of information on two data packs; a 5¼-inch disk drive; the AdamLink Direct Connect Modem, a 300-baud full-duplex modem with

Atari's CES Line-Up

Despite continued belt-tightening, layoffs, and other reorganizational measures, Atari, Inc., arrived at the Summer Consumer Electronics Show with an extensive array of software and a promise to deliver a high-end personal computer before the end of the year.

Banners on the press conference room walls and T-shirts handed out by Atari employees at the Summer Consumer Electronics Show (CES) in Chicago all said the same thing: "June 3, 1984—The Day The Future Began."

Atari chose that date (the first day of CES) as the kickoff for its new line of products. And if the company is to have a prosperous future—if it is to have any future at all—Atari officials know that the reaction to its CES line-up of software and hardware must be very good indeed.

Under President James Morgan's direction, Atari has been going through a top-to-bottom shuffle of people, products, and planning. After staggering financial losses during 1983 and smaller losses in early 1984, Morgan has cut hundreds of employees, closed some research and development facilities, and prepared the remaining Atari employees to be part of a wholly different sort of company—smaller, more focused on selected products, and, perhaps above all, a credible producer of what Morgan terms "interactive electronics."

To give meaning to those efforts, Atari's CES line-up was the strongest presentation of new products in quite a while from the Sunnyvale, California, company.

A New High-End Computer

First, Atari revealed that it will introduce a new high-end computer, reportedly for under $1000, in time for the Christmas buying season. The computer will be an extension of the XL line, very similar to the long-awaited 1450XLD. That computer was shown at two consecutive CES's, but is not going to be released.

No name has yet been given to the new machine (as of this writing). But Atari says that the computer will have 64K of RAM (random access memory) and a built-in, double-sided, dual-density disk drive that stores 352K of RAM (about 250 typewritten pages of information). The disk drive is said to work five times faster than current Atari models.

A data base, called *Atari Grapevine*, will be built into the new machine, as will an autoanswer and autodial 300-baud modem. The computer will also have an enhanced speech synthesis chip which will be capable of reading back words and phrases typed on the keyboard. Atari's new machine will be fully compatible with all Atari XL peripherals and software programs, say company officials.

Expansion System And Add-On Computer

Atari also reportedly plans to offer an expansion system for the new computer. The system will allow the computer to be expanded to 128K and to have some compatibility (about 70–80 percent) with other operating systems, such as CP/M (Control Program for Microcomputers), a popular business-oriented operating system, and MS-DOS (Microsoft Disk Operating System), on which IBM's PC is based. No price has been announced yet, but the expansion system should be introduced before the end of the year.

Although Atari did not officially exhibit the new machine at CES, company officials did conduct closed-door showings for third-party software designers to encourage development of a substantial software base as soon as possible.

Atari will also offer a new introductory computer which will be an add-on to the company's high-end 7800 ProSystem game console, announced in late May. The 7800 Computer Keyboard operates with 4K of RAM, expandable to 20K.

A line of selected computer software, including word processing, creative learning, and personal development programs, will be available for the new introductory computer. And the Computer Keyboard will be compatible with almost all of the Atari computer peripherals. The add-on computer should be available before the end of 1984, priced at under $100.

Atari unveiled several programs in a line of introductory computer software to be used with the 7800 ProSystem Computer Keyboard: *Atari Terminal*, a telecommunications cartridge; *AtariLab*, a science learning series; *Typing Tutor*, a tutorial typing game; *The Word Processor*; and *BASIC*.

The MindLink System

One of the most innovative products revealed by Atari at CES was its new MindLink System

headband, which uses slight electrical impulses from muscles in the forehead to direct the action on your computer screen.

Mindlink, which will be available for Atari's 2600 Video Computer System (VCS) and new 7800 ProSystem game machines this fall and for the XL computers in early 1985, will sell for a suggested retail price of $79. Included will be the headband, two infrared sensors, and a software package.

Atari plans three software packages for the unit initially—an adventure game, a new version of the popular *Breakout* videogame, and a relaxation, biofeedback program.

The headband is surprisingly sensitive. And despite the initial skepticism of some members of the trade press, the MindLink System proved easy to use (without having to wriggle your forehead, ears, or eyebrows to trigger it). There are also obvious long-range possibilities with this kind of technology for the physically handicapped who are not able to use conventional keyboards, joysticks, and other input devices.

Early Learning Software

Atari has teamed with child psychologist Dr. Lee Salk to develop early learning game software for toddlers (one to three years old) to use with the help of their parents. The first package, *Peek-A-Boo*, has eight levels of play and is supposed to help even very young children learn about cause and effect, spatial relationships, colors, shapes, and letters and numbers.

The new videogame cartridges developed by Dr. Salk and Atari will use the previously developed Atari Kids Controller, especially designed for small hands. *Peek-A-Boo* will be available on the Atari 2600 VCS and the 7800 ProSystem for a suggested retail price of $30, but will not be available for Atari computers.

Futuremakers Series For Older Children

Two computer software tours of space, *This Is Ground Control* and *Through the Starbridge*, were introduced as the first products in Atari's new Futuremakers series.

Featuring 3-D animated graphics of planetary approaches and fly-bys, the programs are aimed at users ten years of age and older.

The Futuremakers series should be available on disk for Atari computers about the time you read this, for a suggested price of $39.95 each.

An Abundance Of Software

Atari's CES announcements also included the following products:

• Milestone Series: The Atari Learning Systems group has put together what it considers the best in home computer educational packages not only from Atari, but within the entire industry. Milestone Series software will be not only for Atari computers, but for other systems like Commodore, IBM, and Apple as well. Suggested retail price for most of the packages is $34.95, except where noted differently.

AtariLab Starter Set ($89.95) and *Temperature Module/Light Module* are aimed at children from 4 to 12 years of age, allowing the user to conduct more than 100 experiments.

Yaacov Agam's Interactive Painting (price to be announced) is a package developed by well-known kinetic artist Yaacov Agam, which helps the user generate original art by computer.

Find It! is a series of visual perception activities and puzzles for children of all ages.

The ABC of CPR is a home health library of software, the first module of which, *First Aid*, is a two-part tutorial and simulation designed to help build awareness and background information for handling first-aid emergencies.

Wheeler-Dealer is an economic supply-and-demand game which lets the player learn how to handle a wide variety of business problems while becoming an auto industry magnate.

Simulated Computer, which shows what goes on inside a computer, and *Telly Turtle*, a pictorial pre-Logo version of the turtle graphics concept, are two more titles in the Milestone Series.

• Atari Educational Software: Two new software packages for children 4 to 12 years old were introduced, *Letter Tutor* and *Word Tutor*, priced at $39.95 each.

Atarisoft's New Titles

Atarisoft, the third-party software publishing division of Atari, Inc., announced numerous titles for the Commodore 64 and VIC-20, the Apple II family, and the IBM PC.

Also, for the IBM PCjr, Atarisoft has introduced *Centipede, Donkey Kong, Moon Patrol,* and *Pac-Man.*

Suggested retail price for the Atarisoft games on disk is $34.95, and for cartridge, $44.95.

7800 ProSystem Exhibited

Atari also showcased its new high-end 7800 ProSystem videogame console ($150 retail), which Atari says has the most advanced color graphics of any home computer or videogame currently available.

AdamLink telecommunications software; a 64K Memory Expander, which increases the memory capacity of the system from 80K to 144K RAM; the Adam Universal Interface, which has an RS-232 connector and a Centronics parallel connector which allows a user to connect peripherals and accessories from other manufacturers; and an accessory kit which includes replacement daisywheels, cartridge ribbons, tape head cleaner, and a blank digital data pack.

Sinclair's QL Computer

Sinclair Research Ltd. also showed a new computer at CES, the Sinclair QL, which was introduced in London in January and is scheduled for introduction in the U.S. this fall at a price of $499.

The QL has 128K of RAM and is based on the Motorola 68000 microprocessor, the same microprocessor family used with the Lorraine and the Macintosh. There are two built-in 100K microdrives for mass storage, a 65-key keyboard, and the entire unit weighs just three pounds.

The QL will come with a built-in BASIC as well as four applications software programs: *Quill*, a word processor; *Abacus*, a spreadsheet; *Archive*, a data base; and *Easel*, a graphics package.

Sinclair will sell the QL by mail order in the U.S., and hopes to be receiving orders in time for the Christmas buying rush.

The Software Boom Is On

While computer hardware innovations continue to move the microcomputer world along at a dizzying pace compared to most industries, it is software growth that is sparking some of the greatest interest among consumers and computer industry entrepreneurs.

While innovations in personal computer software may not be as plentiful as some critics would like, there is no question that the caliber of current software is far ahead of the programs produced even a year ago.

Here are some of the more impressive programs introduced at CES:

Muppet Learning Keys (Koala Technologies)—A computer peripheral for the Apple IIe, IIc, and the Commodore 64, *Muppet Learning Keys* is a computer keyboard which simulates the familiar contents of a child's school desk to help youngsters learn basic skills.

Aimed at preschoolers, the package is a three-pound keyboard which parents can help their children use to learn the alphabet, numbers, colors, and shapes. The keyboard has equivalents to such school-desk standards as a ruler, watercolor set, penmanship slate, compass, eraser, and arithmetic exercise book.

Developed in association with Henson Associates, Inc. (creators of the Muppets), and Sunburst Communications, Koala's *Muppet Learning Keys* incorporates Kermit the Frog, Miss Piggy, Gonzo, and other Muppets to provide friendly instructions. The package will sell for $79.95.

Koala Technologies announced the Muppet Learning Keys, *a child's computer keyboard which uses the popular Henson Muppets to help preschoolers learn about the alphabet, numbers, colors, and shapes.*

Commodore, Apple Science Fiction-Based Software

Trillium Science Fiction Series (Spinnaker Software)—Trillium, a new division of Spinnaker, is producing a series of interactive computer adventure games inspired by the novels of a number of best-selling science fiction authors.

The first six titles, all of which will be available in August or early fall on disk for the Commodore 64 ($39.95 each) and the Apple II family ($44.95), are: *Rendezvous With Rama* by Arthur C. Clarke, *Amazon* by Michael Crichton, *Dragonworld* by Byron Preiss and Michael Reaves, *Starman Jones* by Robert Heinlein, *Fahrenheit 451* by Ray Bradbury, and *Shadowkeep*, from which a book has been adapted by Alan Dean Foster.

In these adventure games, the player assumes the identity of the characters. The scripts are professionally written—often with the direct collaboration of the original author—and the games feature high-resolution color graphics. Graphic clues are used, hints are available to reduce frustration, and some contain arcade-style games within the adventure games.

Michael Crichton, for example, wrote the entire script for *Amazon* and worked with the game

A screen from Amazon *by Michael Crichton, one of Spinnaker's new Trillium interactive adventure games based on popular science fiction novels.*

Brøderbund's Dazzle Draw *makes use of the graphics power of the Apple IIc.*

designers. Arthur C. Clarke worked out an ending to *Rendezvous With Rama* for the game.

Trillium plans other games based on science fiction novels by Phillip Jose Farmer, Roger Zelazny, Alfred Bester, and Harry Harrison.

An Apple Adventure

Robot Odyssey I (The Learning Company)—Following on the heels of its very popular educational package, *Rocky's Boots*, The Learning Company is introducing an educational adventure program, *Robot Odyssey I*, designed for teenagers and young adults.

Trapped in an underground world populated by robots, you can only escape by constructing your own robots. The program provides a game format while at the same time teaching the player how to design integrated electronic circuitry, the fundamentals of building robots, and the basics of Boolean logic. There are an Innovation Lab and helpful tutorials which offer step-by-step instruction for players.

Robot Odyssey I is the first program in The Learning Company's new DigiWorld Series, software that combines the interest in adventure gaming with the learning environment of a computer construction kit. The new program will be available for the Apple II family of computers at $49.95.

Magic Spells And Monsters For Commodore, Atari

Archon II: Adept and **Skyfox** (Electronic Arts)—One of the most popular of Electronic Arts' titles has been the original *Archon*, a chess-like strategy game that also includes videogame action. Free Fall Associates, the designers who created the game, have expanded on that theme for this sequel, *Adept*.

Magic reigns in this game, with players conjuring spells, summoning monsters, and winning playing squares through individual combat. The new game gives players an expanded arena in which to fight, with more options and an altered game landscape. For a suggested retail price of $40, *Archon II: Adept* is available for Atari computers (with 48K) and for the Commodore 64.

Apple IIc Graphic Punch

Dazzle Draw (Broderbund)—A complete illustration package created especially for the Apple IIc's double high-resolution capabilities and 128K of memory. The software is icon-based, much like the Macintosh. One unique tool in the package is the ability to draw with pen strokes of various shapes and sizes in 16 colors. *Dazzle Draw* can also be used with a 128K Apple IIe with an 80-column card, a Revision B board, and one disk drive. Suggested price is $49.95.

Activision's Commodore 64 Programs

Zenji and **Toy Bizarre** (Activision)—These two new games are among a dozen titles that Activision is releasing for the Commodore 64 computer. *Zenji* is an intriguing strategy and puzzle game with a distinctly Eastern flavor. The player must connect a glowing maze of elements (the Many) to a pulsating source (the One) to create a single unified green image, or "Zenji." The play is rapid and demanding.

Toy Bizarre lets you control a character named Merton, who awakens in a toy factory which has gone berserk, as gangs of tyrannical toys attempt to take over the shop at midnight. ©

Gerard K. O'Neill

Selby Bateman, Features Editor

"Keep it simple, and make it work" is the informal motto at Gerard O'Neill's Geostar Corporation, a computer-based satellite positioning and communication company on the outskirts of Princeton, New Jersey.

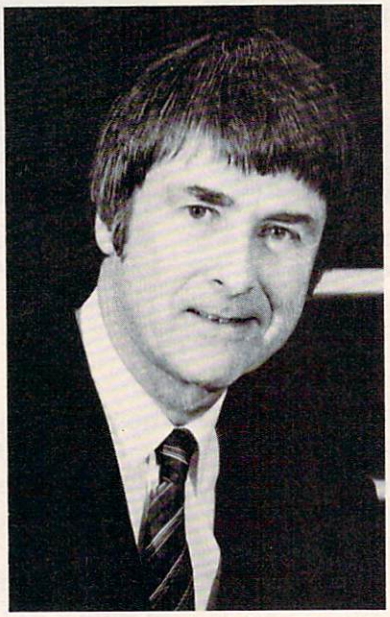
The motto is characteristic of O'Neill, a leading physicist, author, and high-tech entrepreneur, who has a reputation as a visionary scientist with a knack for seeing to the heart of complex issues.

His first major scientific contribution came in 1956 when, as a 29-year-old Princeton physics instructor, he developed the storage-ring technique for colliding particle beams. The technique has become standard for subatomic particle accelerators in the field of high-energy physics.

In his three books, The High Frontier: Human Colonies in Space; 2081: A Hopeful View of the Human Future; *and most recently,* The Technology Edge: Opportunities for America in World Competition, *O'Neill has explored the possibilities of space colonies, satellite communications, computers, and the challenges facing the United States in its economic and technological development.*

For The High Frontier, *O'Neill received critical acclaim and captured the popular imagination with his simple, feasible plan for the development of space colonies. He also founded—and is president of—the Space Studies Institute, a privately funded organization which has done much to further the goals of space exploration.*

Among the more arresting concepts he developed in The High Frontier *was the mass-driver transport device, a device with small buckets on a recirculating conveyor belt driven by magnetic impulses. The device could be used to efficiently eject mined lunar raw materials into space, propelling them to a space station under construction.*



Dr. Gerard K. O'Neill

His latest book, The Technology Edge, *addresses six "hot" technologies which O'Neill believes are crucial emerging industries: microengineering, robotics, genetic engineering, magnetic flight, family aircraft, and space science. If the U.S. does not compete successfully in these areas, he warns, it will lose the technological and economic leadership it has enjoyed.*

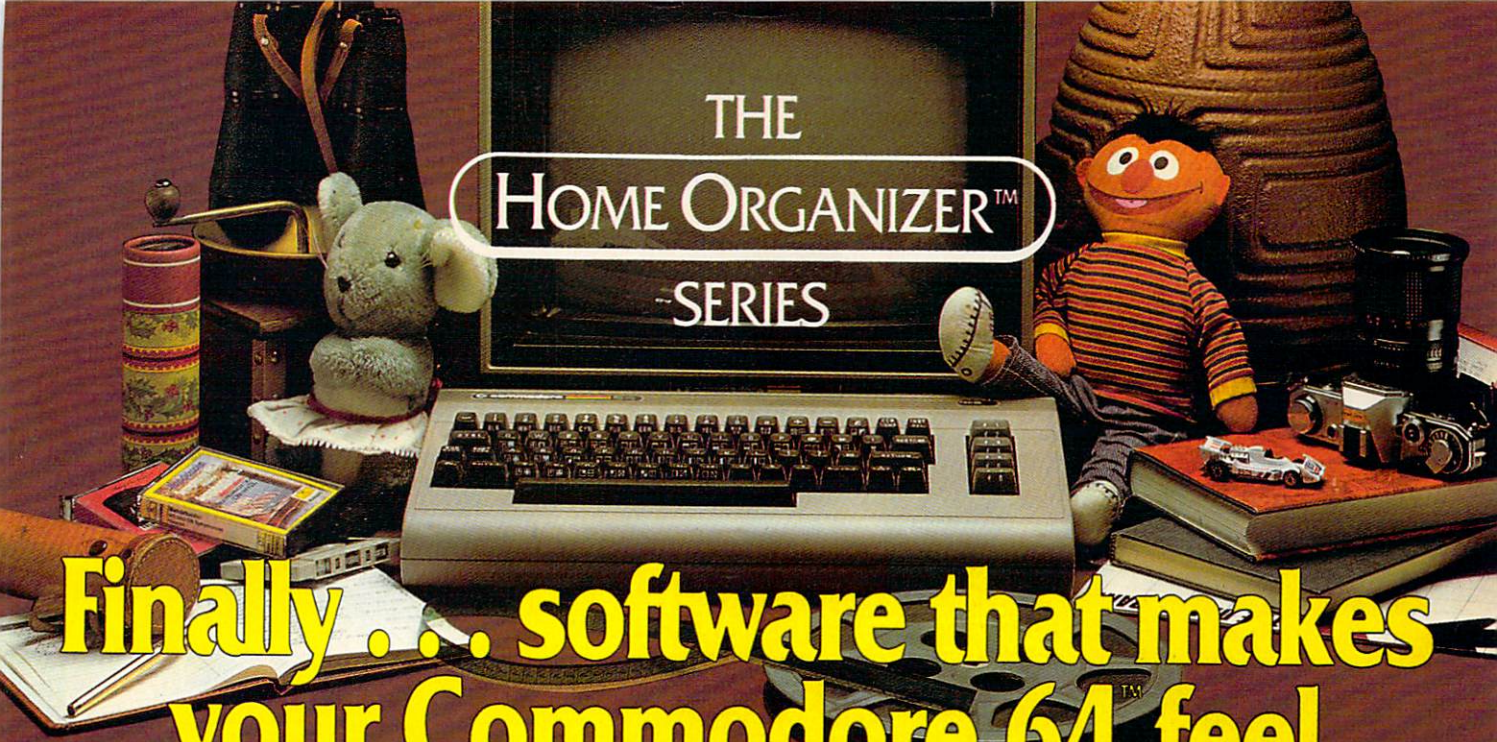
Despite his many other interests, it is the Geostar Corporation which currently occupies most of O'Neill's time and effort. Geostar, a development firm concerned with communication and navigation via satellite, is a perfect blend of O'Neill's farsighted vision and his make-it-work practicality.

The system which O'Neill and his colleagues are developing could revolutionize how we track and monitor aircraft and how we communicate with one another. Initially, the proposed system would have three satellites in geosynchronous orbit over North America. The Geostar central computer facility would use the satellites to route tracking and communication data almost instantaneously for everything from commercial airlines to trucking companies, taxi services, police departments, and even individuals. The key to the system will be a hand-held transceiver which can send and receive messages through the Geostar network.

During the interview, he remarked that an airplane thousands of feet above Princeton was in the process, at that moment, of testing the Geostar system.

An articulate and engaging conversationalist, O'Neill is interested in how microcomputers are affecting our society. He keeps a well-used Apple II Plus within easy reach of his desk. On the day he spoke to COMPUTE!, *O'Neill had been using a new Apple IIc to test the portability of his II Plus programs to the new machine.*

C!: A number of Japanese computer companies are now getting behind what's called the MSX operating system standard. And that will probably be introduced sometime soon in this country. Do you think that in the U.S. we will see a standard operating system?



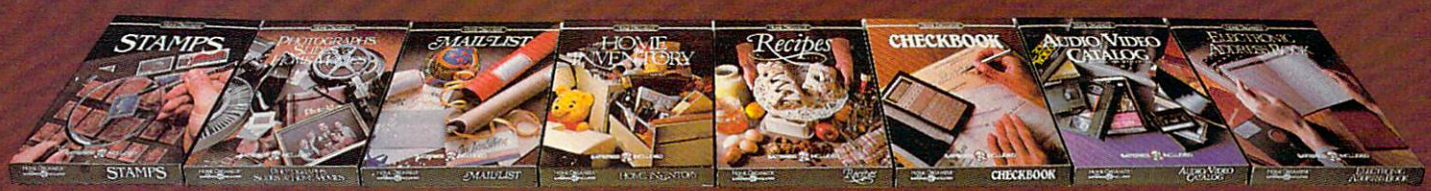
Finally . . . software that makes your Commodore 64™ feel comfortable at home

Now your home computer can help you cook, keep your accounts, find an address or keep track of your record and book libraries—with first-class software specially tailored for the home environment.

The Home Organizer™ series includes a wide range of separate and individual programs for different activities like stamp collecting, personal banking, or home photo and movie collections. Each one is pre-programmed with a "page" format planned out by experts to make it easy for you to store and retrieve the information you'll want for your special activity. You don't have to program anything yourself. Just load the disk and start feeding in your data.

If you're used to run-of-the-mill home computer software, the speed and simplicity of the Home Organizer™ series will surprise you. Each program is written entirely in "machine language", the most basic computer code. So they search, sort and analyze your data with amazing speed.

The Home Organizer™ is fast enough to sort through your household belongings in seconds, yet so simple the children can use it to look up a phone number. Choose any or all program modules that fit your needs. They make ideal gifts, too!

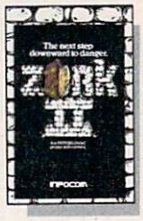


BATTERIES INCLUDED

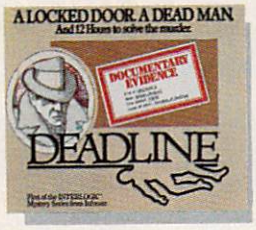
"Excellence in Software"

For a full color brochure write to:
Batteries Included, 186 Queen Street West, Toronto, Canada M5V 1Z1 (416) 596-1405 / 3303 Harbor Blvd., Costa Mesa, CA. 92626 (714) 979-0920


Your greatest challenge lies ahead and downwards.
ZORK I
INFOCOM

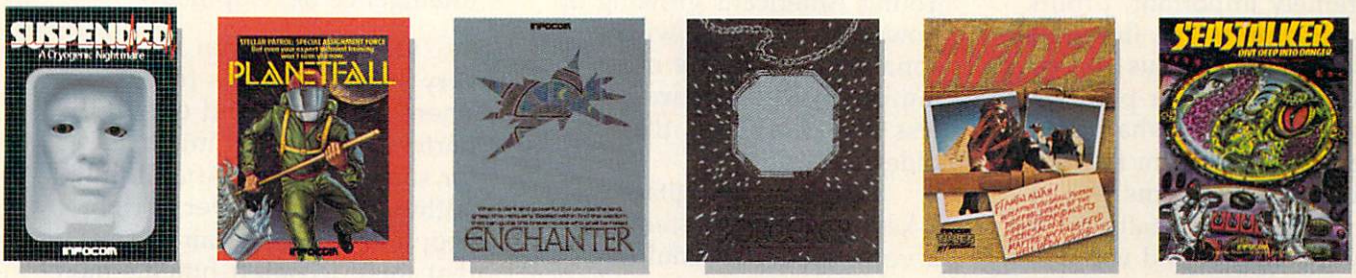

The next step downward is danger.
ZORK II
INFOCOM


It all comes down to this.
ZORK III
INFOCOM


A LOCKED DOOR A DEAD MAN
And 12 Hours to solve the murder.
DOCUMENTARY EVIDENCE
DEADLINE


The WITNESS
SOMEBODY'S GOING TO TAKE THE DEEP SIX!


STARCROSS
INFOCOM



THE INCOMPLETE WORKS OF INFOCOM, INC.

Incomplete, yes. But it's not just because we're always bringing out new stories in the Infocom interactive fiction collection. Nor is it simply due to the fact that with all the writing and re-writing, honing and perfecting that we put into every one of our stories, our work is seemingly never done.

The real reason is: an Infocom work of fiction can never be complete until you become a part of it.

You see, as hard as we work at perfecting our stories, we always leave out one essential element—the main character. And that's where you enter in.

Once you've got Infocom's interactive fiction in your computer, you experience something akin to waking up inside a novel. You find yourself at the center of an exciting plot that continually challenges you with surprising twists, unique characters (many of whom possess extraordinarily developed personalities), and original, logical, often hilarious puzzles. Communication is carried on in the same way as it is in a novel—in prose. And interaction is easy—you type in full English sentences.

But there is this key difference between our tales and conventional novels: Infocom's interactive fiction is active, not passive. The course of events is shaped by the actions you choose to take. And you enjoy enormous freedom in your choice of actions—you have hundreds, even thousands of alternatives at every step. In fact, an Infocom interactive story is roughly the length of a short novel in content, but because you're actively engaged in the plot, your adventure can last for weeks and months.

In other words, only you can complete the works of Infocom, Inc. Because they're stories that grow out of your imagination.

Find out what it's like to get inside a story. Get one from Infocom. Because with Infocom's interactive fiction, there's room for you on every disk.

INFOCOM™

Infocom, Inc., 55 Wheeler Street, Cambridge, MA 02138

For your: Apple II, Atari, Commodore 64, CP/M8", DECmate, DEC Rainbow, DEC RT-11, IBM PC* and PCjr, KAYPRO II, MS-DOS 2.0*, NEC APC, NEC PC-8000, Osborne, Tandy 2000, TI Professional, TI 99/4A, TRS-80 Models I and III.

*Use the IBM PC version for your Compaq, and the MS-DOS 2.0 version for your Wang or Mindset.

O'Neill: The whole issue of having computer programs that remain usable as you go forward in time—usable for the individual person—is I think extremely important. I think manufacturers are, first of all, being far too callous and far too arrogant with their potential customers about what they've been doing to them in the way of operating systems and programs. Fundamentally, if you buy a program and use it and then want to go over and buy somebody else's—or somebody else wants to sell you a computer, say—I think that the first question that they should be able to answer positively is the question: Will your new computer run all of the programs I'm used to?

Now, they can tell you "We've got a whole bunch of other programs which are much more powerful" and have all kinds of bells and whistles and all of that. Fine, nothing wrong with growing. But they should also be able to tell you that, by the way, it will run all of those programs that you had before.

As machines get more powerful in terms of processor capability and memory capacity and so on, it's not that tough to do it. I would say any manufacturer who sets up a general policy of making equipment that will run anybody's programs is sure going to get my business and my owner loyalty forever. The problem is that up to now manufacturers have not even been compatible within their own product lines.

C!: There are predictions that by 1988 some 50 million homes in the U.S. will have personal computers. In what ways do you see this increased awareness of computers affecting America's technological edge in the world?

O'Neill: I think it will help a lot. It's already true, just because of the accident that we work on an alphabet and the Japanese work, of course, with a character-based system, that we as a people are far more familiar with keyboards than they are. Young Americans growing up nowadays, working with personal computers, are much more familiar with keyboards, much less scared of them, than the older generation.

Geostar is a digital system, a keyboard-type system. It's not a voice system. It could be connected to a personal computer anytime. The message transfer capability is entirely consistent with the kind of telecommunications that you like to carry out with your personal computer, or from a portable computer. And, of course, by 1987, today's three or four pound computers that fit in a briefcase are probably going to be shrunk down to a quarter of an inch thick. You can carry those along with a Geostar transceiver, and be in instant touch with anywhere.

> "Manufacturers are ... being far too callous and far too arrogant with their potential customers...."

C!: In the U.S., companies like Apple and IBM and other microcomputer companies are very competitive. There is very little ability to travel from one system to another . . . .

O'Neill: Yeah, that's a sore point with me. I get very exercised over it.

C!: Artificial intelligence is another area in which the Japanese are showing a great deal of concerted effort, just as they are in robotics. What's your view of the pace of change in artificial intelligence development?

O'Neill: You run into some very strong opinions there. There's been a band of supporters for artificial intelligence for some 25 years. And all in all (although they are very bright people) I think it's fair to say that their accomplishments have been substantially less than they were advertising when they started.

It's a very tough subject. One of the fundamental reasons why it's so tough is that if you really want to have machines that think like people, you have to go back to the beginnings of how computers were designed. You don't want a serial, bit-based machine. You need to have a machine which somehow can carry out the associative function of the human brain. Which is a function that we have very little understanding of. You know, we do not understand the associative function of the human brain nearly as well now as we understood binary arithmetic five thousand years ago. So, it's not just a question of how to design a computer, it's to even understand the problem well enough to know how to start it. I think there are sure to be some very exciting developments in artificial intelligence over the next fifty years, but I'd be surprised if they come out of classical computer design of the kind that we're used to now.

C!: In the field of microengineering, we're beginning to see more interest in what are called "biochips"—computer circuits that one day might be based on biological molecules. There are even a few biochemists who feel biochemical engineering may lead to

UNLEASH THE POWER OF YOUR IBM PERSONAL COMPUTER WITH COMPUTE!'s PC & PCjr.

COMPUTE!'s PC & PCjr is the new magazine from *COMPUTE!* to help you unleash the power of your IBM at home, school and office.

In every issue, you'll discover ready-to-type, ready-to-run program listings—including games, educational programs and utilities. Exciting sound and graphics applications. How-to articles for hobbyist programmers. In-depth product reviews. News of the latest developments in the personal computer industry. Expert advice and answers to your technical questions. In short, everything you need to get the most from your IBM computer.

Program Power. *COMPUTE!'s PC & PCjr* Magazine is for people who want to *use* their computer—not just read about computers. You'll find at least half a dozen ready-to-run programs for your PC or PCjr every single month!

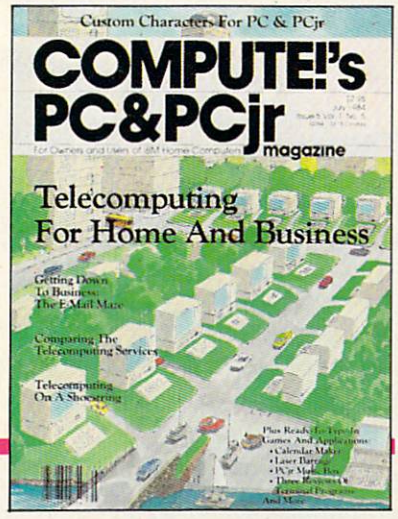
Money-Saving Power. We show how to make your time with your computer more productive. How you can run office programs on your home system. Organize a small business. You'll even discover programs to help you make smarter investments, project your IRA earnings, evaluate tax deductions, and more.

Buying Power. Every issue contains fair, objective reports on the newest software and hardware to hit the market. Plus balanced software and hardware reviews that help you decide.

Game Power. Adventure games and action games, puzzle games and educational games. Escape your captors in "Martian Prisoner." Defend the Earth in "Aardvark Attack." Solve the puzzles of "Mind Boggler." Entertain your children and sharpen their skills with "Munchmath" and "Word Match."

Learning Power. Want to write your own programs? We can help you with programming tutorials and descriptions of how our programs work. Plus a regular monthly column for BASIC beginners.

Telecommunications Power. Learn how your IBM can telecommunicate with other computers over ordinary phone lines. Link up with your office or school computer. Access information services such as Dow Jones, CompuServe, and The Source. Even leave messages in our electronic mailbox!



Save 33% as a Charter Subscriber!

☐ **YES.** Start sending me the power of *COMPUTE!'s PC & PCjr.* Please enter my Charter Subscription at just $24 for 12 issues—a 33% savings. I understand I can cancel at any time for a full pro-rata refund.

Name_____

Address_____

City_____ State_____ Zip_____

☐ Payment enclosed ☐ Bill me

Charge ☐ Visa ☐ MasterCard ☐ American Express

Account No._____

Exp. Date_____

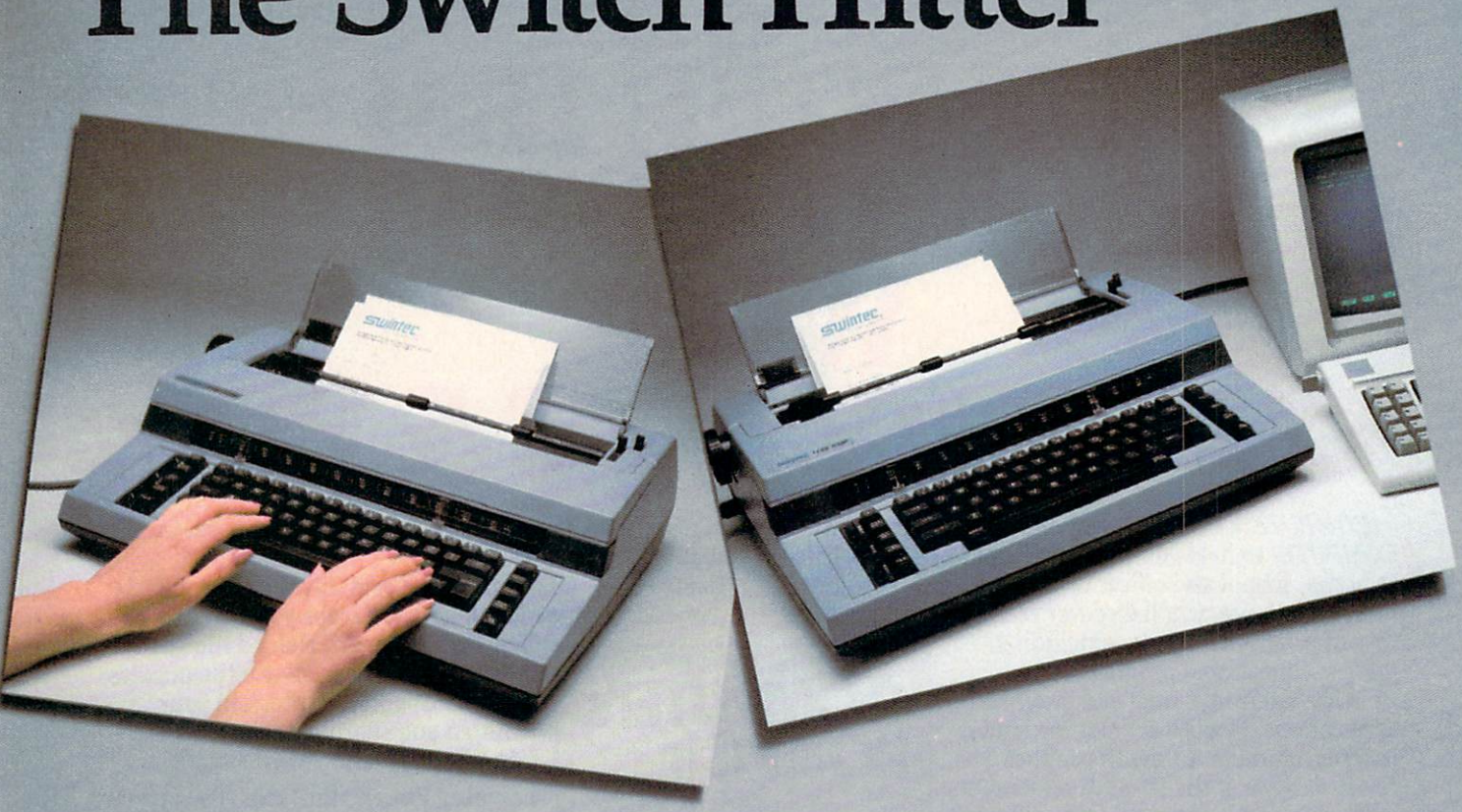
Mail to:

COMPUTE!'s PC&PCjr
P.O. Box 974
Farmingdale, NY 11737

To start bringing home the power of *COMPUTE!'s PC & PCjr,* mail the coupon or the postpaid card bound into this issue.

Act now and you'll have access to powerful Charter Subscriber savings. You'll receive 12 issues for only $24—33% off the newsstand price! And your satisfaction is guaranteed or we'll refund your money on all unmailed issues.

Introducing The Switch Hitter



The computer compatible electronic typewriter from Swintec.

Now there's a full-featured intelligent typewriter that can do double office duty as a letter quality computer printer . . . with just the simple flick of a key switch!

The 1146CMP electronic typewriter from SWINTEC has built-in interface capabilities which allow it to function with virtually *any* computer. The computer compatible capability of the 1146CMP makes it possible for your computer to perform word processing and print-out functions easily.

As a typewriter, the 1146CMP has all the electronic features needed for efficient typing: 46 characters of correction memory, a relocation key for quick positioning after correcting, automatic paper feed, both forward and reverse indexing, and repeat capability on all keys.

Best of all, the 1146CMP is extremely easy to use. No special training is required to perform either typing or word processing operations. And the 1146 CMP is available, complete with computer interface, for under $600.*

The computer is basic to the fully automated office; SWINTEC computer compatible typewriters bring office integration one step nearer.

Switch to SWINTEC. . . and get our Switch Hitter on your team.

*Also available as 1146CM, without interface, at an even lower price

I want to know more about the SWINTEC 1146CMP

☐ Send me more information

☐ Have a SWINTEC dealer contact me

NAME _____ TITLE _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE _____

4-23 C-8-84

swintec™ CORPORATION

23 Poplar Street
East Rutherford, NJ 07073
800/225-0867

some analog, rather than digital, biologically based microchips.

O'Neill: You have to be careful about that, because one of the things we've discovered about genetic hardware is that when you really get down to the level of the way that cells work, they are binary, they are digital. They are not analog devices. The numbers of neurons acting and so on get to be big enough so that you can see what appear to be analog signals, but when you really get down to the level of how any living organism works, it is a very mechanistic system which is much more digital than analog. And the way the genes and the various templates fit together in genetics is a very rigid, very digital structure.

So, I think we'll be following a dead end if we think that by kind of retreating to analog type systems we've somehow improved our chances of going into artificial intelligence. I don't think that's where the key is. But the notion that you don't do things in a linear, serial fashion, but that you do them in an associative fashion, with all kinds of branches—that is fundamental to artificial intelligence.

C!: Do you think we'll see the widespread use of domestic or personal robots in American homes in the next 20 years?

O'Neill: I think so. It's much more likely that the first practical household robot will grow out of hobbyist activities than that it will grow out of the activities of some large organized company. It's the sort of thing that's going to take some fanatic working in a basement or garage to do right.

C!: What forms will these robots have?

O'Neill: It's really a question of where the market is. If you look at robots so far, if you had to try to characterize the successful ones in a single sentence, you would say that they are mechanical single arms. The next step is probably some degree of mobility. But I would guess, to be quite honest, that they are going to turn out to be applications of robotics that don't really stretch the art at all, that will have a profound impact on markets. For example, if you take something like a McDonald's hamburger place, and put relatively low-level industrial robots in there, you may do a better job of making hamburgers.

C!: What's the status of your Space Studies Institute?

O'Neill: I forget offhand the number of months in which its membership has doubled. But it's growing rapidly.

> "It's the sort of thing that's going to take some fanatic working in a basement or garage to do right."

The Institute is just receiving now the results of a two-year study that was carried out under SSI (Space Studies Institute) funding by Rockwell International on the chemical processing of lunar soil. This is the first time that the actual wet chemistry has really been done—where people have put chemicals in test tubes. And that has come out very favorably.

The mass-driver research, which was also Institute funded, has now progressed so well that it has now gone into a computer phase. The mass-driver-three design was basically worked out on an Apple computer. And then through that computer-aided design, the mass-driver-three model was built. It obeyed the CAD/CAM (computer-aided design/computer-aided manufacture) design within one percent. The next phase is to go back to the computer and say, OK, now that you've had this cross-check, let's go ahead and design a complete lunar catapult.

C!: Do you still see a mass-driver as the best vehicle for movement of materials in setting up the first space colony?

O'Neill: Absolutely. It's going so well that it's not expensive for us right now because we're not having to build an elaborate test model. We've done that.

C!: When do you estimate that it will be feasible?

O'Neill: We have some rather close estimates on that because the Institute is on a five-year research program, which by coincidence will conclude in 1987. At that time we expect to have an overall plan to publish which will have every essential technical building block for space industry at least to the benchtop or pilot plan stage of application. And then to go from there to the point of economic productivity is roughly a five-year program. When that five-year program begins, of course, depends on when somebody buys into it to the level of funding that will be needed.

There is a new research program of the Institute within that five-year plan, and we're just about to award the contract for it now. And that will be for the design of a solar-powered satellite specifically to be manufacturable out of lunar materials. That's never been done before and obviously needs doing. And the Institute has the money accumulated from its

members and its Senior Associates to do that work and has gotten the bids in for it and is just about to award a contract.

C!: Despite the explosion in high technology we're experiencing, our space program seems to have a lower national visibility now than in past years. Why is that, and what does it mean for the future?

O'Neill: I think it's a correct perception that the national awareness of it has decreased. Although interestingly enough, all of the surveys that have been done indicate that the national support for a strong space program is broader based now than it ever was before. I think the reason that it's in low visibility is that there is no really very exciting program that NASA has. And we regard the work that we're doing as essentially independent of NASA, although the Institute's work is clearly based on taking all of the technology that has been developed in the first 25 years of the space program and is being developed right now.

C!: Is the most exciting work being done by private companies?

O'Neill: Well, I'm a highly biased source. I think the most significant thing going on is the research that the Institute is funding. That's why we're doing it. If we thought there was something else that had higher potential payoff, that's what we'd be doing.

C!: When you first began advocating colonies in space 15 years ago or more, you had a certain view of the potential it would have and how quickly it might come about. Has that changed at all?

O'Neill: No, it really hasn't changed. The main difference is that I thought of it then as naturally a governmental program because the scale of funding that was required appeared to be very big. As the result of, first, the five years or so of work that I did on my own, and then the ten years of work that has been done with a lot of people involved, it all looks a lot simpler and a lot smaller in scale than it did 15 years ago. Now it looks as if the action program to move out into space and use the energy and materials there in a productive way is probably a seven or eight billion dollar program instead of a 200 billion dollar program. So it's in the scale of projects which have been privately financed. And I think that sometime in the late 1980s, there could be some very exciting, creative new developments in putting together a financial package of that kind which I would think of as probably being done on a consortium basis by a number of companies.

Although most people are not aware of it, the long-term result of the developments of the kind that the Space Studies Institute has been supporting is obviously human habitation in space and the movement out onto the high frontier. There has been since last October a very nice exhibit on that subject, which is easily accessible; namely, the General Electric Horizons Pavilion down at the Epcot Center in Disney World. That's about a 70 or 80 million dollar exhibit with a fantastic ride through four communities of the future. And the one that gets the lion's share of the attention and the time is the space colony, which is very accurately based on blueprints that were supplied to General Electric and to the Disney Enterprises by the Space Studies Institute. So, people who want to see in a very easy way in a few minutes what the long-term potential is there, should go and visit the Horizons Pavilion. And I sure hope they would come out of it wanting to support SSI.

C!: Do you think you will ever go into space?

O'Neill: [Laughs] Sure hope so.

C! You have a reputation as a scientist and as a writer; as someone with an ability to see through to the core of a problem or an opportunity . . . .

O'Neill: I'm glad you see it that way. Not everybody has been that kind [laughs].

C!: How do you handle the inevitable frustrations that occur when the pace of advancement lags, say, in space exploration?

O'Neill: I don't think that it bothers me very much as long as I feel that I'm taking productive action to make things happen as fast as possible, rather than trying to fight with a governmental system which is fundamentally pretty unresponsive. I just find it very much more rewarding, in terms of personal satisfaction, to be a part of the Institute's effort. We're doing it on our own.

C!: In *The Technology Edge*, you put a great deal of emphasis on the fact that the U.S. is going to have to compete to stay in the lead . . . .

O'Neill: Absolutely.

C!: Yet, at the same time, I sense that you have a feeling that international cooperation is desirable in the long run both technologically and economically. How can we achieve both of those goals?

O'Neill: It's a good question. The best thing, of course, is always to go by historical example. Most of the important openings up of economic opportunity, the exploitation of economic opportunity, have occurred in a competitive fashion either privately or governmentally. And the space program is a classic example of that, even to the point where in Japan there are two different, competing space programs.

Biochips:

A Revolution In The Making

Selby Bateman, Features Editor

The silicon chip—the wafer-thin foundation of the computer world—may someday be replaced by a microscopic organic "biochip." Based on today's pioneering work in biotechnology, the biochip would be far smaller, faster, and more powerful than its silicon predecessor.

In the not too distant future, Silicon Valley may have to change its name to Protein Valley.

There's still plenty of time to work out the details of the name change. But there are already indications that the limitations of silicon-based transistors will drive the computer industry to a more effective and potentially more powerful technology. And a few biophysicists and electronics engineers are betting that the eventual winner will be in organic molecules manipulated to form superefficient microchips.

Microscopic Switches

In essence, these biochips would be microscopic switches which would transmit electrical impulses in much the same way that silicon-based chips operate today. The work is now in its earliest stages, and there are many who are skeptical of the long-range practicality of such technology. But small groups of optimistic entrepreneurs are spending millions of dollars on research to demonstrate the feasibility of the idea.

"The prospects for support are excellent both from the government and the civilian sector," says biophysicist James H. McAlear, whose Gentronix Laboratories is a leader in the field.

McAlear and partner John M. Wehrung, an electronics engineer, have already discussed possible applications for biomolecular electronics with officials at the Pentagon and the United States Information Agency (USIA).

Heavy Traffic

Why even consider abandoning the silicon chip?

One major reason stems from the amazing pace of technological change in the computer field. As computer chips have evolved, they have rapidly gotten smaller and less expensive. At the same time, the number of operations required per chip has leapfrogged upward.

A microchip itself is little more than a grid of silicon tracks—sophisticated electrical relays—through which current flows or doesn't flow depending on what operations are being performed. The problem with silicon chips develops when more and smaller tracks are crammed onto a chip. At a certain point, the electrons racing along these paths begin to adversely affect each other in what is called *cross-talk*. The circuitry is also prone to overheat as the electrons start to lose energy during their travels.

Theoretically, organic molecules and specially developed proteins would offer none of these problems. They could also be manipulated to create the on-or-off, binary gates that form the basis of today's digital computers.

Practical applications are likely to include products ranging from erasable laser disks to molecular memory devices, for example. But the first applications may well be in sensing devices.

"Noses On A Chip"

"Noses on a chip" is what Daniel Hillis calls these sensing devices of the future. Hillis heads an artificial intelligence company in the Cambridge, Massachusetts, area that is also rumored to be working on research related to biochip technology.

"Chemo-detectors will be the first area of practical application for biochips," he says. "A chemical detector is basically for smelling.

"For instance, imagine if your wristwatch warned you when you needed to take a shower because you smell funny," he says.

The biotechnology industry has so far achieved its greatest visibility through such recent microengineering feats as genetic cloning. But over the next decade—with technological advances expected to continue at their frenetic pace—biomolecular research on computer microchips and spin-off applications will likely achieve both increased popular awareness and financial backing.

So, let's say we open up the opportunity for magnetic flight systems. I think you're going to find competitive construction of magnetic flight systems in a number of different countries and by different companies within the same country. All it takes is for the opportunity to be perceived, and everybody wants to jump in.

The same thing is going to happen in light aircraft construction. The same thing is going to happen in space. So I don't see it as being an orderly international cooperative program to move in a logical fashion into space. It's going to be a disorderly, helter-skelter, competitive thing. It's just the way human beings do things. And oddly enough, it's probably the most effective way. Part of the reason for that is that very large structures tend to be inefficient and bureaucratic by their very nature. I have seen international cooperative organizations in science working, and they are some of the worst bureaucracies you could ever find. Groups of impassioned young scientists working away to try to make something happen are far more effective per dollar spent than these huge cooperative international programs.

On the other hand, there are certainly examples internationally of operations which are generally perceived as useful, and so naturally worldwide in scope, that they do become effective international programs which cross all ideological boundaries. Intelsat is one example of that.

Where could that sort of thing happen again? I would guess that there would be coordination in setting up solar power satellites in synchronous orbit; coordination to minimize interference with radio systems and so on. I would not expect that it would go to the point that all the solar power satellites would be built by the same entity. I think there would be a number of different competing entities from different countries making them. The saving grace is that solar power satellites are fundamentally a peaceful technology.

C!: What kind of support are you finding for Geostar?

O'Neill: It's been very positive so far. All the heads are nodding together. Many, many industries have come to us and said that we are going to help them a lot. In fact, it's amusing. Many industries knock on our door, and the guys come in and say, "How did you know to design a system that is exactly what we've been looking for?"

"The saving grace is that solar power satellites are fundamentally a peaceful technology."

The land transportation industry, trucking companies, police departments, fire departments, taxi services . . . .

C!: And in the long run?

O'Neill: In the long run, anybody.

C!: You have already completed mountaintop and airplane emulations of the Geostar satellite functions. What's the timetable for the actual satellite?

O'Neill: So far, the company has met all of its milestones. We are looking to begin service to the entire continental United States in 1987.

One of the most critical items for that is the issuance by the Federal Communications Commission of what's called a "notice of proposed rulemaking," which would allocate the spectra for the Geostar service. And that is going very well. There's a very strong possibility that something important will have happened in that area even before your magazine comes out.

The development of the transceivers actually takes just about as long as the development time for the satellites themselves. It's a different kind of technical task, but the time scales are about the same.

C!: What types of services will Geostar provide?

O'Neill: In aviation, the kinds of services that would be provided would be, for example, positioning, very accurately—on the order of meters. We can technically provide what's called radio location, which means feeding back the location of a vehicle or an aircraft to a fleet dispatch headquarters. We can provide for aircraft terrain avoidance, because we will have the stored terrain map. So if we see an aircraft heading toward a TV tower or a mountain, we will be feeding warnings to the pilot at the time.

There would be, of course, a two-way digital message service, all provided through the same device. And you could send a message from any transceiver to any other transceiver with a typical delay of about six-tenths of a second. And lastly, it is also an emergency warning system, because the ground station computer will be tracking aircraft. And if you see an aircraft which is heading toward a collision with terrain, first of all, you'll be sending warnings, automatically generated by the computer, and if the aircraft does crash, you will recognize the fact from several

confirming sources. And that's important, because the so-called emergency locating transmitters (ELT) that are now federally mandated and carried by aircraft have a horrendous false alarm rate—approximately 98 percent of all ELT firings are false alarms.

C!: How does Geostar fit in among the six high-tech fields you discuss in *The Technology Edge*?

O'Neill: Well, the six technology areas that I identified as being, in my judgment, places where there is the biggest opportunity for major new markets up in the tens or hundreds of billions of dollars a year—things that would really make a difference on a worldwide scale of competition—really divide themselves into two halves. The first three are things that people feel they know all about, although they really don't as I tried to point out in the book. There are a lot of things that people didn't realize. The microengineering—which covers all of computer electronics and so on—the robotic area, and what I call genetic hardware. In the first two, the battle has already been joined on a very large scale.

The last three of those six areas are particularly interesting to me because they are still up for grabs. The first one is magnetic flight—very high-speed transport in a vacuum underground using principles of physics which are in fact more than a hundred years old.

The fifth area, the possibility that family aircraft, light aircraft, might be a new growth market, in its turn as big as automobiles were 60 years ago, is one that is the first place where I would see Geostar playing a role (in one of those six areas).

As we become a more and more dispersed society—new industries being built not in the traditional city centers but often small towns and more and more people moving to settle in suburbs and small towns, as is happening—you end up more and more in a situation where traditional transportation systems—which basically go from city center to city center—are just not very effective. If you want to go from New York to San Francisco, great. The airlines are perfectly set up to do an excellent job of that. But if you want to go from some small-town area to another small-town area, which is more and more the case these days with business travel, you don't get served very well. So the market is there.

C!: How would Geostar have an impact on aviation?

O'Neill: The way that Geostar would affect aviation is sort of generically the same way that it would affect a number of other situations in life and affairs. The difference is that in aviation, all the needs come together in one place. The fundamental thing is that the Geostar transceiver is a very light, simple, inexpensive thing, which in effect can run on double-A cells. It's a goal which the manufacturers regard as not at all impossible.

C!: How can the U.S. best maintain its lead in the area of computer development?

O'Neill: Computer development, of course, falls into the first of those areas. It's one where the battle is already joined and nobody has any very big lead. So, the opportunities for getting way out in front are not as good as they are in those last three areas.

In general, for all of the long-term big payoff developments that I was talking about in *The Technology Edge*, I think that the most important single change is a relatively minor one in the law, but it's an important one. And that would be a change that would favor funneling even a small amount of money into long-term investment. Everything in our economic system and our legal structure is set up right now to favor relatively short-term investment. The venture capitalists will tell you that they're in for the long term, but from their point of view three years is a long time.

C!: Right. Whereas the Japanese . . . .

O'Neill: The Japanese think in the decade or multidecade time scale. Now there are a number of structural reasons about the Japanese economic and political structure why that is possible. But rather than trying to imitate that, I think that it makes more sense to do something that we already know works in American society, and that is simply to alter the tax laws a little bit. And the alteration I would make is simply: Set it up so that if someone makes an investment in a company . . . and leaves his money in for a full, say ten years, then all of the earnings and appreciation—not just the capital appreciation associated with that, but the earnings from all of it—ought to be essentially untaxed for a long period of time. There ought to be a tax moratorium extending for at least several years on those returns. What that would do is just to divert a small amount of the roughly four billion dollars in venture capital funding, that now is generated, into long-term investments of that kind. It wouldn't have to be a whole lot. You know, the difference between one percent and none is already important. ©

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

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
```

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).

We regret that we are no longer able to respond to individual inquiries about programs, products, or services appearing in COMPUTE! *due to increasing publication activity. On those infrequent occasions when a published program contains a typo, the correction will appear on the CAPUTE! page, usually within eight weeks. If you have specific questions about items or programs which you've seen in* COMPUTE!, *please send them to Readers' Feedback, P.O. Box 5406, Greensboro, NC 27403.* ©

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: ▒▒▒▒▒▒▒▒▒▒▒▒ Enter these characters with the Atari logo key, {▲}.

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:
{CLR}	SHIFT + CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT + CRSR ↑	
{DOWN}	CRSR ↑	
{LEFT}	SHIFT + CRSR ←	
{RIGHT}	CRSR ←	
{RVS}	CTRL 9	
{OFF}	CTRL 0	
{BLK}	CTRL 1	
{WHT}	CTRL 2	
{RED}	CTRL 3	
{CYN}	CTRL 4	
{PUR}	CTRL 5	

When You Read:	Press:	See:
{GRN}	CTRL 6	
{BLU}	CTRL 7	
{YEL}	CTRL 8	
{F1}	f1	
{F2}	f2	
{F3}	f3	
{F4}	f4	
{F5}	f5	
{F6}	f6	
{F7}	f7	
{F8}	f8	
←	←	
↑	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.

DEVASTATOR

David R. Arnold

You and your comrades approach the hostile Devastator*—a powerful mothership ready to destroy Earth. Out of nowhere, guardian ships attack. You have 30 seconds to destroy all of them—or else Earth is destroyed. Written for the unexpanded VIC, versions are also included for the 64, Color Computer, TI-99/4A, Apple II, and IBM PC and PCjr. Joystick required for all versions except VIC (optional).*

"Devastator" is an action game where you must save Earth from aliens. What makes it different from similar games is that when you fail, Old Terra Firma is destroyed before your eyes.

You and your comrades are in one-man spaceships skimming the surface of a huge alien craft known as *Devastator*. Suddenly, out of nowhere, guardian ships appear, darting and dodging swiftly, causing havoc among your ranks. Blast them by lining up your cross hairs with the center of the spaceships and pressing the fire button. You have a mere 30 seconds to destroy ten ships before *Devastator* annihilates Earth with a death bolt.

The VIC Programs

This program is written in two parts because of the limited memory in an unexpanded VIC-20. Program 1 gives the instructions and customizes the characters. Be sure to save Program 1 before you run it. However, if you wish to view Program 1 before saving it, temporarily add the line 295 END. After you type in Program 2, save it with the name D. (For tape, be sure to save it immediately following Program 1.) Lines 305 and 310 of Program 1 will then cause Program 2 to load and run automatically.

The second program is the actual game. If you hit RUN/STOP and RESTORE anytime during the second program, you must type POKE 36869,255—no line number is needed—to play the game again. This is the location of the customized characters.

Devastator is played with a joystick simply for ease of use. However, if you want to use the keyboard, you can substitute the following lines in Program 2:

```
1000 IFPEEK(197)=17THENR=R-22
1005 IFPEEK(197)=33THENR=R+22
1010 IFPEEK(197)=28THENR=R-1
1015 IFPEEK(197)=36THENR=R+1
1110 POKEL+R,219:IFPEEK(197)<>32THEN1128
```

Delete lines 1016–1022.

The difficulty level of this game can be changed by subtracting or adding time in line 140, or by increasing or decreasing the number of points for ships hit (SC) in line 2000. (Each ship is worth ten points.) You can also make the ships harder to hit by changing the 9 in line 500 to a higher number.

Here is an explanation of Program 2:

Line	
0	Variables.
20	Print Earth and stars.
70	Print first screen of Devastator.
160	Print second screen of Devastator.
250	Print third screen of Devastator.
350	Print fourth screen of Devastator.
500	Subroutine to print ships.
800	Subroutines for sound, joystick, and cross hairs.
1120	PEEK hit of a guardian ship.
1800	Subroutines for printing saucers.
2000	Decide win or loss.
2005	Routine for loss.
2040	"Play again" option.
3000	Routine for win.

Both of these programs use a lot of memory, so don't add extra spaces.

Looks like a Ferrari. Drives like a Rolls. Parks like a Beetle.

Ask your computer dealer to take the cover off a world-class disk drive.

The all new, 1984 Indus GT.™

The most advanced, most handsome disk drive in the world.

A flick of its power switch can turn an Atari into a Ferrari.

Or an Apple into a Red Hot Apple.

Looks like a Ferrari.

The Indus GT is only 2.65" high. But under its front-loading front end is slimline engineering with a distinctive European-Gran flair.

Touch its LED-lit CommandPost™ function control AccuTouch™ buttons. Marvel at how responsive it makes every Atari or Apple home computer.

Drives like a Rolls.

Nestled into its soundproofed chassis is the quietest and most powerful disk drive power system money can buy. At top speed, it's virtually unhearable. Whisper quiet.

Flat out, the GT will drive your Atari track-to-track 0-39 in less than one second. Increasing data transfer 400%. (Faster than any other drive. And as fast as any Apple disk drive.)

And each GT comes with the exclusive *GT DrivingSystem*™ of software programs.* World-class word processing is a breeze with the *GT Estate WordProcessor.*™ And your dealer will describe the two additional programs that allow GT owners to accelerate their computer driving skills. *Included as standard equipment.

Also, the 1984 Indus GT is covered with the *GT PortaCase.*™ A stylish case that conveniently doubles as a 80-disk storage file.

Parks like a Beetle.

The GT's small, sleek, condensed size makes it easy to park.

And its low price makes it easy to buy.

$449 for Atari. $329 for Apple.

So see and test drive the incredible new 1984 Indus GT at your nearest computer dealer soon.

The drive will be well worth it.

INDUS™

The all-new 1984 Indus GT Disk Drive.

The most advanced, most handsome disk drive in the world.

For dealer information, call 1-800-33-INDUS. In California, 1-800-54-INDUS, 213/882-9600.

It all adds up...

HOME COMPUTERS

ATARI®

600XL $149
800XL $229
WHILE SUPPLY LASTS

1200XL ..... CALL
1450XL ..... CALL

850 Interface	$159.00
1010 Recorder	$71.99
1020 Color Printer	$219.00
1025 Dot Matrix Printer	$299.00
1027 Letter Quality Printer	$269.00
1030 Direct Connect Modem	$99.99
1050 Disk Drive	$349.00
1064 Memory Module	$125.00
Touch Tablet/Software	$64.99
Light Pen/Software	$72.99
CX22 Track Ball	$39.99
7097 Atari Logo	$74.99
4018 Pilot (Home)	$57.99
405 Pilot (Educ.)	$99.99
8036 Atari Pilot	$77.99
5049 VisiCalc	$149.99
488 Communicator II	$119.99
CX30 Paddles	$11.99
CX40 Joystick	$7.99
4011 Star Raiders	$31.99
4022 Pac Man	$31.99
4025 Defender	$31.99
8026 Dig Dug	$31.99
8031 Donkey Kong	$35.99
8034 Pole Position	$37.99
8040 Donkey Kong Jr.	$33.99
8043 Ms Pacman	$37.99
8044 Joust	$37.99
8045 Pengo	$33.99
8052 Moon Patrol	$33.99
4003 Assembler	$44.99
8126 Microsoft Basic I or II	$64.99

DISK DRIVES FOR ATARI

PERCOM

AT 88-S1	$279.00
AT 88-S1 PD	$299.00

INDUS

GT Drive	$379.00

RANA

1000	$299.00

TRAK

AT-D2	$389.00
AT-D4	$589.00

MEMORY BOARDS

ATARI

Axlon 32K	$59.99
Axlon 48K	$99.99
Axlon 128K	$299.00

APPLE/FRANKLIN

Axlon 128K	$299.00
Axlon 320K	$849.00

ALIEN VOICE BOX

Atari	$119.00	Apple	$149.00

DISKETTES

MAXELL

5¼" MD-1	$24.99
5¼" MD-2	$34.99
8" FD-1	$39.99
8" FD-2	$49.99

VERBATIM

5¼" SS/DD Value Life	$22.99
5¼" DS/DD Value Life	$29.99

BIB

5¼" Disk Head Cleaner	$14.99

ELEPHANT

5¼" SS/SD	$18.49
5¼" SS/DD	$21.99
5¼" DS/DD	$26.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-n-File 10	$3.99
Flip-n-File 50	$17.99
Flip-n-File 50 w/lock	$24.99
Flip-n-File (400/800 ROM)	$17.99

CONTROLLERS & JOYSTICKS

WICO

Joystick	$21.99
3-way Joystick	$22.99
Power Grip	$21.99
BOSS Joystick	$17.99
ATARI/VIC Trak Ball	$34.99
Apple Trak Ball	$64.99
Apple Analog	$37.99

KRAFT

Joystick	$41.99
Atari Single Fire	$12.99
Atari Switch Hitter	$15.99
Apple Paddles	$34.99
IBM Paddles	$34.99
IBM Joystick	$46.99

Koala

Atari (ROM)	$79.99	IBM	$99.99
C-64 (ROM)	$79.99	Apple/Franklin	$85.99

PRINTERS

AXIOM

AT-100 Atari Interface Printer	$239.00
GP-100 Parallel Interface	$199.00
GP-550 Atari Bidirectional	$319.00
GP-700 Atari Color Printer	$489.00
GP-550 Parallel Printer	$269.00

BMC

401 Letter Quality	$589.00
BX-80 Dot Matrix	$269.00

C.ITOH

Gorilla Banana	$209.00
Prowriter 8510P	$379.00
Prowriter 1550P	$599.00
A10 (18 cps)	$569.00
Hot Dot Matrix	CALL
F10-40	$999.00
F10-55	$1349.00

COMREX

ComWriter II Letter Quality	$499.00

DIABLO

620 Letter Quality	$949.00
630 Letter Quality	$1749.00

DAISYWRITER

2000	$999.00
Tractor Feed	$109.00

EPSON

RX-80, RX-80FT, RX-100	CALL
FX-80, FX-100	CALL
LQ 1500	CALL

IDS

Prism 80...For Configuration	CALL
Prism 32...For Configuration	CALL

JUKI

6100	$499.00

MANNESMAN TALLY

160L	$589.00
180L	$799.00
Spirit 80	$309.00

NEC

8023 Dot Matrix	$389.00
8025 Dot Matrix	$669.00
2010/15/30	$749.00
3510/15/30	$1369.00
7710/15/30	$1799.00

OKIDATA

82, 83, 84, 92, 93, 2350, 2410	CALL

OLYMPIA

Compact 2	$479.00
Compact RO	$509.00
ESW 3000	$1449.00

SMITH CORONA

TP-1000	$449.00
Tractor Feed	$119.00

SILVER REED

500 Letter Quality	$449.00
550 Letter Quality	$549.00
770 Letter Quality	$899.00

STAR

Gemini 10X	$299.00
Gemini 15X	$399.00
Serial Board	$75.00
Radix 10	$599.00
Radix 15	$699.00

TOSHIBA

1340	$869.00
1351	$1699.00

TRANSTAR

120P	$469.00
130P	$649.00
315 Color	$459.00

MODEMS

ANCHOR

Volksmodem	$59.99
Mark IL Serial	$79.99
Mark VII (Auto Ans/Auto Dial)	$99.99
Mark XII (1200 Baud)	$299.99
Mark TRS-80	$99.99
9 Volt Power Supply	$9.99

HAYES

Smartmodem 300	$209.00
Smartmodem 1200	$499.00
Smartmodem 1200B	$449.00
Micromodem IIe	$269.00
Micromodem 100	$299.00
Smart Com II	$89.99
Chronograph	$199.00

NOVATION

J-Cat	$99.99
Cat	$139.00
Smart Cat 103	$179.00
Smart Cat 103/212	$399.00
AutoCat	$219.00
212 AutoCat	$549.00
Apple Cat II	$249.00
212 Apple Cat	$449.00
Apple Cat 212 Upgrade	$259.00
PC Cat Access 1 2 3	$399.00

ZENITH

ZT1	$339.00
ZT-10	$309.00
ZT-11	$369.00

MONITORS

AMDEK

300 Green	$149.00
300 Amber	$159.00
310 Amber	$169.00
Color 1	$279.00
Color 1 Plus	$299.00
Color 2 Plus	$419.00
Color 3	$349.00
Color 4T IBM	$699.00

BMC

1201 (12" Green)	$88.99
1201 Plus (12" Green Hi-Res)	$98.99
9191 Plus	$249.00

GORILLA

12" Green	$88.99
12" Amber	$95.99

NEC

JB 1260 Green	$109.00
JB 1201 Green	$149.99
JB 1205 Amber	$159.99
JB 1215 Color	$259.00
JC 1216 RGB	$429.00
JC 1460 Color	$359.00

PRINCETON GRAPHICS

MAX-12 Amber	$199.00
HX-12 RGB	$539.00
SRT-2 RGB	$649.00

SAKATA

SC-100 Color	$269.00
SG-1000 Green	$129.00
SA-1000 Amber	$139.00

TAXAN

210 Color RGB	$299.00
400 Med-Res RGB	$319.00
415 Hi-Res RGB	$439.00
420 Hi-Res RGB (IBM)	$489.00
100 12" Green	$125.00
105 12" Amber	$135.00

USI

Pi 1, 9" Green	$99.99
Pi 2, 12" Green	$119.99
Pi 3, 12" Amber	$129.99
Pi 4, 9" Amber	$119.99
1400 Color	$269.99

QUADRAM

Quadchrome 8400 Color	$519.00

ZENITH

ZVM122 Amber	$99.99
ZVM123 Green	$89.99
ZVM124 IBM-Amber	$149.00
ZVM131 Color	$309.00
ZVM133 RGB	$429.00
ZVM135 RGB/Composite	$469.00

COMPUTER MAIL ORDER

west
800-648-3311
In NV call (702)588-5654,Dept. 105
Order Status Number: 588-5654
P.O.Box 6689
Stateline, NV 89449

canada
Ontario/Quebec 800-268-3974
Other Provinces800-268-4559
In Toronto call
(416)828-0866,Dept. 105
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 3B
Mississauga, Ontario, Canada L5L1T1

east
800-233-8950
In PA call (717)327-9575,Dept. 105
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. 3rd St., Williamsport, PA 17701

No risk, no deposit on C.O.D. orders and no waiting period for certified checks or money orders. Add 3% (minimum $5) shipping and handling on all orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability and price change. Call today for our catalog.

the best prices

APPLE/FRANKLIN DISK DRIVES

MICRO-SCI

A2	$219.00
A40	$299.00
A70	$319.00
C2 Controller	$79.99
C47 Controller	$89.99

RANA

Elite 1	$279.00
Elite 2	$389.00
Elite 3	$569.00

APPLE IIe STARTER PACK
64K Apple IIe, Disk Drive & Controller, 80 Column Card, Monitor II & DOS 3.3
COMPLETE.....CALL
Call on all other Apple Models

FRANKLIN

ACE 1000 Color Computer	CALL
ACE PRO PLUS System	CALL
ACE 1200 Office Mgmt. System	CALL
ACE PORTABLES	CALL

MBC 550.....CALL
MBC 555.....CALL
MBC 555-2.....CALL

MBC 1100	$1499.00
MBC 1150	$1899.00
MBC 1200	$1849.00
MBC 1250	$2099.00
PR 5500 Printer	$599.00

SANYO

HEWLETT PACKARD

HP 71B	**$419.99**
41CV	**$189.99**
41CX	**$249.99**

HP 11C	$62.99
HP 12C	$92.99
HP 15C	$92.99
HP 16C	$92.99
HP 75D	$879.99
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

SHARP

PC-1500A	**$165.99**
PC-1250A	**$88.99**
CE-125 PrinterCassette	$128.99
CE-150 Color PrinterCassette	$171.99
CE-155 8K RAM	$93.99
CE-161 16K RAM	$134.99

NEC

NEC

PC-8221A Thermal Printer	$149.99
PC-8281A Data Recorder	$99.99
PC-8201-06 8K RAM Chips	$105.00
PC-8206A 32K RAM Cartridge	$329.00

commodore

CBM 8032	**$599**
CBM 8096	$869.00
CBM 9000	$999.00
B128-80	$769.00
8032 to 9000 Upgrade	$499.00
2031 LP Disk Drive	$299.00
8050 Disk Drive	$949.00
8250 Disk Drive	$1199.00
4023 Printer	$379.00
8023 Printer	$569.00
6400 Printer	$1399.00
Z-RAM	$499.00
Silicon Office	$699.00
The Manager	$199.00
SoftROM	$125.00
VisiCalc	$159.00

PROFESSIONAL SOFTWARE

Word Pro 2 Plus	$159.00
Word Pro 3 Plus	$189.00
Word Pro 4 Plus/5 Pluseach	$279.00
InfoPro	$179.00
Administrator	$399.00
Power	$79.99

SX-64Portable	**$839**
CBM 64	**$199**
C1541 Disk Drive	$249.00
C1530 Datasette	$69.99
C1520 Color PrinterPlotter	$129.00
M-801 Dot Matrix Printer	$219.00
C1526 Dot Matrix/Serial	$299.00
C1702 Color Monitor	$259.00
C1311 Joystick	$4.99
C1312 Paddles	$11.99
C1600 VIC Modem	$59.99
C1650 Auto Mdoem	$89.99
Logo 64	$49.99
Pilot 64	$39.99
Word Pro 64 Plus	$59.99
Calc Result 64	$65.99
Calc Result Easy	$39.99
Codewriter 64	$75.99
MCS 801 Color Printer	$499.00
DPS 1101 Daisy Printer	$459.00
Magic Voice Speech Module	$54.99
Desk Organizer Lock	$49.99
Vidtex Telecommunications	$34.95

MSD

SD1 Disk Drive	$349.00
SD2 Disk Drive	$599.00

NEC PRINTERS

NEC 2050	$899.00
NEC 3550	$1669.00

PERCOM/TANDON DISK DRIVES

5¼" 320K Floppy	$219.00
5 Meg Hard w/Controller	$1049.00
10 Meg Hard w/Controller	$1349.00
20 Meg Hard w/Controller	$1899.00

VISICORP

VisiCalc IV	$159.00
VisiWord +	$249.00
Visi-on Application Manager	$79.99
Visi-on Calc	$269.00
Visi-on Graph	$179.00
Visi-on Word	$249.99
Optical Mouse	$189.99

AST RESEARCH

Six Pak Plus	from...$279.00
Combo Plus II	from...$279.00
Mega Plus	from...$309.00
I/O Plus	from...$139.00

QUADRAM

Quadlink	$479.00
Quadboard	as low as...$289.00
Quad 512 Plus	as low as...$249.00
Quadcolor	as low as...$219.00
Chronograph	$89.99
Parallel Interface Board	$89.99
64K RAM Chips Kit	$59.99

IBM

MICROPRO

WordStar Professional Pack	$389.00
InfoStar	$299.00
SpellStar	$159.00
CalcStar	$99.99

MICROSTUF

Crosstalk	$105.00

MICROSOFT

MultiPlan	$159.00

ASHTON-TATE

dBASE II	$339.00
dBASE III	$449.00
Friday!	$185.00

IUS

EasyWriter II	$249.00
EasySpeller	$119.00
EasyFiler	$229.00

CONTINENTAL SOFTWARE

1st Class Mail/Form Letter	$79.99
Home Accountant Plus	$88.99

LOTUS

Symphony	$549.00
1-2-3	$339.00

PROFESSIONAL SOFTWARE

PC Plus/The Boss	$329.00

SYNAPSE

File Manager	$89.99

ZENITH | data systems

PC COMPATIBLE 16 BIT SYSTEMS

Z-150 PC Z-160 PC

Call for price and configurations

SOFTWARE

	C-64	Atari	IBM	Apple
Electronic Arts				
One on One	$29.99	$29.99	$29.99	$29.99
Music Construction	$29.99	$29.99	$29.99	$29.99
Pinball Construction	$29.99	$29.99	$29.99	$29.99
Cut & Paste	$39.99	$39.99	$39.99	$39.99
Hard Hat Mack	$27.99	$27.99	$27.99	$27.99
InfoCom				
Witness	$29.99	$29.99	$29.99	$29.99
Infidel	$29.99	$29.99	$29.99	$29.99
Deadline	$29.99	$29.99	$29.99	$29.99
Planetfall	$29.99	$29.99	$29.99	$29.99
Enchanter	$29.99	$29.99	$29.99	$29.99
Zork1,2,3 ea	$27.99	$27.99	$27.99	$27.99
Suspended	$29.99	$29.99	$29.99	$29.99
Sorcerer	$29.99	$29.99	$29.99	$29.99
AtariSoft				
Joust	$35.99	N/A	$28.99	$28.99
Moon Patrol	$35.99	N/A	$28.99	$28.99
Ms. PacMan	$35.99	N/A	$28.99	$28.99
PacMan	$35.99	N/A	$28.99	$28.99
Donkey Kong	$35.99	N/A	$28.99	$28.99
Pole Position	$35.99	N/A	$28.99	$28.99
Spinnaker				
Aerobics	$28.99	$28.99	$28.99	$28.99
Trans	$24.99	$24.99	$24.99	$24.99
Adventure Creature	$24.99	$24.99	$24.99	$24.99
Aegean Voyage	$24.99	$24.99	$24.99	$24.99
Snooper Troops 1,2 ea	$28.99	$28.99	$28.99	$28.99
Traction Fever	$22.99	$22.99	$22.99	$22.99
Alphabet Zoo	$22.99	$22.99	$22.99	$22.99
In Search of	$24.99	$24.99	$24.99	$24.99
Facemaker	$22.99	$22.99	$22.99	$22.99
Kinder Comp	$17.99	$17.99	$17.99	$17.99
Dynatech				
Code Writer	$79.99	$79.99	$175.99	$155.99
VisiCorp				
VisiCalc	$159.99	$149.99	$159.99	$159.99
VisiCalc Advanced	N/A	N/A	N/A	$269.99
pfs:				
Write	N/A	N/A	$89.99	$89.99
Graph	N/A	N/A	$89.99	$79.99
Report	N/A	N/A	$79.99	$79.99
File	N/A	N/A	$89.99	$79.99
Solutions * as low as	N/A	N/A	$16.99	$16.99

*Call on Titles

COMPUTER MAIL ORDER

west
800-648-3311
In NV call (702)588-5654,Dept. 105
Order Status Number: 588-5654
P.O.Box 6689
Stateline, NV 89449

canada
Ontario/Quebec 800-268-3974
Other Provinces800-268-4559
In Toronto call
(416)828-0866, Dept. 105
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 3B
Mississauga, Ontario, Canada L5L1T1

east
800-233-8950
In PA call (717)327-9575,Dept. 105
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. 3rd St., Williamsport, PA 17701

VISA MasterCard

CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada. INTERNATIONAL ORDERS: All orders placed with U.S. offices for delivery outside the Continental United States must be pre-paid by certified check only. Include 3% (minimum $5) shipping and handling. EDUCATIONAL DISCOUNTS: Additional discounts are available to qualified Educational Institutions. APO & FPO: Add 3% (minimum $5) shipping and handling.

Be careful—the graphics can make this a difficult program to type in. If you would like a copy (VIC version only), send a cassette tape, a self-addressed, stamped mailer, and $3 to:

David R. Arnold
620 Alger
Owosso, MI 48867

BEFORE TYPING...

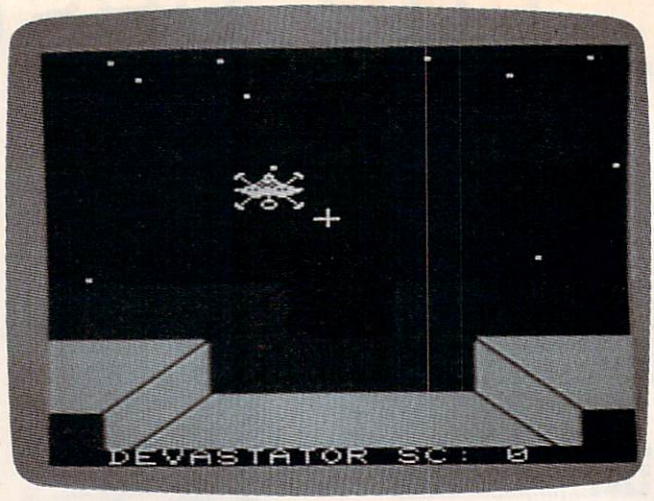
If you're new to computing, please read "How To Type COMPUTE!'s Programs" and "A Beginner's Guide To Typing In Programs."

Program 1:

Devastator – Character Creator (VIC Version)

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

```
100 PRINT"{CLR}":PRINT"{6 DOWN}{5 RIGHT}P
    LEASE WAIT"                    :rem 230
105 PRINT"{3 DOWN} DEFINING CHARACTERS"
                                   :rem 190
110 POKE52,28:POKE56,28:CLR        :rem 67
120 FORI=7168TO7679:POKEI,PEEK(I+25600):N
    EXT                            :rem 148
130 FORJ=1TO22:READC:FORI=0TO7:READD:POKE
    C+I,D:NEXT:NEXT                :rem 160
140 POKE36869,255:POKE36879,10:PRINT"
    {CLR}{9 RIGHT}{BLU}ABCD{DOWN}{4 LEFT}
    IJKL"                          :rem 30
145 PRINT"{4 DOWN}{RVS}{GRN}{6 SPACES}DEV
    ASTATOR":FORT=1TO1000:NEXT     :rem 156
150 PRINT"{HOME}"TAB(A)"{7 DOWN} {YEL}MNO
    {DOWN}{4 LEFT} UVW":A=A+1:IFA>=19THEN
    GOTO200                        :rem 235
160 POKE36878,15:POKE36874,255:POKE36877,
    128:POKE36878,0:GOTO150        :rem 194
200 PRINT"{RVS}YOUR MISSION IS TO
    {4 SPACES}PROTECT EARTH FROM THEAPPRO
    ACHING DEVASTATOR              :rem 214
210 PRINT"{RVS}SHOOT DOWN AT LEAST 10GUAR
    DIAN SHIPS TO{5 SPACES}ENABLE YOUR CO
    MRADES                         :rem 65
220 PRINT"{RVS}TO DESTROY THE{8 SPACES}DE
    VASTATOR.                      :rem 46
230 PRINT"{DOWN}{RVS}(HIT KEY TO CONTINUE
    .)"                            :rem 222
240 GETA$:IFA$=""THEN240           :rem 79
245 PRINT"{HOME}{9 DOWN}"          :rem 24
250 PRINT"{RVS}YOU HAVE ONLY 30{6 SPACES}
    SECONDS TO COMPLETE{3 SPACES}YOUR PAR
    T OF THE{6 SPACES}";           :rem 130
260 PRINT"{RVS}MISSION. POSITION YOURCROS
    SHAIR WITH THE{4 SPACES}JOYSTICK. SHO
    OT BY{4 SPACES}";              :rem 91
270 PRINT"{RVS}PRESSING THE FIRE
    {5 SPACES}BUTTON.{37 SPACES}":rem 248
280 GETA$:IFA$=""THEN280           :rem 87
290 PRINT"{CLR}{9 SPACES}{BLU}ABCD{DOWN}
    {4 LEFT}IJKL":PRINT"{DOWN}{RVS}{RED}R
    ED ALERT! DEVASTATOR APPR?OACHING!"
                                   :rem 223
300 PRINT"{RVS}GRAB YOUR JOYSTICK ANDPREP
    ARE FOR BATTLE.{3 SPACES}FOR TAPE, PR
    ESS PLAY.{BLK}";               :rem 253
305 S$="LO"+CHR$(34)+"D"+CHR$(34)+",1:"+C
    HR$(131)                       :rem 62
307 REM CHANGE 1 TO 8 IN LINE 305 IF YOU
    {SPACE}ARE USING A DISK DRIVE :rem 87
310 FORI=1TOLEN(S$):POKE630+I,ASC(MID$(S$
    ,I)):NEXT:POKE198,I:END        :rem 139
1000 DATA7176,255,255,253,255,127,127,127
     ,127                          :rem 230
1100 DATA7184,254,254,254,127,255,255,255
     ,255                          :rem 235
1200 DATA7192,0,0,199,231,239,239,195,129
                                   :rem 32
1300 DATA7200,63,63,159,207,254,254,252,2
     52                            :rem 121
1400 DATA7240,63,35,49,17,11,7,1,0
                                   :rem 172
1500 DATA7248,255,255,255,255,255,255,255
     ,63                           :rem 194
1600 DATA7256,129,1,3,131,131,199,207,220
                                   :rem 15
1700 DATA7264,252,248,248,240,240,224,128
     ,0                            :rem 121
1800 DATA7208,0,15,59,245,123,31,0,0
                                   :rem 13
1900 DATA7216,0,255,189,90,189,255,0,0
                                   :rem 137
2000 DATA7224,0,240,220,175,220,240,0,0
                                   :rem 140
2010 DATA7272,96,192,160,16,9,6,31,245
                                   :rem 134
2020 DATA7280,32,112,80,248,108,151,111,2
     50                            :rem 104
2030 DATA7288,48,24,40,64,128,0,192,248
                                   :rem 189
2040 DATA7336,63,15,7,8,16,160,192,96
                                   :rem 90
2050 DATA7344,255,255,39,32,112,136,136,1
     12                            :rem 120
2060 DATA7352,240,192,0,128,64,40,24,48
                                   :rem 176
2070 DATA7360,8,16,41,7,61,31,17,35
                                   :rem 233
2080 DATA7368,16,8,148,224,188,248,8,132
                                   :rem 253
2090 DATA7168,255,255,255,255,255,255,255
     ,255                          :rem 251
3000 DATA7296,127,191,223,239,247,251,253
     ,254                          :rem 233
3010 DATA7304,254,253,251,247,239,223,191
     ,127                          :rem 224
```



"Devastator," VIC version.

MICRO-SYS DISTRIBUTORS. THE COMMODORE CONNECTION.

commodore

SOFTWARE FOR C-64

Business

Multiplan (Spreadsheet)	$	63.00
Calc Result (Easy)	$	45.00
Calc Result (Advanced)	$	95.00
Superbase 64	$	95.00
Mirage Concepts (Data Base)	$	89.00
Mirage Concepts (Word Processor) (40/80 clm & 30K Dictionary)	$	89.00
Mirage Concepts (Report Generator)	$	49.00
Home Accountant (Continental)	$	49.00
Tax Advantage (Continental)	$	39.00
Southern Solutions Accounting G/L, A/R, A/P, P/R, I/M ..... each	$	69.95

Utilities

Disk Utility Program (Fast Copy, File Copy, Disassembler & more)	$	49.00
Smart +64 Terminal	$	49.00
TOOL 64	$	39.95
Simon's Basic	$	39.95
80 Column Expander (Cartridge)	$	60.00

ACCESSORIES

Solo Flight (Simulator)	$	34.95
Hellcat Avenger	$	34.95
Oscar by Databar (Bar Code Reader)	$	79.95
CBM 1541 (new version)	$	269.00
Concord Parallel Disk Drive	$	345.00
Concord Slave Drive		Call
MSD Super Disk (Single)	$	395.00
MSD Super Disk (Dual)	$	695.00
1600 Modem	$	63.00
Vic 1650 Automatic Modem	$	109.95
Hayes Smart 300 Modem	$	249.00
Hayes Smart 1200 Modem	$	629.00
Vic 1530 Datasette	$	65.00
Cardco Datasette	$	55.00
5 Slot Expander (64)	$	65.00
Printer Utility Program (Cardco)	$	19.95
64 Relay Cartridge	$	45.00
Numeric Key Pad	$	49.00
Alien Voice Box (Talks & Sings)	$	119.00
When I'm 64 (Voice Box Sings)	$	35.00
Voice Box Dictionary	$	35.00
Texas Instruments LCD Programmer	$	55.95
Verbatim Diskettes:		
Single Sided/Single Density	$	26.00
Single Sided/Double Density	$	30.00
Double Sided/Double Density	$	42.00
Vic 20:		
3-Slot Expander	$	39.00
6-Slot Expander	$	79.95
16K Memory	$	79.95
CBM 4023 Ribbons	$	9.95
CBM 8023 Ribbons	$	9.95
Flip N' File 10, 15, 25, 50		Call
Power Strips w/surge stopper		Call
Computer Care Kit	$	19.95
Disk Cleaning Kit	$	12.95

INTERFACES

Superbox (Transparent IEEE, 3-Slot, Reset)	$	139.95
Interpod (Intelligent IEEE & RS-232)	$	139.95
The Connection (By Tymac) (Commodore Graphics + 2K Buffer)	$	95.00
Cardco +G Parallel Interface	$	89.00
Vic Switch	$	149.95
IEEE to Centroonics	$	149.95
Pet-to-IEEE Cable	$	39.00
IEEE-to-IEEE Cable	$	49.00
4 Prong A/V Cable	$	15.00
Centronics Cable (male to male)	$	34.95
RS232 Cable (male to male)	$	31.95
Networking for C-64 & CBM Equipment		Call

MONITORS

CBM 1702 Color Monitor	$	269.00
Panasonic TR-120 (Green)	$	142.00
Panasonic TR-120 (Amber)	$	156.00
Panasonic DT-1300 Color	$	425.00
Monitor Stand (Tilt & Swivel)	$	29.95
RGB Monitor Cable:		
ET-100C (Apple)	$	33.80
ET-101C (IBM)	$	33.80
Green & Amber Monitors	$	95.00

LETTER QUALITY PRINTERS

CBM 6400 Printer		$1425.00
Abati (20 CPS)	$	475.00
Cardco LQ/1 Printer	$	495.00
NEC Spinwriter		Call

DOT MATRIX PRINTERS

CBM MPS-801 Printer (50cps)	$	245.00
CBM 4023 (100 CPS)	$	339.00
CBM 8023 (160 CPS)	$	565.00
Panasonic KX-P1090 (100 CPS)	$	285.00
Panasonic KX-P1091 (120 CPS)	$	325.00
Panasonic KX-P1092 (180 CPS)		Call
Panasonic KX-P1093		Call

COMMODORE BUSINESS MACHINES

Executive 64 Portable	$	795.00
B128-80 (128K 80 column)	$	695.00
SuperPet (5 languages)	$	1059.00
CBM 8032	$	595.00
CBM 8096	$	675.00
CBM 2031 single disk	$	295.00
CBM 8050 Dual Disk 1 meg.	$	995.00
CBM 8250 Dual Disk 2 meg.	$	1295.00
CBM 9060 Hard Disk/5 Meg.	$	2195.00
64K Expansion Board	$	275.00
SuperPet Upgrade Kit	$	695.00

BUSINESS SOFTWARE — 8032/8096

Wordpro 4 + or 5 +	$	295.00
Calc Result	$	199.95
The Manager	$	199.00
Superbase (8096 only)	$	225.00
BPI Accounting System 6 Separate Modules ..... each	$	325.00
Southern Solutions Accounting 5 Separate Modules ..... each	$	285.00
McTerm Communications Package	$	175.00

BUSINESS SOFTWARE — B128

Superscript II (40K Dictionary)	$	199.00
Superbase (Data Base)	$	199.00
Complete Accounting System (G/L,A/R,A/P,O/E,P/R,I/M)	$	199.00
Calc Result	$	199.00

TERMS

Orders under 50.00 add 10.00 Handling fee
MasterCard, VISA, Money Order, Bank Check
COD (add 5.00)
Add 3% For Credit Cards

All Products In Stock Shipped Within 24 Hours

F.O.B. Dallas, Texas
All Products Shipped With Manufacturers
90 Day Warranty

PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE.

DEALERS INQUIRIES WELCOME

Call to Order
1-800-527-1738
All Others Call
1-214-231-2645

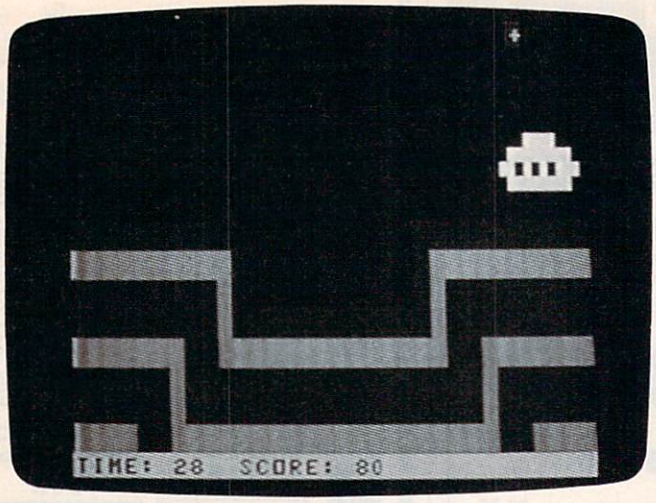
Micro-Sys
DISTRIBUTORS

641 Presidential Drive • Richardson, Texas 75081 • 9:30 a.m.-6:30 p.m. (Mon.-Fri.) • 10:30 a.m.-2:30 p.m. (Sat.)

Program 2:
Devastator – Main Program (VIC Version)

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

```
0 TI$="000000":SC=0:T$="000020"    :rem 36
10 V=36878:S4=36877:POKE36879,10:L=7888:P
   RINT"{CLR}";                    :rem 147
20 PRINT"{WHT}{2 SPACES}.{3 SPACES}.
   {2 SPACES}{BLU}ABCD{WHT} .{5 SPACES}.
   {4 SPACES}.{5 SPACES}{BLU}IJKL{WHT}
   {4 SPACES}.{11 SPACES}.{4 DOWN}";
                                   :rem 117
60 PRINT".{12 SPACES}.{12 SPACES}.
   {11 SPACES}.{4 SPACES}.{8 SPACES}.
                                   :rem 41
65 PRINT"{54 SPACES}.{2 SPACES}.
   {4 SPACES}.{4 SPACES}.{2 SPACES}"
                                   :rem 244
70 J=1:PRINT"{BLU}@@@@@@@@Q{4 SPACES}P@@@
   @@@@{PUR}@@@@@@@@Q{BLU}@{4 SPACES}@
   {PUR}P@@@@@@@@@@@@@@Q@{BLU}@{4 SPACES}@
   {PUR}@P@@@@@@";                 :rem 121
100 PRINT"@@@@@Q@@{BLU}Q@@@@P{PUR}@@P@@@@
    @@@@@Q@@Q@@@@@@P@@P@@@@{BLU}@@@Q{PUR}
    @@Q@@@@@@@@P@@{BLU}P@@@";      :rem 120
130 PRINT"@@Q@{PUR}@Q@@@@@@@@@@P@{BLU}@P@
    @@Q@@{PUR}Q@@@@@@@@@@@@@P{BLU}@@P@Q@@Q
    @@@@@@@@@@@@@@P@@P";           :rem 92
140 PRINT"{RVS}{YEL}{2 RIGHT}DEVASTATOR S
    C:"SC;:IFTI$=>"000030"THEN2000:rem 75
150 GOTO800                        :rem 103
160 J=2:PRINT"{BLU}@@@@@@@@Q{4 SPACES}P@@
    @@@@@@@@@@@@@Q@{4 SPACES}@P@@@@@@@@@@
    @@@Q@@{4 SPACES}@@P@@@@@@{PUR}@@@";
                                   :rem 244
190 PRINT"@@Q{BLU}@@Q@@@@P@@{PUR}P@@@@@@@
    @@Q@{BLU}@Q@@@@@@P@{PUR}@P@@@@@@@Q@@@
    {BLU}Q@@@@@@@@P{PUR}@@P@@@@@Q@@@Q";
                                   :rem 255
220 PRINT"@@@@@@@@@@P@@P@@{BLU}@Q{PUR}@@Q
    @@@@@@@@@@@@P@@{BLU}P@Q@{PUR}@Q@@@@@@
    @@@@@@@@P@{BLU}@P";:GOTO800   :rem 228
250 J=3:PRINT"{PUR}@@@@@@@@Q{4 SPACES}P@@
    @@@@@@{BLU}@@@@@@@Q{PUR}@{4 SPACES}@
    {BLU}P@@@@@@@@@@@@@Q@{PUR}@{4 SPACES}
    @{BLU}@P@@@@@";                :rem 107
280 PRINT"@@@@@@Q@@{PUR}Q@@@@P{BLU}@@P@@@
    @@{PUR}@@@@Q{BLU}@@Q@@@@@@P@@{PUR}P@@
    @@@@@Q@{BLU}@Q@@@@@@@@P@{PUR}@P@@@";
                                   :rem 249
320 PRINT"@@Q@@{BLU}Q@@@@@@@@@@P{PUR}@@P@
    @@Q@@Q@@@@@@@@@@@@P@@P@{BLU}Q{PUR}@@Q
    @@@@@@@@@@@@@@P@@{BLU}P";     :rem 124
330 GOTO800                        :rem 103
350 J=4:PRINT"{PUR}@@@@@@@@Q{4 SPACES}P@@
    @@@@@@@@@@@@@Q@{4 SPACES}@P@@@@@@@
    {BLU}@@@@@@Q{PUR}@@{4 SPACES}@@{BLU}P
    @@@@@@@";                      :rem 50
380 PRINT"@@@@Q@{PUR}@Q@@@@P@{BLU}@P@@@@@
    @@@@Q@@{PUR}Q@@@@@@P{BLU}@@P@@@@@@@Q@
    @Q@@@@@@@@P@@P@@@{PUR}@@Q{BLU}@";
                                   :rem 239
420 PRINT"@Q@@@@@@@@@@P@@{PUR}P@@@Q@{BLU}
    @Q@@@@@@@@@@@@P@{PUR}@P@Q@@{BLU}Q@@@@
    @@@@@@@@@@P{PUR}@@P";:GOTO800:rem 244
500 F=INT(RND(.)*9):IFTI$>=T$THEN1800
                                   :rem 143
510 PRINTTAB(F)"{6 UP}{4 SPACES}MNO
    {19 SPACES}UVW{34 SPACES}";    :rem 98
515 PRINT"{36 SPACES}":RETURN      :rem 134
520 Y=INT(RND(.)*3):PRINTTAB(Y+F)"{6 UP}
    {22 SPACES}";:IFTI$>=T$THEN1900
                                   :rem 61
530 PRINT"{WHT}{8 SPACES}XY{64 SPACES}";
                                   :rem 90
535 PRINT"{3 SPACES}":RETURN       :rem 136
800 POKEV,5:POKES4,140:POKEV,0:PRINT"
    {11 UP}"                       :rem 144
1000 POKE37154,127:A=(PEEK(37137)AND28)OR
     (PEEK(37152)AND128):A=ABS((A-100)/4)
     -7                            :rem 73
1005 POKE37154,255:IFA=7THEN1080   :rem 55
1010 ONAGOTO1015,1016,1017,,1018,1019,,,,
     ,1020,1021,1022               :rem 34
1015 R=R+21:GOTO1080               :rem 114
1016 R=R-23:GOTO1080               :rem 119
1017 R=R-1:GOTO1080                :rem 68
1018 R=R+22:GOTO1080               :rem 118
1019 R=R-22:GOTO1080               :rem 121
1020 R=R+1:GOTO1080                :rem 60
1021 R=R-21:GOTO1080               :rem 113
1022 R=R+23:GOTO1080               :rem 114
1080 IFR<-44THENR=-44              :rem 206
1085 IFR>44THENR=44                :rem 123
1100 ONJGOSUB500,520,500,520       :rem 5
1110 POKEL+R,219:B=PEEK(37137)AND32:IFBTH
     ENONJGOTO160,250,350,70       :rem 250
1120 IFPEEK((L+R)-1)=13ORPEEK((L+R)-1)=21
     ORPEEK((L+R)-1)=24THEN1130    :rem 55
1125 IFPEEK((L+R)+1)=25ORPEEK((L+R)-1)=5T
     HEN1130                       :rem 221
1128 ONJGOTO160,250,350,70         :rem 157
1130 POKES4,220:FORS=8TO255STEP5:POKE3687
     9,S:POKEV,4:POKEV,0:NEXT:SC=SC+10:GO
     TO10                          :rem 131
1800 PRINTTAB(F)"{4 UP}{YEL}{4 SPACES}EFG
     {48 SPACES}":RETURN           :rem 214
1900 PRINTTAB(Y+F)"{YEL}{3 SPACES}EFG
     {51 SPACES}";                 :rem 56
1905 PRINT"{20 SPACES}":RETURN     :rem 186
2000 IFSC>=100THENPRINT"{HOME}{4 DOWN}
     {7 SPACES}{RVS}YOU WIN!":POKEV,9:FOR
     T=1TO500:NEXT:GOTO3000        :rem 86
2005 PRINT"{RVS}{10 UP}{3 SPACES}N{UP}N
     {UP}N{UP}{LEFT}M{UP}{LEFT}N{UP}N{UP}
     N{UP}N{LEFT}{UP}M{LEFT}{UP}N{UP}N"
                                   :rem 74
2010 PRINT"{HOME}{RVS}{4 SPACES}[C]
     [F] [6 B] [D] [C]{2 SPACES}
     [C]{2 SPACES}[F]{2 SPACES}[D]
     {SPACE}[F] [4 B]{2 SPACES}[F]
     {SPACE}[D]{3 SPACES}[C]";:rem 74
2020 PRINT"{7 SPACES}[D]{SHIFT-SPACE}
     [D]{SHIFT-SPACE}[C] ":FORT=15TO0
     STEP-1:POKEV,T:FORL=1TO100:NEXTL:NEX
     TT                            :rem 78
2030 PRINT"{RVS}{RED} YOU HAVE FAILED IN
     {7 SPACES}YOUR MISSION!{6 SPACES}EAR
     TH IS DESTROYED!{2 SPACES}"  :rem 151
2040 POKE198,0:PRINT"{RVS}HIT F1
     {2 SPACES}TO PLAY AGAIN."     :rem 211
2050 GETP$:IFP$=""THEN2050         :rem 207
2060 IFP$="{F1}"THEN0              :rem 48
2070 IFP$<>"{F1}"THEN2050          :rem 5
3000 FORT=8164TO7856STEP-1:M=INT(RND(.)*4
     ):B=INT(RND(1)*2):POKEV,M:POKET,251+
     B:NEXT                        :rem 60
3010 PRINT"{RVS}DEVASTATOR{2 SPACES}DESTR
     OYED!":POKEV,0:GOTO2040       :rem 204
```



Another alien saucer waits to be destroyed in "Devastator," Color Computer version.

Program 3:
Devastator – Color Computer Version
by Charles Brannon, Program Editor

```
100 REM COLOR COMPUTER DEVASTATOR
105 CLEAR 1000:Z=1:PS=RND(159):PP=1
    024+111
107 B$=CHR$(128):B$=B$+B$+B$+B$+B$+B$+
    B$+B$
110 DIM FRAME$(2,7)
120 FOR I=0 TO 2
125 RESTORE
130 FOR J=0 TO 7
140 READ A$
150 FOR K=1 TO LEN(A$)
160 C$=MID$(A$,K,1)
165 IF C$=" " THEN C$=CHR$(128):GOT
    0180
170 IF VAL(C$)=I+1 THEN C$=CHR$(239
    ) ELSE C$=CHR$(175)
180 MID$(A$,K,1)=C$:NEXT
185 CLS J:PRINT@237,23-(I*8+J);
190 FRAME$(I,J)=A$:NEXT:NEXT
200 CLS 0:PRINT@480
210 PRINT@224,"";:FORJ=0TO7:PRINT F
    RAME$(FR,J);:NEXT:FR=FR+1:IFFR=
    3THENFR=0
215 TM=TM+1:PRINT@480,"TIME:";40-TM
    ;:IF TM=40 THEN 500
220 IF LL=0 THEN LL=RND(10):GOSUB30
    00:DX=R:GOSUB3000:DY=R:GOSUB300
    0:DZ=R
230 PRINT@PS,B$;:PRINT@PS+32,B$;
240 PS=PS+DX+32*DY:IFPS<0ORPS>159TH
    ENPS=PS-DX-32*DY
250 GOSUB 3000:Z=Z+R:IF Z<1 OR Z>6
    THEN Z=Z-R
255 GOSUB 2000
260 LL=LL-1
270 POKE PP,128:QX=JOYSTK(0):QY=JOY
    STK(1)
275 TP=PP+(QX<20)-(QX>44)+32*(QY<20
    )-32*(QY>40)
276 IF TP>1024 AND TP<1215 THEN PP=
    TP
277 P=PEEK(PP):IFP=128THENPOKEPP,43
    :GOTO210
280 Y=INT(PS/32):X=PS-Y*32:X=X*2:Y=
    Y*2
290 FORI=1TO200STEP10:PRINT@PS+RND(
    7)-1+(RND(2)-1)*32,CHR$(RND(128
    )+127);:SOUND I,1:NEXT
295 PRINT@PS,B$;:PRINT@PS+32,B$;:PS
    =RND(191)
300 SOUND 255,2:SOUND250,2:SOUND100
    ,2:SOUND 255,2
305 PTS=PTS+(7-Z)*10:PRINT@490,"SCO
    RE:";PTS;:TM=0:GOTO210
499 REM EXPLOSION OF EARTH
500 WPOS=202:CLS0
510 PRINT@WP+1,CHR$(193)CHR$(195)CH
    R$(195)CHR$(195)CHR$(194);
520 PRINT@WP+32,CHR$(161)CHR$(175)C
    HR$(175)CHR$(143)CHR$(175)CHR$(
    175)CHR$(162);
530 PRINT@WP+64,CHR$(143)CHR$(143)C
    HR$(175)CHR$(175)CHR$(175)CHR$(
    175)CHR$(175);
540 PRINT@WP+96,CHR$(196)CHR$(175)C
    HR$(175)CHR$(175)CHR$(175)CHR$(
    175)CHR$(168);
550 PRINT@WP+129,CHR$(196)CHR$(204)
    CHR$(204)CHR$(204)CHR$(204);
560 PRINT@WP+161,"EARTH";
570 FORZ=1TO6:SOUND100,2:PS=7-Z:PRI
    NT@PS,B$;:PRINT@PS+32,B$;:PS=6-
    Z:SOUND200,2:GOSUB2000:NEXTZ
575 FOR J=1 TO 2
```

Color Computer Notes

Use a joystick plugged into the right port to play the Color Computer version of "Devastator" (Program 3). Type the DATA statements carefully; they determine the shape and color of the moving trench. The program reads the patterns of 1's, 2's, and 3's and creates three different "views" of the trench, using the solid-colored blocks in the character set. When these are shown in succession, you get the illusion of moving bands. This is all made feasible, of course, by the Color Computer's very fast PRINTing speed.

The alien ship is drawn by several subroutines; each draws a different-sized ship. By erasing and redrawing, the alien ship can be made to appear to weave in and out three-dimensionally. The alien ship is also drawn with relatively low-resolution, quarter-square characters. The main program checks for a collision between the cross hairs and the alien simply by comparing their X,Y coordinates.

You have a limited amount of time to shoot the alien. If you take too long (the clock counts down to zero), a colored beam blasts and reduces the Earth to a smear of colorful dots. There's always the next game!

NEW 128K COMMODORE - 80 COLUMN

COMPUTER SYSTEM SALE!

HOME • BUSINESS • WORD PROCESSING

List Price $4008.95

LOOK AT ALL YOU GET FOR ONLY ***$1195***

	LIST PRICE
• B128 COMMODORE 128K 80 COLUMN COMPUTER	995.00
• 8050 DUAL DISK DRIVE (over 1 million bytes)	1795.00
★ 4023 - 100 CPS - 80 COLUMN BIDIRECTIONAL PRINTER	499.00
• 12" HI RESOLUTION 80 COLUMN GREEN OR AMBER MONITOR	249.00
• SUPER SCRIPT 80 COLUMN PROFESSIONAL WORDPROCESSOR	149.50
• SUPER BASE PROFESSIONAL DATA BASE	149.50
• BOX OF 10 LORAN LIFETIME GUARANTEED DISKS	49.95
• 1100 SHEETS FANFOLD PAPER	19.95
• ALL CABLES NEEDED FOR INTERFACING	102.05
TOTAL LIST PRICE	**4008.95**

SALE PRICE $1195.00

Printer replacement options (replace the 4023 with the following at these sale prices)

	LIST	SALE
★ Olympia Executive Letter Quality Serial Printer/Typewriter	849.00	**499.00**
★ Comstar Hi-Speed 160 CPS 15½" Serial-Business Printer	779.00	**449.00**

Plus You Can Order These Business Programs At Sale Prices

	LIST	SALE		LIST	SALE
ACCOUNTS RECEIVABLE	149.95	**99.00**	INVENTORY	149.95	**99.00**
ACCOUNTS PAYABLE	149.95	**99.00**	GENERAL LEDGER	149.95	**99.00**
PAYROLL	149.95	**99.00**			

15 DAY FREE TRIAL We give you 15 days to try out this SUPER SYSTEM PACKAGE!! If it doesn't meet your expectations, just send it back to use prepaid and we will refund your purchase price!!

90 DAY IMMEDIATE REPLACEMENT WARRANTY If any of the SUPER SYSTEM PACKAGE equipment or programs fail due to faulty workmanship or material we will replace it IMMEDIATELY at no charge!!

Add $50.00 for shipping and handling!!
$100.00 for Canada, Puerto Rico, Hawaii orders.
WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D. to U.S. addresses only.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

COMMODORE 64
(more power than Apple II at half the price)

$99.50*

- 170K DISK DRIVE $159.00*
- TRACTION FRICTION PRINTER $79.00*

COMPUTER AND SOFTWARE SALE

WE HAVE THE BEST SERVICE

WE HAVE THE LOWEST PRICES

VIC-20
(a real computer at the price of a toy)

$79.50

- COM-64 POWER FOR VIC-20 $79.00
- NEW VOICE SYNTHESIZER $59.00 (Com-64 or VIC-20)

* COMMODORE 64 COMPUTER $99.50
You pay only $199.50 when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE COUPON we pack with your computer that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied, your net computer cost is $99.50!!

*170K DISK DRIVE $159.00
You pay only $259.00 when you order the 170K Disk Drive! LESS the value of the SPECIAL SOFTWARE COUPON we pack with your disk drive that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied, your net disk drive cost is $159.00.

*TRACTION FRICTION PRINTER $79.00
You pay only $179.00 when you order the Comstar T/F deluxe line printer that prints 8 1/2 x 11 full size, single sheet, roll or fan fold paper, labels etc. 40, 66, 80, 132 columns. Impact dot matrix, bi-directional, 80 CPS. LESS the value of the SPECIAL SOFTWARE COUPON we pack with your printer that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied your net printer cost is only $79.00.

4 COLOR PRINTER/PLOTTER $99.00
Lowest cost, 4 color, 80 column, letter quality PRINTER/PLOTTER for Com-64 or VIC-20 computers!! List programs. High resolution graphics for charts and geometric figures. INCLUDES INTERFACE AND SPECIAL SOFTWARE SAVINGS COUPON!!

80 COLUMN BOARD $99.00
Now you program 80 COLUMNS on the screen at one time! Converts your Commodore 64 to 80 COLUMNS when you plug in the 80 COLUMN EXPANSION BOARD!! PLUS—you can get an 80 COLUMN BOARD WORD PROCESSOR with mail merge, terminal emulator, ELECTRONIC SPREAD SHEET. List $59.00 **SALE $24.95** if purchased with 80 COLUMN BOARD!! (Tape or Disk)

80 COLUMNS IN COLOR EXECUTIVE WORD PROCESSOR $69.00
This EXECUTIVE WORD PROCESSOR is the finest available for the COMMODORE 64 computer! The ULTIMATE for PROFESSIONAL Word-processing application! DISPLAYS 40 OR 80 COLUMNS IN COLOR or Black and White! Simple to operate, powerful text editing with a 250 WORD DICTIONARY, complete cursor and insert/delete key controls line and paragraph insertion, automatic deletion, centering, margin settings and output to all printers! Includes a powerful mail merge. 20,000 WORD DICTIONARY - List $24.95 **SALE $19.95.** EXECUTIVE DATA BASE - List $69.00 **SALE $49.00.** (Disk

SPECIAL SOFTWARE COUPON

We pack a SPECIAL SOFTWARE COUPON with every COMMODORE 64 COMPUTER-DISK DRIVE-PRINTER-MONITOR we sell! This coupon allows you to SAVE OVER $100 OFF SALE PRICES! Up to $500 savings are possible!!

PROFESSIONAL SOFTWARE COMMODORE 64

Name	List	Sale	Coupon
Executive Word Processor	$99.00	$69.00	$59.00
Executive Data Base	$69.00	$59.00	$39.00
20,000 Word Dictionary	$24.95	$19.95	$14.95
Electronic Spreadsheet	$59.95	$49.00	$39.00
Accounting Pack	$49.00	$39.00	$29.00
Total 5.2 Word Processor			
Tape	$69.00	$49.00	$34.00
Disk	$79.95	$59.00	$39.00
Total Text 2.6 Word Processor			
Tape	$44.95	$34.95	$22.00
Disk	$49.00	$39.00	$27.00
Total Label 2.6			
Tape	$24.95	$18.00	$12.00
Disk	$29.95	$23.00	$15.00
Programmers Helper (Disk)	$59.00	$39.95	$29.95
80 Column Screen (Disk)	$59.95	$39.95	$29.95
Crush-Crumble-Chomp (Tape/Disk)	$29.95	$24.95	$19.95
Pitstop (Cartridge)	$39.95	$29.95	$24.95
Typing Teacher (Tape/Disk)	$29.95	$24.95	$15.00
Sprite Designer (Disk)	$16.95	$14.95	$10.00
Fireball Joy Stick	$24.95	$15.95	$10.00
Light Pen	$39.95	$16.95	$14.95
Dust Cover	$ 8.95	$ 6.95	$ 4.60

(See 100 coupon items in our catalog!)

Write or call for Sample SPECIAL SOFTWARE COUPON!

EXECUTIVE QUALITY PROFESSIONAL BUSINESS SOFTWARE

The Cadillac of business programs for Commodore 64 Computers

Item	List	*SALE	Coupon
Inventory Management	$99.00	$59.00	$49.00
Accounts Receivable	$99.00	$59.00	$49.00
Accounts Payable	$99.00	$59.00	$49.00
Payroll	$99.00	$59.00	$49.00
General Ledger	$99.00	$59.00	$49.00

VIC-20 COMPUTER $79.50
This 25K VIC-20 computer includes a full size 66 key typewriter keyboard color and graphics keys, upper/lower case, full screen editor, 16K level II microsoft basic, sound and music, real time floating point decimal, self teaching book, connects to any T.V. or monitor!

COM-64 POWER FOR VIC-20 $79.00
Just plug in our 32K RAM MEMORY EXPANDER and you get as much usable programming power as the Commodore-64 computer!! Master control switches on cover, Gold Edge connectors, five year warranty (FREE $29.95; CARTRIDGE GAME)

NEW VOICE SYNTHESIZER $59.00
For Com-64 or VIC-20 computers. Just plug it in and you can program words and sentences, adjust volume and pitch, make talking adventure games, sound action games and customized talkies!! FOR ONLY $19.95 you can add TEXT TO SPEECH, just type a word and hear your computer talk—ADD SOUND TO "ZORK," SCOTT ADAMS AND AARDVARK ADVENTURE GAMES!! (Disk or tape).

16K RAM CARTRIDGE $49.00
Increases VIC-20 programming power 4 times. Expands total memory to 41K (41,000 bytes). Memory block switches are an outside cover! CARDCO Includes FREE $29.95 game!!

8K RAM CARTRIDGE $34.95
Increases VIC-20 programming power 2 1/2 times. Expands total memory to 33K (33,000 bytes). Includes FREE $16.95 game.

3 SLOT SWITCHABLE EXPANDER $24.95
Allows you to add 3 cartridges at one time—switch select to turn slots on or off—PLUS reset button. A must for your VIC-20 computer!!

60K MEMORY EXPANDER $49.00
Sixslot Board — Switch selectable — Reset button — Ribbon cable — CARDCO. A must to get the most out of your VIC-20 Computer!

9" GREEN SCREEN MONITOR $69.00
Excellent quality SANYO, easy to read, 80 columns x 24 lines, Green Phosphorous screen with anti-glare, metal cabinet! Saves your T.V. PLUS $9.95 for connecting cable. Com-64 or VIC-20.

12" GREEN OR AMBER MONITOR $99.00
Your choice of green or amber screen monitor, top quality, SANYO, 80 columns x 24 lines, easy to ready, anti-glare, faster scanning! A must for word processing PLUS $9.95 for connecting cable. Com-64 or VIC-20.

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

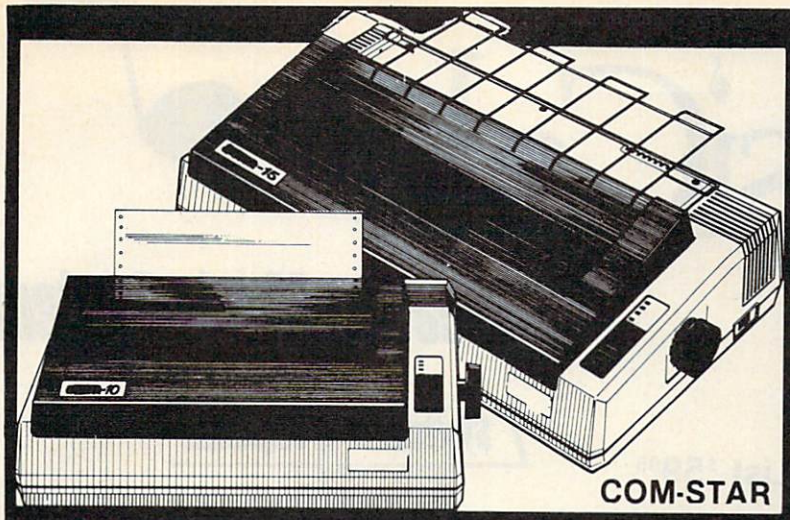
Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD — C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor
Friction
Printer

only $ 179**

15 Day Free Trial -180 Day Immediate Replacement Warranty

- Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.
- Fast 80-120-160 Characters Per Second • 40, 46, 66, 80, 96, 132 Characters Per Line Spacing
- Word Processing • Print Labels, Letters, Graphs and Tables • List Your Programs
- Print Out Data from Modem Services • "The Most Important Accessory for Your Computer"

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal paper! Best thermal printer in the U.S.A.! (Centronics Parallel Interface).

**DELUXE COMSTAR T/F 80 CPS PRINTER—$179.00

The COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interface).

Premium Quality—120 CPS COMSTAR T/F SUPER-10X PRINTER—$289.00

COMSTAR T/F (Tractor Friction) SUPER-10X PRINTER gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters, plus 2K of user definable characters! The COMSTAR T/F SUPER-10X PRINTER was Rated No. 1 by "Popular Science Magazine." It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX 80).

Premium Quality—120 CPS COMSTAR T/F SUPER-15½" PRINTER—$379.00

COMSTAR T/F SUPER 15½" PRINTER has all the features of the COMSTAR T/F SUPER-10X PRINTER plus a 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100).

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 10" PRINTER—$399.00

SUPER HIGH SPEED COMSTAR T/F (Tractor Friction) PRINTER has all the features of the COMSTAR SUPER-10X PRINTER plus SUPER HIGH SPEED PRINTING—160 CPS, 100% duty cycle, 8K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs. RED HOT BUSINESS PRINTER at an unbelievable low price!! (Serial or Centronics Parallel Interface)

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 15½" PRINTER—$529.00

SUPER HIGH SPEED COMSTAR T/F 15½" PRINTER has all the features of the SUPER HIGH SPEED COMSTAR T/F 10" PRINTER plus a 15½" carriage and more powerful electronics to handle larger ledger business forms! Exclusive bottom paper feed!!

PARALLEL INTERFACES

For VIC-20 and COM-64—$69.00
For Apple Computers—$79.00

NOTE: Other printer interfaces are available at computer stores!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

Add $17.50 for shipping, handling and insurance. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

Introducing MusiCalc™

The Creative Music System

SALE 39.95

List $59.95

Synthesizer & Sequencer 1

Turn Your Commodore-64 Into A Sophisticated Musical Instrument

"The Program That Gives You A Reason To Buy A Commodore-64."

New York Times.

With Musicalc **anyone** can • Make and record sophisticated music • Print out sheet music from your creations • Turn your computer into a **Cord Organ** • **No Experience Necessary!**

To prove it we will send you a **Free Record** with music created on a **Commodore 64 computer and Musicalc**

To get your **Free Record** call **Protecto Enterprizes**

MusiCalc™

Synthesizer & Sequencer 1

MusiCalc ScoreWriter 2

MusiCalc Keyboard Maker 3

Synthesizer & Sequencer

This 1st step turns your Commodore-64 into a **Cord Organ** — a three voice synthesizer and fully interactive step sequencer play along with pre-recorded songs or develop your own and record the music you create.

List $59.00. **Sale $39.95.**

ScoreWriter

Combine with Musicalc 1 and a graphics printer (Super-10) to produce sheet music from your original composition. List $39.95 **Sale $29.95** Coupon $24.95

Keyboard Maker

Turns your Commodore-64 into a musical keyboard. Comes with over 30 pre-set keyboard scales from Classical to Rock. Requires Musicalc 1. List $39.95. **Sale $29.95**. Coupon $24.95.

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

COMMODORE-64

CHALK BOARD
COLOR TOUCH TABLET

Chalk Board Touch-Tablet is a revolutionary new 12" x 12" touch sensitive surface that lets you bypass your COMMODORE-64 Computer keyboard. Just touch the pad and watch your ideas appear on your t.v. screen in living color! Power Pad is drawing pad, color graphics, color canvas and piano keyboard, jigsaw puzzle, game board for any age. A fantastic, entertaining, learning experience! • **Free $29.95 Paint Brush Program!**

List Price $129.90 **Sale $59.00**

LEARNING PAD SALE!!!

• **Bigger — Better — Lower Price Than Koalapad!** • Fantastic Learning Tool • First Graders To Senior Citizens • Learn By Touching Tablet • Color Graphics • Drawing Pad • Game Board • Jigsaw Puzzle • Piano Key Board • Music • Visual Arts • Math • Science • Apple (MacIntosh) Mouse Capability

LEO'S 'LECTRIC PAINT BRUSH. When you use Leo's 'Lectric Paintbrush software, you are ready for magical, multi-colored electronic finger painting. Make your own pictures. Color them. Change them. Save them. List $29.95. **Sale (Free with purchase of CHALKBOARD LEARNING PAD for $59.00)**. (Cart)

LEARNING PAD PROGRAMMING KIT. Once you're familiar with the COMMODORE-64 Computer keyboard and you understand beginning BASIC, the LEARNING PAD programming kit sets you free to develop games and programs! List $29.95. **Sale $19.95**. (Disk)

LOGICMASTER. With over 180,000 different game designs ... and over 200 million secret codes ... LogicMaster is the most fun you've ever had with your powers of reasoning. Solve the codes all by yourself or work together with family or friends. List $39.95. **Sale $29.95**. (Cart)

BEARJAM. As children play this game and feed the friendly animated bear, they sharpen the visual skills so essential for success in learing. BearJam is a great reading-readiness game. List $39.95. **Sale $29.95.** (Cart)

PIANO MAESTRO. Chalk Board's MicroMaestro software turns your PowerPad into a piano keyboard. Touch the keys. You hear the music and see your composition right on the screen. It is the fun way to learn music. List $29.95. **Sale $24.95** (Cart)

LEO'S GOLF LINKS. This golf game for one or more players lets you design each hole, including fairways, roughs, traps and greens. Then using woods, irons and putters, you play the course. List $39.95. **Sale $29.95**. (Cart)

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.

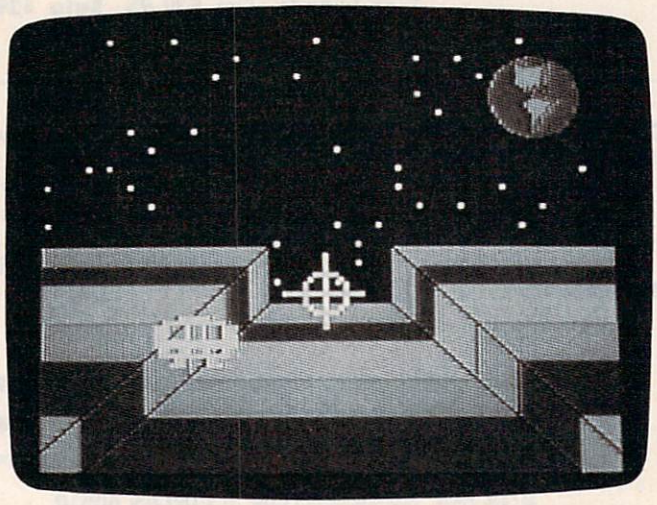
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

```
580 Z=6:FORPS=1TO20:PRINT@PS-1,CHR$
    (128);:PRINT@PS+31,CHR$(128);:G
    OSUB2000:NEXT
585 SOUND 10,2
590 FOR PS=19 TO 1 STEP-1:PRINT@PS+
    6,CHR$(128);:PRINT@PS+38,CHR$(1
    28);:GOSUB2000:NEXT
595 SOUND 50,2:NEXT
597 FORI=0TO16:SOUND255,1:SET(14+I,
    3+I,5):SET(14+I+RND(2)-RND(2),3
    +I+RND(2)-RND(2),RND(8)):NEXT
599 FORI=1TO100:SET(25-RND(10)+RND(
    10),16-RND(10)+RND(10),RND(9)-1
    ):NEXT
600 FORI=1TO50:CLSRND(9)-1:NEXT:FOR
    I=255TO1STEP-17:SOUNDI,1:NEXT
610 PRINT"PLAY AGAIN, HUMANOID? (Y/
    N):";
620 A$=INKEY$:IFA$=""THEN620
630 IF A$="Y" THEN PTS=0:TM=0:GOTO2
    00
640 CLS:END
1000 DATA 1111111111{10 SPACES}1111
     1111111
1010 DATA 2222222221{10 SPACES}1222
     2222222
1020 DATA 3333333321{10 SPACES}1233
     3333333
1030 DATA 1111111321111111111111231
     1111111
1040 DATA 2222221322222222222222231
     2222222
1050 DATA 3333321333333333333333331
     2333333
1060 DATA 1111321111111111111111111
     2311111
1070 DATA 2221322222222222222222222
     2312222
2000 ON Z GOSUB 2100,2200,2210,2400
     ,2500,2600
2010 RETURN
2100 PRINT@PS,CHR$(145);:RETURN
2200 PRINT@PS,CHR$(147)CHR$(146);:R
     ETURN
2210 PRINT@PS,CHR$(151)CHR$(146);:P
     RINT@PS+32,CHR$(148);:RETURN
2400 PRINT@PS,CHR$(150);CHR$(158);C
     HR$(146);:PRINT@PS+32,CHR$(148
     )CHR$(156);:RETURN
```



A game of "Devastator" is just starting. TI version.

```
2500 PRINT@PS,CHR$(151)CHR$(157)CHR
     $(157)CHR$(157)CHR$(146);:PRIN
     T@PS+32,CHR$(148)CHR$(156)CHR$
     (156)CHR$(156);:RETURN
2600 PRINT@PS+1,CHR$(147)CHR$(159)C
     HR$(159)CHR$(147);:PRINT@PS+32
     ,CHR$(148)CHR$(155)CHR$(155)CH
     R$(155)CHR$(159)CHR$(152);:RET
     URN
2999 REM *RANDOM -1,0,1*
3000 R=RND(0):R=(R<.3)-(R>.6):RETUR
     N
```

Program 4:

Devastator – TI-99/4A (Extended BASIC) Version

by Patrick Parrish, Programming Supervisor

```
99 REM DEVASTATOR
100 GOTO 150
110 FOR F=12 TO 14 :: CALL COLOR(F,
    2,1):: NEXT F :: RETURN
120 FOR F=10 TO 16 :: CALL SCREEN(F
    ):: NEXT F :: CALL SCREEN(2)::
    RETURN
130 FOR V=1 TO 30 :: CALL SOUND(D1,
    F1,V,F2,V):: NEXT V :: RETURN
140 FOR ROW=2 TO 7 :: CALL HCHAR(RO
    W,22,32,7):: NEXT ROW :: RETURN
150 RANDOMIZE
160 DIM E$(13)
170 CALL CLEAR :: CALL SCREEN(2)
180 GOSUB 530
190 GOSUB 1030 :: CALL CLEAR :: CAL
    L SCREEN(2)
200 FOR H=2 TO 14 :: CALL COLOR(H,2
    ,2):: NEXT H
210 FOR J=1 TO 4 :: FOR I=1 TO 11 :
    : CALL HCHAR(I,INT(RND*28)+3,46
    ):: NEXT I :: NEXT J
220 DISPLAY AT(13,1):"""""""""a
    {6 SPACES}b"""""""""
230 DISPLAY AT(14,1):"hhhhhhhhhi"
    {6 SPACES}"jhhhhhhhhh"
240 DISPLAY AT(15,1):"ppppppppqh"
    {6 SPACES}"hrpppppppp"
250 DISPLAY AT(16,1):"ppppppppqpha""
    """"bhprpppppppp"
260 DISPLAY AT(17,1):""""""appihhh
    hhhhhjppb""""""
270 DISPLAY AT(18,1):""""""a"pqppppp
    pppppprp"b""""""
280 DISPLAY AT(19,1):"hhhhi""qppppp
    pppppppr""jhhhh"
290 DISPLAY AT(20,1):"hhhih"a""""""
    """""""b"hjhhh"
300 DISPLAY AT(21,1):"hhihha""""""""
    """""""bhhjhh"
310 DISPLAY AT(22,1):"pqhhihhhhhhhh
    hhhhhhhhhhjhhrp"
320 DISPLAY AT(23,1):"qphihhhhhhhhh
    hhhhhhhhhhhjhpr" :: DISPLAY AT(
    24,1):"ppihhhhhhhhhhhhhhhhhhhhh
    hjpp"
330 FOR J=1 TO 2 :: FOR I=12 TO 14
    :: CALL HCHAR(I,INT(RND*6)+14,4
    6):: NEXT I :: NEXT J
340 DISPLAY AT(2,24):CHR$(120)&CHR$
    (121):: DISPLAY AT(3,23):CHR$(1
    22)&CHR$(136)&CHR$(137)&CHR$(12
    3)
```

TI-99/4A Version Notes

The TI-99/4A version of "Devastator" (Program 4) is written in Extended BASIC and requires a joystick. As the game begins, you are cruising above the ominous *Devastator*. A guardian ship from *Devastator* appears. You must eliminate this alien ship and at least nine others that follow in a given period. If you fail, *Devastator* blasts Earth with a lethal laser.

Two levels of difficulty are offered in this version. On either level, you can eliminate the guardian ship by simply positioning the cross hairs over them using the joystick. The main difference between skill levels is the size of these guardian ships (which are actually sprites). The CALL MAGNIFY statement in line 420 produces ships of two sizes. Consequently, on level one, guardian ships are large and can be easily destroyed, but level two features smaller ships which require greater dexterity to eliminate.

The primary game loop for the program is from line 450 to 510. The counter W in line 500 is increased each time through the loop. When W reaches 200, the game is over and Earth is either blasted or not, depending on whether you've destroyed the required number of guardian ships. If the game as written is just too easy or too difficult for you on the skill levels offered, vary the time limit (200) to achieve a comfortable level of play.

The programming techniques used here might aid you in writing your own programs on the TI. You may notice that program execution appears to pause between the title page and the appearance of the playfield (background). Actually, the playfield is being set up, but since the foreground and background colors of all characters are defined as black, nothing appears at this point because the screen color is also black. When all characters on the playfield have been printed, color codes are assigned simultaneously using the CALL COLOR statement so that the entire game field appears at once.

Another trick, also achieved with color coding of characters, gives the game a 3-D effect. The *Devastator* is first printed in lines 220 to 320, using redefined characters from three character sets. By constantly shifting the foreground and background colors of these character sets in line 450, an illusion of movement is produced. Thus, as you watch the screen, you feel that you are actually circling this colossal ship.

```
350 DISPLAY AT(4,22):CHR$(124)&CHR$
    (125)&CHR$(138)&CHR$(139)&CHR$(
    125)&CHR$(126)
360 DISPLAY AT(5,22):CHR$(127)&CHR$
    (125)&CHR$(140)&CHR$(141)&CHR$(
    125)&CHR$(128)
370 DISPLAY AT(6,23):CHR$(129)&CHR$
    (142)&CHR$(143)&CHR$(130)
380 DISPLAY AT(7,24):CHR$(131)&CHR$
    (132)
390 CALL COLOR(12,6,1):: CALL COLOR
    (13,6,1):: CALL COLOR(14,3,6)
400 FOR F=2 TO 8 :: CALL COLOR(F,16
    ,1):: NEXT F
410 CALL SPRITE(#2,108,11,80,80)
420 CALL MAGNIFY(LEVEL):: SPEED=8 :
    : TOL=30 :: IF LEVEL=3 THEN TOL
    =15
430 CALL SPRITE(#1,100,16,100,110)
440 A=9 :: B=10 :: C=11
450 T=A :: A=B :: B=C :: C=T
460 CALL COLOR(A,2,5):: CALL COLOR(
    B,2,14):: CALL COLOR(C,2,7)
470 CALL MOTION(#2,INT(RND*40-20),I
    NT(RND*40-20))
480 CALL JOYST(1,X1,Y1):: CALL MOTI
    ON(#1,-Y1*SPEED,X1*SPEED)
490 CALL COINC(#1,#2,TOL,G):: IF G
    THEN GOSUB 700
500 W=W+1 :: IF W>200 THEN 770
510 GOTO 450
520 REM DEFINE CHARS
530 A$="" :: B$="0102040810204080"
    :: C$="8040201008040201"
540 CALL CHAR(95,B$)
550 FOR I=96 TO 112 STEP 8 :: CALL
    CHAR(I,A$):: CALL CHAR(I+1,B$)
560 CALL CHAR(I+2,C$):: NEXT I
570 FOR I=0 TO 13 :: READ E$(I):: C
    ALL CHAR(120+I,E$(I)):: NEXT I
580 FOR I=0 TO 7 :: READ E$(I):: CA
    LL CHAR(I+136,E$(I)):: NEXT I
590 DATA 00000000000000F7F,000000000
    000F0FE,01030F1F3F7FFFFF
600 DATA 80C0F0F8FCFEFFFF,0001010103030303,FFFFFFFFFFFFFFFF
610 DATA 0080808C0C0C0C0,0303030301010100,C0C0C0C080808000
620 DATA FF7F3F3F1F0F0703,FFFEFCFC8F0E0C0,7F0F000000000000
630 DATA FEF0000000000000,0800667C18666810
640 DATA E0F07F7F7FFFFFFF,0818F8F8F0F8F0F0,7F7F7F3D1C0E0201
650 DATA F0F0908800180000,03070F0F0F070703,F0FFFFFEFCFCF8F0
660 DATA 0303010101010101,E0C0C0C0808080000
670 CALL CHAR(108,"0073FE2E2E2FFFF667F0C1C000000000E0FC474747FFFF73FE3038000000000")
```

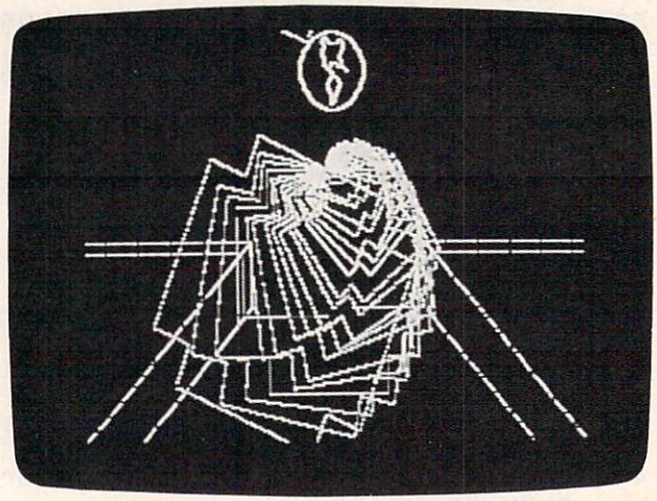
```
680 CALL CHAR(100,"0000000003040808
    FF0808040300000008080808E090888
    8FF888890E0808080")
690 RETURN
700 REM ALIEN SHIP DESTROYED
710 CALL DELSPRITE(#2):: CALL MOTIO
    N(#1,0,0)
720 CALL SCREEN(15):: CALL SCREEN(1
    0)
730 CALL SCREEN(2):: FOR DVOL=1 TO
    24 STEP 4 :: CALL SOUND(100,-7,
    DVOL):: NEXT DVOL
740 CALL SCREEN(2)
750 D=D+1 :: CALL SPRITE(#2,108,11,
    INT(RND*192)+1,INT(RND*256)+1)
760 RETURN
770 IF D<10 THEN 810
780 GOTO 950
790 FOR I=30 TO 1 STEP -2 :: CALL S
    OUND(-1000,-5,I):: NEXT I :: RE
    TURN
800 REM EARTH DESTROYED
810 GOSUB 790 :: FOR I=8 TO 0 STEP
    -1 :: CALL HCHAR(7+I,25-I,95)::
     CALL COLOR(8,INT(RND*8)+9,1)::
     NEXT I
820 FOR J=1 TO 40 :: NEXT J
830 FOR I=8 TO 0 STEP -1 :: CALL HC
    HAR(7+I,25-I,32):: NEXT I
840 GOSUB 120 :: D1=-100 :: F1=-6 :
    : F2=110 :: GOSUB 130 :: GOSUB
    120 :: GOSUB 110 :: GOSUB 140
850 J=0 :: I=0
860 DISPLAY AT(1,23+I):CHR$(133)::
    DISPLAY AT(1,26+J):CHR$(133)
870 DISPLAY AT(2,22+I):CHR$(133)&CH
    R$(133):: DISPLAY AT(2,26+J):CH
    R$(133)&CHR$(133)
880 DISPLAY AT(3,21+I):CHR$(133)&CH
    R$(133)&CHR$(133):: DISPLAY AT(
    4,25+J):CHR$(133)&CHR$(133)&CHR
    $(133)
890 DISPLAY AT(5,22+I):CHR$(133)&CH
    R$(133):: DISPLAY AT(5,25+J):CH
    R$(133)&CHR$(133)&CHR$(133):: G
    OSUB 120
900 DISPLAY AT(6,25+J):CHR$(133)::
    DISPLAY AT(7,23+I):CHR$(133)::
    DISPLAY AT(7,27+J):CHR$(133)
910 DISPLAY AT(8,22+I):CHR$(133)&CH
    R$(133):: DISPLAY AT(9+J,24):CH
    R$(133)&CHR$(133)
920 CALL COLOR(13,9,1):: GOSUB 120
    :: D1=30 :: F1=-6 :: F2=110 ::
    GOSUB 130 :: IF J=1 THEN 940
930 I=-1 :: J=1 :: GOSUB 110 :: GOS
    UB 120 :: GOTO 860
940 FOR F=1 TO 100 :: NEXT F
950 CALL DELSPRITE(ALL):: W=0
960 CALL CLEAR :: CALL SCREEN(2)::
    DISPLAY AT(8,1):"ALIEN SHIPS DE
    STROYED: ";D
970 IF D>HD THEN HD=D
980 DISPLAY AT(13,6):"BEST ROUND: "
    ;HD
990 D=0 :: DISPLAY AT(17,1):"PLAY A
    GAIN, CAPTAIN (Y/N)?"
1000 CALL KEY(0,KEY,ST):: IF ST=0 T
     HEN 1000
1010 IF (KEY=89)+(KEY=121)THEN CALL
     CLEAR :: GOTO 200
1020 DISPLAY AT(21,6):"SO LONG" ::
     FOR I=1 TO 500 :: NEXT I :: ST
     OP
1030 FOR J=2 TO 8 :: CALL COLOR(J,1
     ,1):: NEXT J
1040 PRINT "{4 SPACES}D E V A S T A
      T O R" :: PRINT :: PRINT
1050 PRINT "YOUR MISSION IS TO PROT
     ECT" :: PRINT "EARTH FROM THE
     APPROACHING"
1060 PRINT "DEVASTATOR. SHOOT DOWN
     AT" :: PRINT "LEAST 10 GUARDIA
     N SHIPS TO"
1070 PRINT "ENABLE YOUR COMRADES TO
     " :: PRINT "DESTROY THE DEVAST
     ATOR."
1080 PRINT :: PRINT "YOU HAVE ONLY
     LIMITED TIME" :: PRINT "IN WHI
     CH TO COMPLETE YOUR"
1090 PRINT "MISSION. POSITION YOUR"
      :: PRINT "CROSSHAIR WITH THE
     JOYSTICK."
1100 FOR J=2 TO 8 :: CALL COLOR(J,1
     5,1):: NEXT J
1110 PRINT :: PRINT "ENTER YOUR SKI
     LL LEVEL(1,2),CAPTAIN?" :: ACC
     EPT AT(23,10)BEEP VALIDATE("12
     ")SIZE(1):LEVEL$
1120 LEVEL=5-VAL(LEVEL$)
1130 GOSUB 790
1140 PRINT :: PRINT :: PRINT "THE D
     EVASTATOR IS APPROACH-"
1150 PRINT "ING. GRAB YOUR JOYSTICK
     ," :: PRINT "AND PREPARE TO DO
      BATTLE."
1160 FOR I=1 TO 750 :: NEXT I
1170 RETURN
```

Program 5:

Devastator — Apple II Version

by Todd Koumrian

```
5  TEXT : HOME : VTAB 10: HTAB 15: PRINT
     "READING DATA"
10  GOSUB 8000
15  HGR : POKE  - 16302,0:EX = 140:EY =
     90:Q = 1:DL = 10
```



Another invader is about to appear in the Apple version of "Devastator."

Apple Devastator

Todd Koumrian

"Devastator" for the Apple (Program 5) is a joystick-controlled hi-res game written in Applesoft with several machine language (ML) subroutines. When playing Devastator, you need not hold down the fire button; merely placing the cross hairs on the moving alien interceptor will insure its destruction. However, if you take too long, *Devastator* will have enough time to destroy Earth.

The cross hairs and the alien interceptors are drawn using shape tables. The Applesoft SCALE and ROT commands are used to create the arrival and explosion of the interceptors. The shape table is POKEd in at line 8030 and sits at $300.

Earth and its subsequent destruction are handled by short ML routines. The world-draw routine resides at $1900 and is CALLed once every loop through the main program or whenever the image is garbled. The routine stores the bit image on the screen memory from a data table at $1980 to $1A6F. World-draw OR's the image with what is on the screen and then stores it so that it does not erase what is already there.

The destruction of Earth at the end of the game is handled by an ML routine at $1A70. It stores random garbage in a randomly selected line and byte in screen memory. Since the routine confines the garbage to the area around the image of Earth for a number of cycles and then expands it to the edges of the screen, the explosion appears to expand quickly. The ML random number generator used at $1AFF is a common one that generates random nybbles and masks them together for random byte values. A short lookup table is used by both the world-draw and world-destroy routines to find quickly the addresses of the first 40 lines on the screen. The table lies between $1930 and $197F; its use has been well documented in the past year.

When you're typing in Devastator for the Apple, it is important that the data be absolutely correct. If the data for the shape tables or the world-image has errors, the images will look malformed. If there are errors in the data for the ML routines, the computer will most likely crash or write all over your program. If you have a printer, use it to check the data, and remember to always save your program before you run it.

```
20  SCALE= 1: ROT= 0
25  CALL 6400
30  HCOLOR= 7: HPLOT 0,100 TO 91,100 TO
    91,130 TO 189,130 TO 189,100 TO 27
    9,100
35  HPLOT 91,100 TO 0,191: HPLOT 189,10
    0 TO 279,191
40  HPLOT 91,130 TO 30,191: HPLOT 189,1
    30 TO 249,191
45  GOTO 3999
50 I = I + 1: IF I > 3 THEN I = 1
55  ON I GOTO 100,200,300
100  HCOLOR= 7: GOSUB 1000: HCOLOR= 4: GOSUB
    3000: RETURN
200  HCOLOR= 7: GOSUB 2000: HCOLOR= 4: GOSUB
    1000: RETURN
300  HCOLOR= 7: GOSUB 3000: HCOLOR= 4: GOSUB
    2000: RETURN
1000  HPLOT 0,105 TO 84,105: HPLOT 86,1
    07 TO 86,132: HPLOT 88,134 TO 190,
    134: HPLOT 192,132 TO 192,106: HPLOT
    195,105 TO 279,105
1010  RETURN
2000  HPLOT 0,125 TO 63,125: HPLOT 65,1
    27 TO 65,153: HPLOT 69,155 TO 210,
    155: HPLOT 212,152 TO 212,127: HPLOT
    216,125 TO 279,125
2010  RETURN
3000  HPLOT 0,155 TO 33,155: HPLOT 35,1
    57 TO 35,183: HPLOT 38,185 TO 241,
    185: HPLOT 243,182 TO 243,157: HPLOT
    245,155 TO 279,155
3010  RETURN
3999 X = 140:Y = 90
4000  HCOLOR= 0: SCALE= 1: DRAW 1 AT X,
    Y:PX = X:PY = Y
4010 X =  PDL (0)
4020 Y =  PDL (1): IF Y > 124 THEN Y =
    124
4030  IF Y < 6 THEN Y = 6
4040  IF X > 95 AND X < 165 THEN 4060
4050  IF Y > 94 THEN  HCOLOR= 7:X = PX:
    Y = PY: DRAW 1 AT PX,PY: GOSUB 50
4060  HCOLOR= 7: DRAW 1 AT X,Y
4070  GOSUB 50
4090  IF  ABS (EY - Y) > 9 THEN 4120
4100  IF T = 3 AND EX - X > 3 AND EX -
    X < 13 AND  ABS (EY - Y) < 6 THEN
    5000
4110  IF T = 4 AND EX - X >  - 9 AND EX
    - X < 13 THEN 5000
4120  IF F = 0 THEN 4140
4130  HCOLOR= 0: SCALE= SC: DRAW SS AT
    EX,EY
4140 W =  INT ( RND (1) * 2) + 1: IF W =
    2 THEN W =  - 1
4150 EX = EX + W *  INT ( RND (1) * 30)
    :EY = EY + W *  INT ( RND (1) * 20
    )
4160  IF EX < 0 THEN EX = 0
4170  IF EX > 260 THEN EX = 260
4180  IF EY < 8 THEN EY = 8
4190  IF EY > 121 THEN EY = 121
4200  IF EX > 95 AND EX < 165 THEN 4220

4210  IF EY > 90 THEN EY = 90
4220 DI = DI + Q *  INT ( RND (1) * 20)
    : IF DI > 100 THEN DI = 100: IF  INT
    ( RND (1) * 2) = 0 THEN Q =  - 1
4230  IF DI < 0 THEN DI = 0: IF  INT ( RND
    (1) * 2) = 0 THEN Q = 1
```

```
4240  IF DI < 30 THEN SC = 1:SS = 2
4250  IF DI > 30 AND DI < 70 THEN SC =
    2:SS = 2
4260  IF DI > 71 THEN SC = 1:SS = 3
4270  HCOLOR= 7: SCALE= SC: DRAW SS AT
    EX,EY
4280 T = SS + SC
4290 F = 1
4300 TI = TI + 1
4310  IF TI > DL THEN 10000
4320  CALL 6400
4330  GOTO 4000
5000  HCOLOR= 0: DRAW 1 AT X,Y
5010  HCOLOR= 7: FOR I = SC TO SC + 15:
     SCALE= I: DRAW SS AT EX,EY: POKE
    6952,15 + I: POKE 6953,3: CALL 695
    4: NEXT
5020  HCOLOR= 0: FOR I = SC TO SC + 15:
     SCALE= I: DRAW SS AT EX,EY: POKE
    6952,30 + I: POKE 6953,3: CALL 695
    4: NEXT
5030 SR = SR + 10 * (101 - DI)
5040  CALL 6400
5050 DI = 0
5060 EX =  INT ( RND (1) * 60) + 95:EY =
     INT ( RND (1) * 80): HCOLOR= 7
5070  FOR I = 20 TO 1 STEP  - 1: ROT= 1
    .05 * I - 1: SCALE= I: DRAW 2 AT E
    X,EY: POKE 6952,I + 40: POKE 6953,
    3: CALL 6954: NEXT
5080  HCOLOR= 0: FOR I = 20 TO 1 STEP  -
    1: ROT= 1.05 * I - 1: SCALE= I: DRAW
    2 AT EX,EY: POKE 6952,20 + I: POKE
    6953,3: CALL 6954: NEXT
5090 DD = DD + 1
5100  IF (DD / 4) =  INT (DD / 4) THEN
    DL = DL - 2
5110  IF DL < 2 THEN DL = 2
5120 TI = 0
5130  GOTO 20
8000 I = 768
8010  POKE 232,0: POKE 233,3
8020  READ A: IF A =  - 1 THEN 9030
8030  POKE I,A:I = I + 1: GOTO 8020
9000  DATA  3,0,8,0,31,0,43,0,45,45,45
    ,45,45,64,36,164,146,82,41,45,45,4
    5,45,221,219,219,219,210,54,54,0
9010  DATA  36,37,45,45,46,54,54,55,63
    ,63,60,36,0,36,36,45,36,45,45,36,4
    5,45,45,54,45,45,54,45,54,54,54,54
    ,63
9020  DATA  54,63,63,54,63,63,63,36,63
    ,63,36,63,36,36,0,-1
9030 AD = 6448
9040  FOR I = 0 TO 1: FOR J = 0 TO 1: FOR
    K = 0 TO 7: POKE AD + (I * 16 + J *
    8) + K,32 + (4 * K) + I: NEXT : NEXT
    : NEXT
9050  FOR K = 0 TO 7: POKE AD + (I * 16
    ) + K,32 + (4 * K) + I: NEXT
9060  FOR Q = 0 TO 4: FOR J = 0 TO 7: IF
    (Q / 2) =  INT (Q / 2) THEN W = 0:
     GOTO 9080
9070 W = 1
9080  POKE AD + (I * 15) + 10 + J + (8 *
    Q),128 * W
9090  NEXT : NEXT
9299  FOR I = 6400 TO 6447: READ A: POKE
    I,A: NEXT : GOTO 9399
9300  DATA  32,74,255,169,0,168,170,1
    33,0,164,0,185,48,25,133,4,185,88,
    25,133,3,160,17,189
9310  DATA  128,25,17,3,145,3,232,20
    0,192,23,208,243,230,0,165,0,201,4
    0,208,221,32,63,255,96
9399  FOR I = 6528 TO 6974: READ A: POKE
    I,A: NEXT : RETURN
9400  DATA  0,0,124,15,0,0,0,64,15,124,
    0,0,0,112,1,96,3,0,0,60,14,0,15,0,
    0,14,31,56,28,0,0,7,59,124,56,0,64
    ,3,119,111,112,0,64,1,6,96,96,0,96
    ,1,6,96,96,1,112,0,7,96,64,3

9410  DATA  56,0,3,48,0,7,24,0,3,48,0,6
    ,24,0,3,24,0,6,28,0,7,24,0,14,12,0
    ,6,24,0,12,14,0,6,48,0,28,6,0,102,
    55,0,24,6,0,110,60,0,24,6,0,124,12
    4,0,24,6,0,56,64,1,24
9420  DATA  6,0,112,0,3,24,6,0,96,1,0,2
    4,6,0,64,1,0,24,6,0,96,7,0,24,14,0
    ,112,12,0,28,12,0,48,12,0,12,28,0,
    24,24,0,14,24,0,24,24,0,6,24,0,24,
    28,0,6,56,0,56,12,0,7

9430  DATA  112,0,48,14,64,3,96,1,112
    ,6,96,1,64,1,96,7,96,0,64,3,96,3,1
    12,0,0,7,96,3,56,0,0,14,96,3,28,0,
    0,60,96,1,15,0,0,112,1,96,3,0,0,64
    ,15,124,0,0,0,0,124,15,0,0
9500  DATA  32,74,255,169,0,133,1,133,
    5,162,5,181,78,149,6,202,208,249,1
    69,0,133,4,32,180,26,230,4,165,4,2
    01,127,208,245,230,1,165,1,201,3,2
    08,233,169,0,133,4
9510  DATA  32,219,26,230,4,165,4,201,
    127,208,245,230,5,165,5,201,5,208,
    233,32,63,255,96,32,255,26,41,63,2
    01,39,16,247,170,189,48,25,133,3,1
    89,88,25,133,2,32,255

9520  DATA  26,41,7,201,7,240,247,24,1
    05,17,168,32,255,26,145,2,96,32,25
    5,26,41,63,201,39,16,247,170,189,4
    8,25,133,3,189,88,25,133,2,32,255,
    26,41,63,201,39,16
9530  DATA  247,168,32,255,26,145,2,96
    ,32,14,27,133,12,32,14,27,10,10,10
    ,10,5,12,96,56,165,7,101,10,101,11
    ,133,6,162,4,181,6,149,7,202,16,24
    9,165,6,41,15,141,48,192,96
9600  DATA  0,0,173,48,192,136,208,5,20
    6,41,27,240,9,202,208,245,174,40,2
    7,76,42,27,96

10000  HCOLOR= 7: SCALE= 1: FOR I = 127
     TO 20 STEP  - 5: ROT= I: DRAW 2 AT
    135,I: HCOLOR= 0: ROT= I + 5: DRAW
    2 AT 135,I + 5: HCOLOR= 7: NEXT
10010  FOR I = 1 TO 7 STEP 2: HPLOT 135
     + I,0 TO 135 + I,130: HPLOT 135 -
    I,0 TO 135 - I,130: NEXT
10020  CALL 6768: HOME : VTAB 21: HTAB
    7: PRINT "YOU MADE "SR" POINTS BEF
    ORE": VTAB 22: HTAB 9: PRINT "PLAN
    ETARY DESTRUCTION"
10030  VTAB 23: PRINT "PRESS BUTTON (0)
     FOR ANOTHER CHANCE TO";
10040  VTAB 24: HTAB 15: PRINT "SAVE EA
    RTH";
10050  POKE  - 16301,0
10060  IF  PEEK ( - 16287) > 127 THEN 1
    00060
10070  IF  PEEK ( - 16287) < 128 THEN 1
    00070
10080  CLEAR : GOTO 15
```

IBM Notes: Devastator

Charles Brannon, Program Editor

The *Devastator*, an alien ship of incredible power, is now approaching the earth. The *Devastator* roams the galaxy, destroying planets and absorbing matter-energy transformations. Unfortunately, it's now Earth's turn to be the matter.

The combined technology of the planet has managed to assemble a primitive ship, one that can at best discourage the *Devastator*. You are the pilot of that ship, mankind's last, best hope.

You've been briefed thoroughly: The *Devastator* sends out ten ships, one at a time. Each ship plants an explosive satellite above the earth. After all ten charges have been laid, the *Devastator* detonates them, destroying the planet utterly. It didn't expect to encounter you, though....

You'll need an IBM PC with BASICA (advanced BASIC), or a PCjr with Cartridge BASIC, as well as a joystick, to play "Devastator." After you RUN the game, read the instructions to familiarize yourself with the game. To begin play, hold the joystick to the lower right corner, then press the button. This lets the program calibrate itself to your joystick (since the range of the joysticks is not standard).

After a pause, while the game is being set up (the background colors will change to assure you your machine's not dead), the main viewscreen appears, inside dotted lines. You're orbiting the massive *Devastator*. Terra Firma is in the upper left corner of the viewscreen, and a dreaded alien ship is hovering about. Move the cross hairs with your joystick, center it on the alien, then press fire. If you made a hit, the screen will flash red and a new alien will appear. But if you miss, the alien ship darts away, making it harder to reaim. The alien ship will plant its charge after ten seconds. However, the more ships you hit, the faster they get.

At first the *Devastator* hardly notices you, but after you begin to destroy the ships, the *Devastator* modifies them to reach Earth faster. Every time you hit five ships, future ships will reach Earth a second sooner. Your control panel shows you a countdown of time remaining before the charge is planted. Each time an alien lays a charge, Earth will flash, and the deadly ring around Earth becomes more complete. When ten charges have been set, Earth shudders in nil-space, then flashes outward at the speed of light. You may not have saved Earth, but at least it went out with a bang!

Programming Tips

The program uses the medium resolution, four color mode (SCREEN 1). All the animation is done using PUT and GET. First, three views of the trench are drawn. Each one starts with a different color, so when they are viewed in succession, you get the illusion of moving bands, which in turn makes you feel like you are orbiting the *Devastator*. Each view is saved in an array (with GET), then displayed with PUT. The cross hairs, the alien, and the planet Earth are also drawn, then nabbed with GET.

To animate, you must erase the old image, redraw the image at the new location, erase, draw, etc. In drawing and erasing, though, it's too easy to erase the underlying background. The trick is how you lay down the image. If you just place it on the screen, you are overlaying and destroying the dots under the image. Instead, you can use a quasi-mathematical function called XOR (exclusive OR) to both draw and erase the ship. Let's follow XOR with a binary example.

Let's say the image is one byte wide and one line high: 10101010. This would create a dotted line in high resolution, or a colored line in medium. Underneath the image might be a single dot: 00100000. When the two bytes are XOR'd together:

```
XOR 00100000  (background)
    10101010  (shape)
    10001010  (new background)
```

(The rule for XOR is 0 XOR 0=0, 0 XOR 1=1, 1 XOR 0=1, 1 XOR 1=0.)

Now watch the magic as we XOR the answer back with the image:

```
    10001010  (new background)
XOR 10101010  (original shape)
    00100000  (restored background)
```

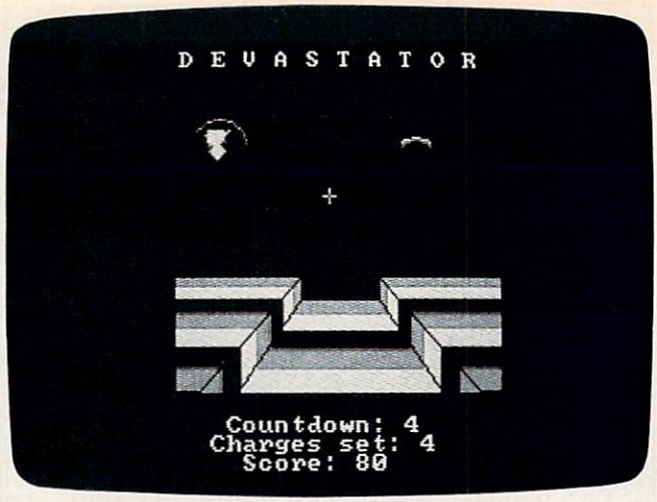
The image is erased, but the original dot is back! The same idea applies to a shape made up of lots of bytes. You can XOR it against the background, then XOR again to restore the background (and erase the shape).

Program 6: PC And PCjr

by Charles Brannon, Program Editor

```
1 SCREEN 0,0,0:CLS:GOSUB 3000:GOSUB 4000
:STRIG ON:KEY OFF:GOSUB 2000
2 SCREEN 1:COLOR 0,0:DEFINT A-Z
3 DIM SHAPE1(1002),SHAPE2(1002),SHAPE3(1
002),EARTH(52),CROSS(10),ALIEN(30)
4 CX!=155/CX!:CY!=63/CY!:GOSUB 1000
5 X=0:Y=0:XP=80:YP=100:TIME.LIMIT=10:SCO
RE!=0
6 TIME$="00:00:00":LOCATE 21,14:PRINT US
ING "Countdown:##";TIME.LIMIT
7 AX=80+140*RND:AY=30+60*RND:PUT (AX,AY)
,ALIEN
10 FOR I=1 TO 3
20 IF I=1 THEN PUT (XP,YP),SHAPE1,PSET
30 IF I=2 THEN PUT (XP,YP),SHAPE2,PSET
40 IF I=3 THEN PUT (XP,YP),SHAPE3,PSET
60 PUT (X,Y),CROSS:X=STICK(0)*CX!+78:Y=S
TICK(1)*CY!+30:PUT (X,Y),CROSS
70 PUT (AX,AY),ALIEN:Z!=RND:AX=AX+4*(Z!<
.3)*(AX<220)-4*(Z!>.6)*(AX>XP):Z!=RND:AY
=AY+4*(Z!<.3)*(AY<80)-4*(Z!>.6)*(AY>30)
80 PUT (AX,AY),ALIEN
90 IF TIMER>=TIME.LIMIT THEN 200
100 IF STRIG(1)=0 THEN NEXT:LOCATE 21,24
:PRINT USING "##";TIME.LIMIT-TIMER:GOTO
10
110 IF ABS(AX-X+4.5)>7 OR ABS(AY-Y+1.5)>
6 THEN PUT (AX,AY),ALIEN:GOTO 7
115 SAVE.TIME=TIMER
120 FOR I=1 TO 15:PUT (AX,AY),ALIEN:COLO
R 4:COLOR 0:SOUND 100+10*RND(1),.5:NEXT
125 ALIENS=ALIENS+1:SCORE!=SCORE!+10*(TI
ME.LIMIT-SAVE.TIME):IF (ALIENS MOD 5)=0
THEN IF TIME.LIMIT >1 THEN TIME.LIMIT=TI
ME.LIMIT-1
130 ALIENS=ALIENS+1:LOCATE 23,15:PRINT"S
core:";SCORE!:GOTO 6
200 PUT (AX,AY),ALIEN:FOR I=1 TO 10:SOUN
D I*100,1:PUT (90,31),EARTH:NEXT
201 RADS1!=PI!*36*CHARGES/180
205 CHARGES=CHARGES+1:RADS!=PI!*36*CHARG
ES/180:CIRCLE (100,41),13,2,RADS1!,RADS!

210 LOCATE 22,13:PRINT"Charges set:";CHA
RGES:IF CHARGES<10 THEN 6
220 'Earth explodes
230 FOR I!=1 TO 30 STEP .3:PUT (90-I!*RN
D+I!*RND,31-I!*RND+I*RND),EARTH:SOUND 10
000*RND+100,.1:COLOR 15*RND,RND:NEXT
232 FOR I=1 TO 40 STEP 2 :CIRCLE (100,41
),I:SOUND 100+150*RND,.1:NEXT
235 IF SCORE!>HSCORE! THEN HSCORE!=SCORE
!
240 CLS:COLOR 10,0:PRINT"Score:";SCORE!:
PRINT:PRINT"High Score:";HSCORE!:PRINT:P
RINT"Better luck next time...":PRINT
250 PRINT"Press fire button to play agai
n."
260 IF STRIG(1)=0 THEN 260
270 CLS:COLOR 0,0:GOSUB 1165:ALIENS=0:CH
ARGES=0:GOTO 5
1000 DEF SEG=&HF000:IF PEEK(&HFFFE)=&HFD
 THEN A=INP(&H3DA):OUT &H3DA,0:OUT &H3DA
,2 ELSE OUT &H3D8,2
1001 FOR BASE=1 TO 3:CLS:COLOR RND*16
1005 X=60:Y=0:Z=10:C=3-BASE:R=3:M=2
1010 FOR I=1 TO 7:C=-C*(C<3)+1
1020 LINE (0,Y)-(X,Y),C:LINE -(X,Y+Z),C:
LINE -(160-X,Y+Z),C:LINE -(160-X,Y),C:LI
NE-(160,Y),C
1030 LINE (0,Y+R)-(X-R,Y+R),C:LINE -(X-R
,Y+Z+R+M),C:LINE -(160-X+R,Y+Z+R+M),C:LI
NE -(160-X+R,Y+R),C:LINE-(160,Y+R),C
1035 LINE (160,Y)-(160,Y+R),C
1040 X=X-R-1:Y=Y+R+1:Z=Z+M:R=R+.7
1050 NEXT:COLOR RND*16
1060 Y=0:R=3:C=3-BASE
1070 FOR I=1 TO 7:C=-C*(C<3)+1
1080 PAINT (1,Y+2),C,C
1090 Y=Y+R+1:R=R+.5
1100 NEXT
1110 LINE (60,0)-(X,Y),0:LINE (100,0)-(1
60-X,Y),0
1120 LINE (60,10)-(X,Y+Z),0:LINE (100,10
)-(160-X,Y+Z),0
1130 IF BASE=1 THEN GET (0,0)-(159,49),S
HAPE1
1140 IF BASE=2 THEN GET (0,0)-(159,49),S
HAPE2
1150 IF BASE=3 THEN GET (0,0)-(159,49),S
HAPE3
1160 NEXT:COLOR 0
1162 CLS:CIRCLE (10,10),10,1:PAINT (10,1
0),1,1:DRAW "c3bm4,4r8drerg3dnfg2f3g4h4e
2hguhu2hebf3p3,3"
1163 GET (0,0)-(19,19),EARTH:CLS
1165 LINE (3,0)-(3,6):LINE (0,3)-(6,3):P
RESET (3,3):GET (0,0)-(6,6),CROSS
1170 LINE (78,29)-(241,151),1,B,13107
1175 PUT (90,31),EARTH
1176 PI!=3.141593
1177 CIRCLE (30,10),5,1,2*PI!,PI!,.5:LIN
E (25,10)-(35,10),1:PAINT (28,8),1,1:CIR
CLE (30,13),7,3,2*PI!,PI!,.4:CIRCLE (30,
14),7,2,2*PI!,PI!,.4
1180 GET (22,4)-(38,14),ALIEN:PUT (22,4)
,ALIEN:BEEP
1185 DEF SEG=&HF000:IF PEEK(&HFFFE)=&HFD
 THEN A=INP(&H3DA):OUT &H3DA,0:OUT &H3DA
,10 ELSE OUT &H3D8,10
1190 LOCATE 1,11:PRINT"D E V A S T A T O
 R"
1200 RETURN
```



The IBM version of "Devastator."

```
2000 SCREEN 0,1:WIDTH 40:COLOR ,7,7:CLS:
COLOR 4:LOCATE 1,15,0:PRINT"DEVASTATOR":
COLOR 0
2010 PRINT:PRINT"The Devastator, an alie
n ship of
2020 PRINT"incredible power, is now appr
oaching
2030 PRINT"the Earth.  Its mission is to
 utterly
2040 PRINT"destroy the planet, and absor
b the
2050 PRINT"energy released by the explos
ion
2060 PRINT"Earth's primitive technology
can
2070 PRINT"assemble only one spaceship t
hat
2080 PRINT"can hope to discourage the De
vastator.
2090 PRINT:PRINT"You, as the pilot of th
at craft,
2100 PRINT"are truly Earth's last hope."
:PRINT
2110 PRINT"The Devastator sends out smal
l ships
2120 PRINT"to plant the charges that wil
l effect
2130 PRINT"Earth's demise.  They will at
tempt to
2140 PRINT"evade you, but cannot shoot b
ack.
2150 PRINT"You have only a few seconds t
o shoot
2160 PRINT"each ship before it plants a
charge.
2170 PRINT"Once ten charges have been pl
anted,
2180 COLOR 1:PRINT"it's too late."
2190 LOCATE 25,1:COLOR 31:PRINT"Hold sti
ck to lower right,press button.";
2200 A=RND
2205 IF STICK(0)>CX! THEN CX!=STICK(0)
2207 IF STICK(1)>CY! THEN CY!=STICK(1)
2208 IF STRIG(1)=0 THEN 2200
2210 RETURN
3000 DEF SEG=0:IF (PEEK(&H410) AND &H30)
<>&H30 THEN DEF SEG:RETURN
3010 SCREEN 0:PRINT"Color Adaptor Requir
ed."
3020 END
4000 ON ERROR GOTO 4040
4010 PLAY "P16"
4020 GOTO 4060
4030 SCREEN 0:COLOR 31,0,0
4040 PRINT "Advanced BASIC (BASICA) Requ
ired.":COLOR 7:RESUME 4050
4050 ON ERROR GOTO 0:END
4060 ON ERROR GOTO 0:RETURN
```

Program 7: 64 Devastator—BASIC Portion

by Gregg Peele, Assistant Programming Supervisor

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

```
50 POKE56,49:CLR                    :rem 179
60 PRINT"{CLR}{12 DOWN}SELECT
   {RVS}1{OFF} OR {RVS}2{OFF} PLAYER GAME
   "                                :rem 84
70 GETA$:IFA$<"1"ORA$>"2"THEN70   :rem 219
80 POKE828,ASC(A$)-49:POKE829,2   :rem 246
```

Notes For The 64 Version

The 64 version of "Devastator" uses machine language, coupled with sprite and character graphics, to produce a realistic battle scenario. You must defend the earth against all invaders. To insure the earth's safety, you must strike your opponent's ship at least once every ten seconds. Failing this challenge will place the earth in imminent danger.

Several options for game play are available. Initially, you may select either a one- or a two-player game. The one-player game pits you against a computer-controlled ship. This ship evades your attack with random movements. Choosing the two-player option pits you against an opponent who is actively avoiding your attack.

Player one (in either the one- or two-player game) must use a joystick in port 2 to move a crosshair around the screen. (In the two-player game, player two controls the alien ship with a joystick in port 1.) When the crosshair comes in contact with the alien ship, you must fire to achieve a successful strike against the opposing ship. Ten points are awarded for each successful strike. A score of 1000 will save the earth.

The game has three levels of difficulty. You can change levels at any time by pressing the top three function keys. The f1 key gives the lowest level, f3 the second highest, and f5 the most difficult level. The SHIFT/LOCK key can be used to increase or decrease the size of your crosshair, which also affects the difficulty of the game.

To use Devastator for the 64, you must first enter Program 8 using MLX. The starting address for this program is 49152 and the ending address is 50891. After saving this program, type in Program 7 and save it. To run Devastator, first load the program that you created with MLX like this:

LOAD *"program name"*,8,1

for disk, or

LOAD *"program name"*,1,1

for tape. Then type NEW (hit RETURN) and load and run Program 7.

```
100 POKE53281,0:POKE52992,0:POKE646,1:POK
    E53275,8                          :rem 40
120 SI= 54272:FOR T= SI TO SI+24:POKET,0:
    NEXT                              :rem 32
125 POKESI+24,15:POKESI+5,17:POKESI+6,245
    :POKESI,100:POKESI+1,100          :rem 140
150 PRINT"{CLR}":FOR T= 1030TO1444STEP 41
    :POKET,223:POKET+54272,1:NEXT:rem 195
200 FOR T= 1057TO 1484STEP 39:POKET,233:P
    OKET+54272,1:NEXT                 :rem 54
250 FOR T= 1991TO 1624STEP-39:POKET,105:P
    OKET+1,233:POKET+54272,1          :rem 205
300 POKET+54272+1,1:NEXT              :rem 119
350 FOR T= 2016TO1624STEP-41:POKET,95:POK
    ET-1,223:POKET+54272,1            :rem 149
400 POKET+54272-1,1:NEXT              :rem 122
450 FOR T= 1480 TO 1480+120STEP40:POKET,1
    60:POKET+54272,1:POKET+7,160      :rem 87
500 POKET+54279,1:NEXT                :rem 36
550 J=1023:Z=J:A=1:TR=40:GOSUB 600:J=1064
    :Z=J:A=-1:TR=40:GOSUB 600:GOTO800
                                      :rem 85
600 J=J+A:IF J>2024 THEN RETURN       :rem 251
650 IF PEEK(J)<>32THENZ=Z+TR:J=Z:GOTO600
                                      :rem 222
700 POKEJ,160:POKEJ+54272,1           :rem 18
750 GOTO600                           :rem 107
800 CL=12:FORT=1640TO 2023:M=PEEK(T):IFM<
    >32ANDM<>233ANDM<>223THEN1050:rem 137
850 IF M=32THENPOKET,247              :rem 150
900 IFT>1754 THEN CL=15               :rem 236
950 IFT >1868 THEN CL=1               :rem 194
1000 POKET+54272,CL                   :rem 46
1050 NEXT                             :rem 5
1150 FOR T= 1600TO2023:IF PEEK(T)=32 THEN
     POKET+54272,12                   :rem 21
1200 NEXT                             :rem 2
1210 SYS49152:PRINT"{HOME}{10 RIGHT}SCORE
     "                                :rem 132
1250 SYS49200:SYS49424:IF PEEK(52992)=255
     THEN 1283                        :rem 38
1260 IF PEEK(52992)=1THEN1275         :rem 69
1270 GOTO 1250                        :rem 203
1275 SYS50871                         :rem 211
1280 PRINT"{HOME}{13 RIGHT}{2 DOWN}{WHT}Y
     OU SAVED THE{DOWN}{9 LEFT}EARTH":GOT
     O1287                            :rem 239
1283 POKE 54276,129:FOR T= 200 TO 202:POK
     E2043,T:POKE54273,RND(0)*60+40
                                      :rem 24
1284 FORG=1TO 254:POKE54273,RND(0)*60+40:
     NEXT:NEXT:POKE53248+21,0         :rem 18
1285 POKE54276,128:FOR T= 1TO1500:NEXT
                                      :rem 147
1286 PRINT"{HOME}{13 RIGHT}{2 DOWN}{WHT}T
     HE EARTH HAS{DOWN}{14 LEFT}BEEN DEST
     ROYED."                          :rem 40
1287 PRINT"{13 RIGHT}{WHT}HIT RETURN TO
     {DOWN}{13 LEFT}PLAY AGAIN OR"
                                      :rem 202
1288 PRINT"{15 RIGHT}{WHT}PRESS 'Q'
     {DOWN}{10 LEFT}TO QUIT.":POKE198,0
                                      :rem 54
1289 GET A$:IF A$<>CHR$(13)AND A$<>"Q"AND
      A$<>CHR$(141)THEN1289           :rem 91
1290 IF A$<>CHR$(13)ANDA$<>CHR$(141) THEN
     PRINT"{CLR}":POKE53248+21,0:END
                                      :rem 135
1291 FOR T= 1024 TO 1400:IF PEEK(T)<160 T
     HEN POKET,32                     :rem 29
1292 NEXT:SYS49152:POKE52992,0:GOTO 1210
                                      :rem 82
```

Program 8:
64 Devastator—Machine Language Portion

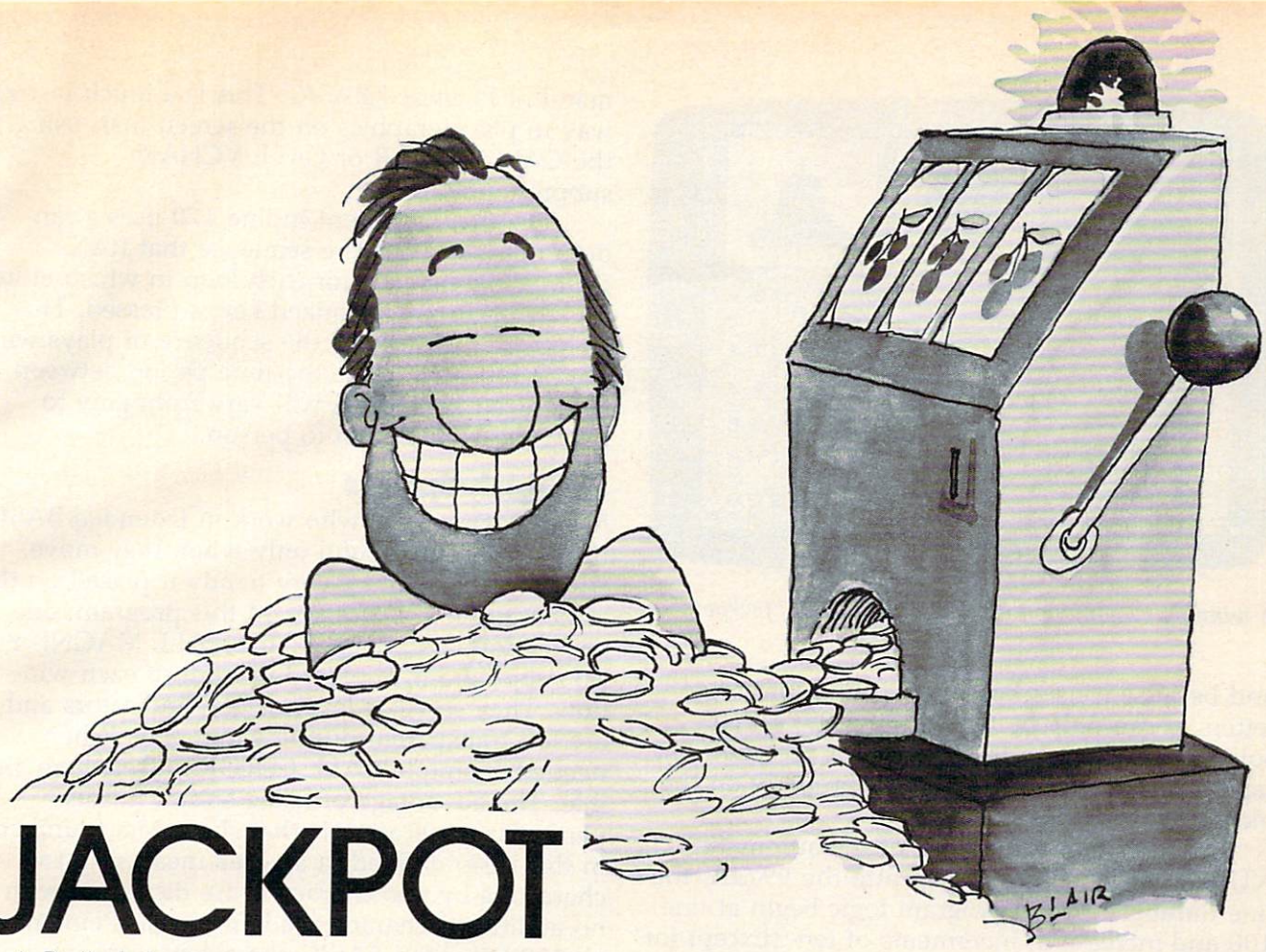
by Gregg Peele, Assistant Programming Supervisor

```
49152 :169,000,133,160,133,161,244
49158 :133,162,141,216,207,141,238
49164 :217,207,032,149,197,032,078
49170 :076,193,032,051,194,169,221
49176 :012,141,215,207,169,017,017
49182 :141,253,207,169,009,141,182
49188 :252,207,169,030,141,251,062
49194 :207,169,022,141,250,207,014
49200 :169,000,141,254,207,032,083
49206 :181,192,206,253,207,173,242
49212 :253,207,240,008,169,001,170
49218 :141,254,207,076,077,192,245
49224 :169,016,141,253,207,032,122
49230 :181,192,169,000,141,254,247
49236 :207,032,187,192,206,252,136
49242 :207,173,252,207,240,008,153
49248 :169,001,141,254,207,076,176
49254 :109,192,169,016,141,252,213
49260 :207,032,187,192,169,000,127
49266 :141,254,207,032,193,192,109
49272 :238,251,207,173,251,207,167
49278 :201,039,176,008,169,001,208
49284 :141,254,207,076,143,192,121
49290 :169,023,141,251,207,032,193
49296 :193,192,169,000,141,254,069
49302 :207,032,199,192,238,250,244
49308 :207,173,250,207,201,039,209
49314 :176,008,169,001,141,254,143
49320 :207,076,177,192,169,023,244
49326 :141,250,207,032,199,192,171
49332 :096,172,253,207,076,202,162
49338 :192,172,252,207,076,202,007
49344 :192,172,251,207,076,202,012
49350 :192,172,250,207,169,000,164
49356 :133,251,169,004,133,252,122
49362 :173,254,207,240,013,177,250
49368 :251,201,160,208,020,169,201
49374 :221,145,251,076,241,192,068
49380 :177,251,201,221,208,007,013
49386 :169,160,145,251,076,241,252
49392 :192,024,165,251,105,040,249
49398 :133,251,165,252,105,000,128
49404 :133,252,056,165,251,233,062
49410 :000,141,255,207,165,252,254
49416 :233,008,013,255,207,144,100
49422 :195,096,169,000,133,253,092
49428 :169,006,133,254,160,000,230
49434 :177,253,201,247,240,018,138
49440 :201,192,240,007,201,239,088
49446 :240,017,076,061,193,169,026
49452 :239,145,253,076,061,193,243
49458 :169,192,145,253,076,061,178
49464 :193,169,247,145,253,200,239
49470 :208,218,230,254,165,254,111
49476 :201,008,208,208,032,183,140
49482 :198,096,160,000,185,115,060
49488 :193,153,064,003,200,192,117
49494 :192,208,245,169,014,141,031
49500 :248,007,169,010,141,028,183
49506 :208,169,003,141,037,208,096
49512 :169,009,141,040,208,169,072
```

```
49518 :013,141,038,208,096,000,094
49524 :000,000,000,000,000,000,116
49530 :000,000,000,000,000,000,122
49536 :000,000,000,000,000,000,128
49542 :000,000,000,000,000,000,134
49548 :000,000,000,020,000,000,160
49554 :085,000,001,085,064,005,130
49560 :125,080,031,255,244,117,236
49566 :085,093,118,150,157,006,255
49572 :150,144,001,085,064,000,096
49578 :000,000,000,000,000,000,170
49584 :000,000,000,000,000,000,176
49590 :000,000,000,000,000,000,182
49596 :000,000,000,000,000,000,188
49602 :000,024,000,000,024,000,242
49608 :000,024,000,000,024,000,248
49614 :003,255,192,003,255,192,082
49620 :000,024,000,000,024,000,004
49626 :000,024,000,000,024,000,010
49632 :000,000,000,000,000,000,224
49638 :000,000,000,000,000,000,230
49644 :000,000,000,000,000,000,236
49650 :255,255,255,255,255,255,236
49656 :255,192,000,003,192,000,122
49662 :003,192,000,003,192,000,132
49668 :003,192,000,003,192,000,138
49674 :003,192,000,003,192,000,144
49680 :003,192,000,003,192,000,150
49686 :003,192,000,003,192,000,156
49692 :003,192,000,003,192,000,162
49698 :003,192,000,003,192,000,168
49704 :003,192,000,003,255,255,236
49710 :255,255,255,255,255,169,210
49716 :173,141,000,208,141,001,204
49722 :208,141,002,208,141,003,249
49728 :208,141,224,207,141,226,187
49734 :207,169,003,141,240,207,013
49740 :169,000,141,016,208,141,239
49746 :225,207,141,227,207,169,234
49752 :013,141,249,007,169,010,165
49758 :141,239,207,120,169,110,056
49764 :141,020,003,169,194,141,000
49770 :021,003,088,096,162,000,220
49776 :032,126,194,162,001,032,147
49782 :126,194,032,229,196,076,203
49788 :049,234,169,015,141,021,241
49794 :208,169,014,141,248,007,149
49800 :169,007,141,039,208,169,101
49806 :010,141,040,208,169,015,213
49812 :141,250,007,173,001,208,160
49818 :141,005,208,173,000,208,121
49824 :141,004,208,173,016,208,142
49830 :041,001,240,011,169,004,120
49836 :013,016,208,141,016,208,006
49842 :076,189,194,169,251,045,078
49848 :016,208,141,016,208,173,178
49854 :060,003,208,055,224,001,229
49860 :208,051,206,248,207,173,009
49866 :248,207,208,024,169,255,033
49872 :141,015,212,169,128,141,246
49878 :018,212,173,027,212,041,129
49884 :007,141,246,207,173,215,185
49890 :207,141,248,207,173,246,168
49896 :207,168,185,240,194,076,022
49902 :252,194,014,013,011,010,220
49908 :009,007,006,005,015,189,219
49914 :000,220,041,015,157,228,143
49920 :207,056,169,015,253,228,160
49926 :207,157,232,207,173,141,099
49932 :002,208,014,169,005,013,167
49938 :029,208,141,029,208,141,006
49944 :023,208,076,051,195,201,010
49950 :001,208,018,173,001,220,139
49956 :041,016,240,011,169,200,201
49962 :045,029,208,141,029,208,190
49968 :141,023,208,165,197,201,215
49974 :004,208,008,169,002,141,074
49980 :061,003,076,086,195,201,170
49986 :005,208,008,169,003,141,088
49992 :061,003,076,086,195,201,182
49998 :006,208,005,169,005,141,100
50004 :061,003,189,232,207,168,176
50010 :224,001,208,002,162,002,177
50016 :152,010,168,185,109,195,147
50022 :072,185,108,195,072,096,062
50028 :227,196,183,196,187,196,013
50034 :227,196,195,196,199,196,043
50040 :206,196,227,196,191,196,052
50046 :220,196,213,196,169,120,216
50052 :221,001,208,176,010,056,036
50058 :189,001,208,237,061,003,069
50064 :157,001,208,096,169,229,236
50070 :221,001,208,144,010,024,246
50076 :189,001,208,109,061,003,215
50082 :157,001,208,096,056,189,101
50088 :224,207,233,045,157,228,238
50094 :207,189,225,207,233,001,212
50100 :029,228,207,144,013,169,202
50106 :045,157,224,207,169,001,221
50112 :157,225,207,076,216,195,244
50118 :024,189,224,207,109,061,244
50124 :003,157,224,207,189,225,185
50130 :207,105,000,157,225,207,087
50136 :056,189,224,207,233,000,101
50142 :157,228,207,189,225,207,155
50148 :233,001,029,228,207,144,046
50154 :019,224,002,240,034,173,158
50160 :016,208,009,001,141,016,119
50166 :208,189,224,207,157,000,207
50172 :208,096,224,002,240,030,028
50178 :173,016,208,041,254,141,067
50184 :016,208,189,224,207,157,241
50190 :000,208,096,173,016,208,203
50196 :009,002,141,016,208,189,073
50202 :224,207,157,000,208,096,150
50208 :173,016,208,041,253,141,096
50214 :016,208,189,224,207,157,015
50220 :000,208,096,056,189,224,049
50226 :207,233,045,157,228,207,103
50232 :189,225,207,233,000,029,171
50238 :228,207,176,013,169,044,131
50244 :157,224,207,169,000,157,214
50250 :225,207,076,097,196,056,163
50256 :189,224,207,237,061,003,233
50262 :157,224,207,189,225,207,015
50268 :233,000,157,225,207,056,202
50274 :189,224,207,233,000,157,084
50280 :228,207,189,225,207,233,113
50286 :001,029,228,207,144,019,226
50292 :224,002,240,034,173,016,037
50298 :208,009,001,141,016,208,193
50304 :189,224,207,157,000,208,089
50310 :096,224,002,240,030,173,131
50316 :016,208,041,254,141,016,048
50322 :208,189,224,207,157,000,107
50328 :208,096,173,016,208,009,094
50334 :002,141,016,208,189,224,170
50340 :207,157,000,208,096,173,237
50346 :016,208,041,253,141,016,077
50352 :208,189,224,207,157,000,137
50358 :208,096,032,130,195,096,171
50364 :032,148,195,096,032,166,089
```

```
50370 :195,096,032,047,196,096,088
50376 :032,130,195,032,047,196,064
50382 :096,032,148,195,032,047,244
50388 :196,096,032,148,195,032,143
50394 :166,195,096,032,130,195,008
50400 :032,166,195,096,096,173,214
50406 :030,208,141,242,207,206,240
50412 :240,207,024,173,240,207,047
50418 :105,030,141,001,212,173,136
50424 :240,207,240,003,076,078,068
50430 :197,169,000,141,033,208,234
50436 :141,032,208,169,003,141,186
50442 :240,207,173,000,220,041,123
50448 :016,240,013,169,013,141,096
50454 :249,007,169,001,141,244,065
50460 :207,076,078,197,173,244,235
50466 :207,240,041,169,008,141,072
50472 :004,212,169,033,141,004,091
50478 :212,169,032,141,004,212,048
50484 :169,000,141,244,207,173,218
50490 :242,207,041,001,240,014,035
50496 :206,239,207,169,002,141,004
50502 :033,208,141,032,208,032,212
50508 :091,197,169,002,197,161,125
50514 :208,006,169,088,197,162,144
50520 :240,041,096,169,202,141,209
50526 :249,007,169,020,141,240,152
50532 :207,032,157,198,173,239,082
50538 :207,240,001,096,169,010,061
50544 :141,239,207,206,215,207,047
50550 :173,215,207,201,003,144,037
50556 :001,096,169,001,076,133,088
50562 :197,169,255,141,000,207,075
50568 :120,169,049,141,020,003,126
50574 :169,234,141,021,003,088,030
```

```
50580 :096,160,000,185,221,197,239
50586 :153,000,050,200,192,193,174
50592 :208,245,169,008,013,021,056
50598 :208,141,021,208,169,160,049
50604 :141,006,208,169,060,141,129
50610 :007,208,169,008,013,029,100
50616 :208,141,029,208,013,023,038
50622 :208,141,023,208,013,028,043
50628 :208,141,028,208,169,014,196
50634 :141,037,208,169,005,141,135
50640 :038,208,169,003,141,042,041
50646 :208,169,200,141,251,007,166
50652 :096,000,000,000,002,169,231
50658 :000,002,254,064,011,255,044
50664 :144,043,255,208,042,255,155
50670 :164,170,254,169,170,190,075
50676 :169,170,186,169,186,186,030
50682 :169,170,254,169,170,254,156
50688 :173,043,255,180,042,255,180
50694 :180,042,254,180,010,254,158
50700 :144,002,185,064,000,169,064
50706 :000,000,000,000,000,000,018
50712 :000,000,000,000,000,000,024
50718 :000,000,000,136,000,000,166
50724 :254,000,011,207,128,011,135
50730 :255,192,002,207,036,034,000
50736 :254,040,040,188,169,042,013
50742 :050,169,128,000,000,170,059
50748 :014,169,034,062,044,008,135
50754 :252,132,040,207,048,002,235
50760 :254,128,002,206,000,002,152
50766 :185,000,000,000,000,000,007
50772 :000,000,000,000,000,000,084
50778 :000,000,000,000,000,000,090
50784 :000,136,000,000,012,000,244
50790 :008,000,128,011,000,192,185
50796 :000,003,004,034,050,000,199
50802 :000,188,160,032,002,008,248
50808 :000,000,000,128,000,136,128
50814 :034,062,044,008,000,132,150
50820 :000,192,048,000,192,128,180
50826 :002,000,000,000,184,000,068
50832 :000,000,000,000,000,000,144
50838 :000,000,000,000,000,000,150
50844 :000,169,000,133,160,133,239
50850 :161,133,162,024,169,010,053
50856 :109,216,207,141,216,207,240
50862 :173,217,207,105,000,141,249
50868 :217,207,096,160,024,162,022
50874 :000,024,032,240,255,173,142
50880 :217,207,174,216,207,032,221
50886 :205,189,096,013,013,013,215
```



JACKPOT

Rick Rothstein

Now you can experience the thrill of slot machines without the danger of losing your money. These programs will show you how the bandits work and also how difficult it really is to hit a jackpot! Versions for TI-99/4A with Extended BASIC, Commodore 64, VIC-20, Atari, and IBM PC/PCjr (Color/Graphics Monitor Adapter required on PC).

Have you ever been to a casino in Las Vegas or Atlantic City? If so, your first visit probably left you dumbstruck over the sheer number of slot machines waiting to take your money. These nefarious one-arm bandits dazzle you with bright lights and promises of instant wealth.

A recent trip to Atlantic City—and an unprofitable encounter with some of these machines—prompted me to write "Jackpot." The program features three very different playing levels. Level one offers true casino odds; level two offers very generous odds which gives the player roughly the same odds that a casino normally enjoys; and level three will, in the long run, make you the owner of the casino.

Frustrating Experiences

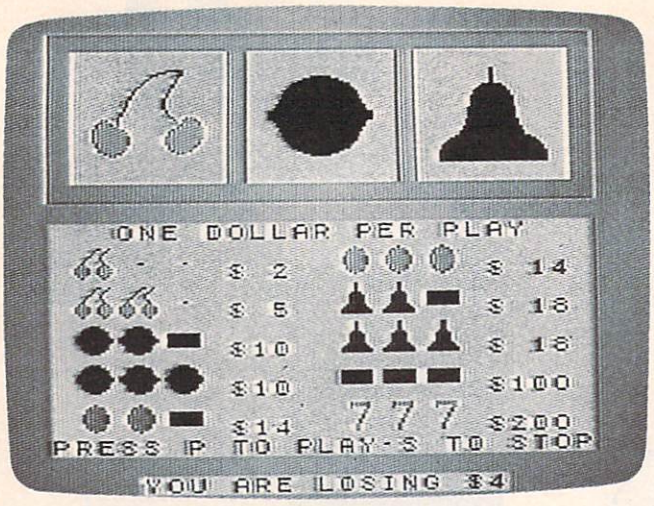
After you experience the frustrations of playing against the legitimate odds of level one, level two should give you a small measure of satisfaction if you play it long enough. Level three was included for you to play *after* you discover that level two, although tilted in your favor, is not overly generous.

Colorful graphics are used to display a payout chart, your current monetary status, and three large windows through which cherries, limes, plums, bells, bars, or lucky sevens will show. The shape displayed in each window is picked at random from 20-position wheels containing the above six shapes scattered randomly around them. The number of times each shape appears on each wheel was selected to produce the desired odds for each level of play.

Payout?

Before play begins in the TI version, the number 1 is displayed in each window, and the prompt ENTER LEVEL appears under the payout chart. If you press the space bar, the displayed level number will change. Press ENTER to begin the game at the displayed level. The payout chart continually prompts you to press the letter P to play and S to stop (the game). In addition, pressing AID (FCTN-7) will allow you to enter a new level of play, and pressing REDO (FCTN-8) will reset your money status to even while retaining the same level of play.

This program is written in Extended BASIC,



A winning combination on the TI version of "Jackpot."

and because it uses both upper- and lowercase letters, it can only be typed into a 99/4A console. However, once the program is recorded on tape or disk it will load and run properly on the older 99/4 console.

In order to facilitate use of the automatic NUMBERing command built into the 99/4A, the line numbers for the program logic begin at line 100 and increase in increments of ten. (Except for the introductory REMarks, all other REMark statements have line numbers ending in five and may be omitted.)

A Character Of Its Own

One of the strongest features of the 99/4A is its ability to display high-resolution graphics and up to 16 colors simultaneously. This program makes excellent use of these features by using seven *different* colors and redefining all 112 Extended BASIC characters which make up the highly detailed displays.

Although the program logic and mathematical theory of slot machines will not be explained, here are some of the programming techniques used in the TI version:

Line 120 sets the foreground and background colors of character set 0, which contains the cursor symbol and the edge character, to the same color and then fills the screen with the cursor symbol. Although the characters in this set cannot be redefined, turning the foreground and background to the same color has the same effect as redefining them to solid blocks of color. Filling the screen with one of these characters produces a solid background color which is independent of any other character—something the blank character cannot do.

After all of the characters have been redefined, they are combined into strings and placed on the screen with the DISPLAY AT command of Extended BASIC. This is a much faster way to place graphics on the screen than using the CALL HCHAR or CALL VCHAR subprograms.

The first statement in line 170 uses a random number, from the sequence that RANDOMIZE generates, for each loop in which either no key or an unrecognized key is pressed. This technique insures that the sequence of plays will not be repeated, since the time period between recognized keypresses will vary from play to play and from person to person.

Sluggish Sprites

Most programmers who work in Extended BASIC think sprites are useful only when they move. Actually, they can be very handy if placed on the screen and left stationary. In this program, one sprite, doubled in size by the CALL MAGNIFY (2) subprogram, is placed in front of each window. They serve as level-of-play indicators and are left transparent during game play. When needed, a simple CALL COLOR makes them visible. The advantage of using sprites in this particular application is that characters (numbers in this case) defined in an area measuring two characters by two characters are displayed with no additional character redefinitions. (Remember, all 112 Extended BASIC characters were redefined and used for the display graphics.) Without sprites, 12 additional character redefinitions would have been necessary to create the three large-sized numbers needed for the level-of-play indicator.

If you wish to save the time and effort of typing this program in, I will be glad to make a copy for you (TI version *only*). Just send $3, a blank cassette or disk, and a self-addressed, stamped mailer to:

Rick Rothstein
P.O. Box 4169
Trenton, NJ 08610

Program 1: TI-99/4A Jackpot

```
99 REM EXTENDED BASIC REQUIRED
100 CALL CLEAR :: CALL SCREEN(12)::
     CALL COLOR(0,12,12):: CALL HCH
    AR(1,1,30,768)
110 CALL COLOR(1,5,16,2,7,16,3,2,16
    ,4,2,16,5,2,16,6,7,16,7,2,16,8,
    7,16)
120 CALL COLOR(9,13,16,10,14,16,11,
    14,16,12,5,16,13,13,16,14,13,16
    )
130 RANDOMIZE :: LEVEL=49 :: TOTAL=
    0 :: OPTION BASE 1 :: DIM SHAPE
    $(6,5),WHEEL$( 3,3),PICK(3):: G
    OTO 310
135 REM ** P,S OR AID PRESSED **
140 RANDOM=RND :: CALL KEY(0,KEY,ST
    ATUS):: IF STATUS=0 THEN 140
```

```
150 IF KEY=83 OR KEY=115 THEN CALL
    CLEAR :: CALL COLOR(1,1,1):: EN
    D ELSE IF KEY= 80 OR KEY=112 TH
    EN TOTAL=TOTAL-1 :: GOTO 200
160 IF KEY=6 THEN TOTAL=0 :: GOSUB
    810 :: GOTO 140 ELSE IF KEY<>1
    THEN 140
170 GOSUB 770 :: CALL COLOR(#1,2,#2
    ,2,#3,2):: DISPLAY AT(24,1)BEEP
    :RPT$(CHR$(30),8)&"ECTERwFEVEF"
    &RPT$(CHR$(30),9)
175 REM ** CHANGE LEVEL **
180 CALL KEY(0,KEY,STATUS):: IF STA
    TUS<1 THEN 180 ELSE IF KEY=13 T
    HEN GOSUB 780 :: GOTO 140 ELSE
    IF KEY<>32 THEN 180
190 LEVEL=LEVEL+1+3*(LEVEL>50):: DI
    SPLAY AT(1,2)SIZE(1)BEEP:"K" ::
     CALL PATTERN(#1,LEVEL,#2,LEVEL
    ,#3,LEVEL):: GOTO 180
195 REM ** PICK 3 SHAPES **
200 CALL SOUND(50,-2,0):: GOSUB 810
     :: GOSUB 770 :: FOR I=1 TO 3 :
    : PICK(I)=VAL(SEG$(WHEEL$(LEVEL
    -48,I),INT(20*RND+1),1)):: NEXT
     I
205 REM ** DISPLAY SHAPES **
210 FOR I=4 TO 20 STEP 8 :: FOR J=3
     TO 7 :: DISPLAY AT(J,I)SIZE(5)
    :SHAPE$(PICK((I+4)/8),J-2):: NE
    XT J
220 CALL SOUND(50,-6,0):: NEXT I ::
     CALL SOUND(100,44000,30)
225 REM ** CHECK FOR WIN **
```



```
230 IF PICK(1)=1 THEN IF PICK(2)=1
    THEN COINS=5 :: GOTO 270 ELSE C
    OINS=2 :: GOTO 270
240 IF PICK(1)<>PICK(2)THEN 140 ELS
    E IF ((PICK(2)<>PICK(3)OR PICK(
    2)=5)AND PICK( 3)<>5)OR(PICK(2)
    =6 AND PICK(3)=5)THEN 140
250 IF PICK(1)<5 THEN COINS=2+4*PIC
    K(1):: GOTO 270
260 IF PICK(1)=5 THEN COINS=100 ::
    GOTO 280 ELSE COINS=200 :: GOTO
     290
265 REM ** UPDATE MONEY STATUS **
270 FOR I=1 TO COINS :: TOTAL=TOTAL
    +1 :: GOSUB 810 :: CALL SOUND(2
    5,1000,0,3250,0,6750,0):: NEXT
    I :: GOTO 140
280 FOR I=5 TO COINS STEP 5 :: TOTA
    L=TOTAL+5 :: GOSUB 810 :: CALL
    SOUND(35,1000, 0,3250,0,6750,0)
    :: NEXT I :: GOTO 140
290 FOR I=40 TO COINS STEP 40 :: FO
    R SIREN=700 TO 900 STEP 10 :: C
    ALL SOUND(-99,SIREN,0):: NEXT S
    IREN :: TOTAL=TOTAL+40 :: GOSUB
     810
300 FOR SIREN=900 TO 700 STEP -20 :
    : CALL SOUND(-200,SIREN,0):: NE
    XT SIREN :: NEXT I :: GOTO 140
305 REM ** DEFINE GRAPHICS **
310 CALL CHAR(33,"181818181818FF0
    1030707 0F0F0F0F80C0E0E0F0F0F0F"
    ,36,"0F1F1F1F1F1F3F3FF0F8F8F8F8
    F8FCFC0000000000000030F1F3F7F7FFFF
    FFFFFFF")
```

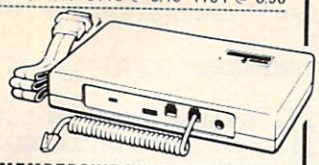
```
320 CALL CHAR(32,"FFFFFFFFFFFFFFFF"
    ,40,"0000000000001C3E7F0000000078
    FCFEFE7FFFFFFFFFFFFF7FC0F0F8F8F8F
    CFCFCFC")
330 CALL CHAR(44,"030F1F1F3F3F3F3FFF
    FFFFFFFFFFFFEFE7F3F3F0F0F03000000
    F8F8F0E08")
340 CALL CHAR(48,"0038444444443800000
    1030101010380000384408102070C00
    00384418044438")
350 CALL CHAR(52,"0008182848 7C08000
    07840780444380000384078444443800
    007C040810202")
360 CALL CHAR(56,"0038443844443800000
    3844443C0478000000000000003F3F3F
    000000000000FCFCFC")
370 CALL CHAR(60,"3F3F3F000000000000F
    CFCFC0000000000000385430185438000
    FFFFFFFFFFFFFFFF")
380 CALL CHAR(64,"0038101010103800000
    3844447C4444000003C223C22223C00
    0044464544C44")
390 CALL CHAR(68,"0078444444447800000
    7C4078404070C0000404040404070C00
    0038444404C4438")
400 CALL CHAR(72,"1F1F0F03000000000F
    CF8F0C0000000000183C7E7E7E3C1800
    FFFFFFFFFFFFFFFF")
410 CALL CHAR(76,"0000070704000000000
    000F8F8183030600000000010101000006
    0C0C080808")
420 CALL CHAR(80,"0078444478404000000
    0444428101010000078444478484400
    0038443008443 8")
430 CALL CHAR(84,"007C101010101000000
    0444444444438000004444444428100000
    0044444454542 8")
440 CALL CHAR(88,"FFFFFFFFFFFFF8F8F
    FFFFFFFFFFF0000FFFFFFFFFFFFFEFC
    0101030707 0F1F 1F")
450 CALL CHAR(92,"FCF8F0E0C0C0808 03
    F3F3F3E7E7E7E7E7C"&RPT$("FC",15
    ),96,"FF7F1F07000000000FFFFFFFFF
    F000000FFFEF8E000000000FFFFFFFF
    FFFFFFFF")
460 CALL CHAR(100,"000003 0F1F1F3F7F
    0000C0F0F8F8FCFE7F3F1F1F0F03000
    0FEFCF8F8F0C")
470 CALL CHAR(104,"00030F1F3F7FFFFF
    00C0F0F8FCFEFFFF0101010303070707
    78080 80C0C0E0E0E")
480 CALL CHAR(108,"070F0F0F0F0F0F07
    E0F0F0F0F0F0F0E0070707030301010
    1E0E0E0C0C0808 08")
490 CALL CHAR(112,"FFFF7F3F1F0F03 00
    FFFFFEFCF8F0C000FFFFFFFFFFFFFFFF
    F000001070F0F1F1F")
500 CALL CHAR(116,"0000 80E0F0F0F8F8
    1F1F0F0F0F0701 0000F8F8F0F0E0800000
    00")
510 CALL CHAR(120,"FCFEFEFFFFFFFFFF
    0000000000C0F0F8000010101010307 0
    7000808080 80C0E0E")
520 CALL CHAR(124,"07070F1F3F3F0000
    E0E0F0F8FCFC0000C000000000000000
    00000003")
530 CALL CHAR(128,"0001060810204080
    3FC1030204040808010202040408080
    810102020404040 4")
540 CALL CHAR(132,"402020100804020 1
    0000000001020408000064881020201
    0000000000071F7FFF")
550 CALL CHAR(136,"000000FFFFFFFFFF
    0000000E0F8FEFF0103030707 0F0F1
    F80C0C0E0E0F0F0F8")
560 CALL CHAR(140,"1F3FFFFFFFF3F1F
    F8FCFFFFFFFFFCF81F0F0F0707 03030
    1F8F0F0E0E0C0C08")
565 REM ** CREATE SHAPES **
570 SHAPE$(1,1)="ww"&CHR$(128)&CHR$
    (129)&"w" :: SHAPE$(1,2)="w"&CH
    R$(130)&"w"&CHR$(131)&"w" :: SH
    APE$(1,3)="("&CHR$(131)&"w"&CHR
    $(132)&")"
580 SHAPE$(1,4)="*+w,-" :: SHAPE$(1
    ,5)="./wHI" :: SHAPE$(2,1)="w"&
    CHR$(135)&CHR$(136)&CHR$(137)&"
    w"
590 SHAPE$(2,2)=CHR$(138)&"ccc"&CHR
    $(139):: SHAPE$(2,3)=CHR$(140)&
    "ccc"&CHR$(141):: SHAPE$(2,4)=C
    HR$(142)&"ccc"&CHR$(143)
600 SHAPE$(2,5)="w'abw" :: SHAPE$(3
    ,1)="whriw" :: SHAPE$(3,2)="jrr
    rk" :: SHAPE$(3,3)="lrrrm" :: S
    HAPE$(3,4)="nrrro" :: SHAPE$(3,
    5)="wprqw"
610 SHAPE$(4,1)="ww!ww" :: SHAPE$(4
    ,2)="w"" #w" :: SHAPE$(4,3)="w$
     %w" :: SHAPE$(4,4)="&' xy" ::
    SHAPE$(4,5)="'{3 SPACES}x"
620 SHAPE$(5,1)="wwwww" :: SHAPE$(5
    ,2)="?????" :: SHAPE$(5,3)="?BA
    R?" :: SHAPE$(5,4)="?????" :: S
    HAPE$(5,5)="wwwww"
630 SHAPE$(6,1)="wXYZw" :: SHAPE$(6
    ,2)="ww[\w" :: SHAPE$(6,3)="ww]
    ww" :: SHAPE$(6,4)="ww^ww" :: S
    HAPE$(6,5)="ww_ww"

635 REM ** DISPLAY GRAPHICS **
640 DISPLAY AT(1,2)SIZE(25):RPT$("K
    ",25):: GOSUB 770 :: DISPLAY AT
    (9,2)SIZE(25):RPT$("K",25)
650 DISPLAY AT(11,2)SIZE(25):"www0C
    EwD0FFARwPERwPFAQwww"
660 DISPLAY AT(12,2)SIZE(25):"w"&CH
    R$(133)&CHR$(134)&"wwwwwwwwwwst
    ststwwwwww" :: DISPLAY AT(13,2)
    SIZE(25):"wJJw{,}w{,}w>w2wwuvuv
    uvw>w14w"
670 DISPLAY AT(14,2)SIZE(25):"w"&CH
    R$(133)&CHR$(134)&CHR$(133)&CHR
    $(134)&"wwwwwwwwz{z{:;wwwwww"
680 DISPLAY AT(15,2)SIZE(25):"wJJJJ
    w{,}w>w5ww!{,}!{,}<=w>w18w" ::
    DISPLAY AT(16,2)SIZE(25):"wdede
    :;wwwwwwz{z{z{wwwwww"
690 DISPLAY AT(17,2)SIZE(25):"wfgfg
    <=w>10ww!{,}!{,}!{,}w>w18w" ::
    DISPLAY AT(18,2)SIZE(25):"wdede
    dewwwwww:;:;:;wwwwww"
700 DISPLAY AT(19,2)SIZE(25):"wfgfg
    fgw>10ww<=<=<=w>100w" :: DISPLA
    Y AT(20,2)SIZE(25):"wstst:;wwww
    wwLMLMLMwwwwww"
710 DISPLAY AT(21,2)SIZE(25):"wuvuv
    <=w>14wwNONONOw>200w" :: DISPLA
    Y AT(22,2)SIZE(25):"PRESSwPwT0w
    PFAQ"&CHR$(127)&"SwT0wST0P"
720 CALL MAGNIFY(2):: CALL SPRITE(#
    1,LEVEL,1,29,53,#2,LEVEL,1,29,1
    17,#3,LEVEL,1,29,181)
725 REM ** PUT SHAPES ON WHEEL **
```

SPECIALS on INTEGRATED CIRCUITS

6502 @ 4.90 6520 @ 4.00 6522 @ 5.00 4116 @ 1.85
2532 @ 5.90 2716 @ 4.45 6116 @ 6.45 4164 @ 6.90

Anchor Automation Signalman MODEMS



FREE SOURCE MEMBERSHIP WITH SIGNALMAN

All Signalman Modems are Direct Connect, and provide the best price-performance values. **Dealer and OEM inquiries invited**

Volksmodem with computer cable	**68**
Mark VII Auto Dial/Auto Answer	99
Mark XII Smart Modem 1200/300	279
DC HAYES Smartmodem	**219**
DC Hayes Smartmodem 1200/300	519

PROM QUEEN for C64 or VIC	**130**
Apple Emulator for Commodore 64	**call**
STAT Statistics Package for C64	**95**
Solid Oak 2 Level Stand for C64 or VIC	**29**
C64/VIC Switch (networking)	119
BACKUP V1.0 tape copier for C64 or VIC	**20**
CARDBOARD/6 Motherboard - VIC	64
CARDBOARD/5 Motherboard - C64	56
CARD PRINT G Printer Int. with Graphics	72
CARDBOARD/3s Motherboard - VIC	22
CARDCO C64/VIC Calculator Keypad	32
CARDRAM/16 RAM Expansion - VIC	44
Complete CARDCO Line in stock	
CIE and VIE IEEE Interfaces in stock	85
MSD Super Drive for C64 or IEEE	**365**
MAE Assembler for C64	50
Koala Pad Touch Tablet—C64 or VIC	**75**
CBC 4/12 Analog to Digital 4 chan/12 bit	**179**
MULTIPLAN for C64	**79**
Dust Cover for C64 or VIC	6
Grand Master Chess for C64	24
Musicalc Software for C64 in stock	
SM Software for C64 in stock	
BusCard II from Batteries Included	**149**
ULTRA BASIC – 64 with Turtle Graphics	37
Super Disk Utility – C64 – includes backup	19
Trackball (Electra Concepts) C64	29
MicroChess – C64 – 8 levels of play	17
HES MODEM with software for C64	**45**
Commodore 64 Programmers Reference Guide	16
WordPro 3+/64 with Spellright	**85**
VIController (also C64) – BSR Controller	50
COM VOICE Synthesizer for C64 or VIC	139
VIC products in stock – call for extra discounts.	
Victory Software for VIC and C64 in stock.	

APPLE—FRANKLIN ITEMS

FRANKLIN—complete line in stock	
QUENTIN Drives for Apple/Franklin	**189**
Swapper Stopper	**26**
automatic switch between paddles and joystick	
KRAFT Apple Joystick	**32**
Kraft Apple Paddle Pair	32
Koala Pad Touch Tablet-Apple/Franklin	**90**
SPINNAKER Software in stock	
Broderbund Software in stock	
16K RAM Card for Apple	59
Multiplan—Microsoft	**185**
Solid Oak 2 Level Stand for Apple	**29**
Serial Card for Apple	89
MPC RAM/80 column card for IIe (AP/TXT)	**139**
Z80 Softcard and CP/M (Microsoft)	235
RANA Elite I with Controller	389
Parallel Printer Interface/Cable	69
Microtek and MPC Interfaces in stock	
Grappler + Interface	135
DC Hayes Micromodem II, IIe with Smartcom	245
PFS: File or PFS: Report or PFS: Graph	**95**
Videx 80 Column Card	209
Apple Blue Book	**19**

commodore

See us for Personal, Business, and Educational requirements. Educational Discounts available.

PETSCAN I $245 base price

Allows you to connect up to 30 CBM/PET Computers to shared disk drives and printers. Completely transparent to the user. Perfect for schools or multiple word processing configurations. Base configuration supports 2 computers. Additional computer hookups $100 each.

COMPACK/STCP $115

Intelligent Terminal Package for PET, CBM, C64
Includes ACIA Hardware / STCP Software

SCREENMAKER 80 Column Adapter for C64 **139**
Provide big screen capability for business applications.

Copy-Writer Word Processor for C64 **49**
Full-featured package with 800 lines of text in memory. Includes double column printing, graphic capability, full printer support.

Special Screenmaker/Copy-Writer Combo **179**

PAPER CLIP Word Processor – CBM/C64	**60**
ORACLE (Consultant) Data Base by Batteries Included	**89**
SPINNAKER Software C64, Apple, IBM, Atari	
VICTORY Software for C64 in stock	
Compute! Books in Stock	
POWER ROM Utilities for PET/CBM	78
WordPro 4+ - 8032, disk, printer	285
VISICALC for PET, ATARI, or Apple	189
SM-KIT enhanced PET/CBM ROM Utilities	**40**
PET Spacemaker II ROM Switch	36
Dust Cover for PET, CBM, 4040, or 8050	8
CmC Interfaces (ADA1800, ADA1450, SADI in stock)	
HES Software and Hardware in stock	
UMI Products in stock	

FlexFile for PET/CBM/C64 $59

DataBase, Report Writer with calculations, Mailing Lists. Easy to use, and can be modified.

FORTH for PET/C64 full FIG model – Cargile/Riley **50**
includes all FORTH 79 Standard extensions, structured 6502 assembler with nested decision macros, standard 16x64 screens, ability to read/write BASIC sequential files, sample programs, introductory + reference manual.

Metacompliler for FORTH for independent object code **30**

Floating Point for FORTH **20**

BRIDGE 2.2 for C64 **$25**
Excellent Bridge program. Computer bids and plays against you. Hands are computer generated, but you may set up hands and modify contract. Allows you to claim balance of tricks, replay hand, and review previous trick.

KMMM PASCAL IV for PET/CBM/C64 **99**
Virtually full Jensen-Wirth implementation is now suitable for advanced placement courses.

EARL for PET/CBM/C64 Disk-based ASSEMBLER **65**

DISK I.C.U. (Intensive Care Unit) **$40**
COMPLETE DISK RECOVERY SYSTEM FOR 4040/1541
Edit disk blocks; duplicate disks, skipping over bad blocks; unscratch scratched files; check and correct scrambled files; recover improperly closed files. Includes complete diagnostic facilities, extensive treatment of relative files, optional output to printer, and comprehesive user manual (an excellent tutorial on disk operation and theory). Furnished on copy-protected disk. Backup available $10.

Smart Terminal Software for C64/VIC **10**

CBM Public Domain Software—C64/PET 27 disks **75**

DITTO DISK 64 **36**
Disk copy utility for C64. Make your own disk copies, even if the original is copy protected.

STAT for PET/CBM/C64 and Apple **95**
Comprehensive Statistical Analysis Routines
Includes complete file handling capabilities, summary statistics, confidence intervals, hypothesis tests, exponential mean tests, multiple and power series regression, analysis of variance, histograms, and non-parametric tests.

PageMate 60 Command Word Processor **20**
Full-featured package for all Commodore computers. Full screen editing, and supports disk, tape, and all printers.

DISK SPECIALS

Scotch (3M) 5" ss/dd	10/ 2.10	50/ 1.90	100/ 1.86
Scotch (3M) 5" ds/dd	10/ 2.65	50/ 2.45	100/ 2.40
Scotch (3M) 8" ss/sd	10/ 2.20	50/ 2.00	100/ 1.98
Scotch (3M) 8" ss/dd	10/ 2.80	50/ 2.50	100/ 2.47

We stock VERBATIM DISKS

Write for Dealer and OEM prices.

Sentinal 5" ss/dd	10/ 1.80	50/ 1.75	100/ 1.65
Sentinal 5" ds/dd	10/ 2.40	50/ 2.35	100/ 2.25

We stock Dysan disks

Wabash 5" ss/sd	10/ 1.45	50/ 1.40	100/ 1.35
Wabash 5" ss/dd	10/ 1.60	50/ 1.55	100/ 1.50
Wabash 5" ds/dd	10/ 1.95	50/ 1.90	100/ 1.80

We stock MAXELL DISKS

Write for dealer and OEM prices.

Disk Storage Pages 10 for $4 Hub Rings 50 for $6
Disk Library Cases 8"—3.00 5"—2.25
Head Disk Cleaning Kits 12
AMARAY Disk Storage Systems in stock.
Innovative Concepts FLIP 'N' FILES in stock.

CASSETTE TAPES—AGFA PE-611 PREMIUM

C-10	10/ .61	50/ .58	100/ .50
C-30	10/ .85	50/ .82	100/ .70

ZENITH data systems

ZVM-122A	99	ZVM-123G	89
ZVM-131	300	ZVM-135	490
Z100 16-bit/8-bit System			**CALL**
Z29 Terminal (DEC and ADM compatible)			680

Z-150 IBM PC COMPATIBLE **CALL**
Z-160 PORTABLE PC **CALL**
We stock entire Zenith line.

USI Video Monitors – Green or AMBER 20 MHz hi-res
Dealer and OEM inquiries invited

WRITE FOR IBM PC COMPATIBLE PRICES	
MultiPlan—IBM or Apple	**179**
Quadboard for IBM available	
KOALA PAD Touch Tablets—Apple, Atari, IBM, CBM	
Peachtext 5000 Software Package	**199**
PFS Software for IBM and Apple in stock	
SPINNAKER Software C64/VIC, Apple, IBM, Atari	
VOTRAX Personal Speech System	269
BMC 9191+ Color Monitor (plus model)	**245**
BMC 12A 12" Green Monitor	79
Dynax (Brother) DX-15 Daisy Wheel Printer	**459**
Brother HR-25 Daisy Wheel Printer (25 cps)	**749**
Itoh Prowriter Parallel Printer	379
Panasonic 1090 Printer with Correspondence Mode	**279**
Gemini 10X	**289**
EPSON, Okidata, Star Micronics printers in stock	
USI CompuMOD 4 R F Modulator	29
We Stock AMDEK Monitors	
A P Products	15% OFF
COMPUTER COVERUPS IN STOCK	
BROOKS 6 Outlet Surge Suppressor/Noise Filter	**54**
Surge Suppressor-6 outlet	29
Electrohome 1302-2 13" Hi-res RGB Monitor	335
Panasonic 12" Monitor (20 MHz) with audio	137
Synertek SYM-1 Microcomputer	**189**

Hewlett Packard

Write or call for prices.

DATASHIELD BACKUP POWER SOURCE **$265**
Battery back up Uninterruptible Power Supply with surge and noise filtering. The answer to your power problems.

ATARI – WE STOCK ENTIRE LINE
SPINNAKER and Broderbund Software in stock.

215-822-7727
252 Bethlehem Pike
Colmar, PA 18915

A B Computers

WRITE FOR CATALOG. Add $1.50 per order for United Parcel. We pay balance of UPS surface shipping charges on all prepaid orders (add extra for mail, APO/FPO, air). Prices include cash discount. Regular prices slightly higher. Prices subject to change.

```
730 FOR I=1 TO 3 :: FOR J=1 TO 3 ::
    READ ORDER$ :: WHEEL$(I,J)=ORD
    ER$ :: NEXT J :: NEXT I :: KEY=
    1 :: GOTO 160
735 REM ** ORDER OF SHAPES **
740 DATA 253123642453142532 34,14216
    3131564252131 32,234243254243642
    34324
750 DATA 126531241351243152 46,62543
    512136423146352,243524635234235
    42364
760 DATA 521346461215315362 41,56231
    534146213125645,234562463562543
    52634
765 REM ** CLEAR WINDOWS **
770 FOR I=2 TO 8 :: DISPLAY AT(I,2)
    SIZE(25):"Kwwwwwwwwwwwwwwwwwwww
    wwwwK" :: NEXT I :: RETURN
775 REM ** INITIAL WINDOW SHAPES **
780 FOR I=1 TO 3 :: PICK(I)=VAL(SEG
    $(WHEEL$(LEVEL-48,I),INT(20*RND
    +1),1)):: NEXT I
790 CALL COLOR(#1,1,#2,1,#3,1):: TO
    TAL=0 :: FOR I=4 TO 20 STEP 8 :
    : FOR J=3 TO 7
800 DISPLAY AT(J,I)SIZE(5):SHAPE$(P
    ICK((I+4)/8),J-2):: NEXT J :: C
    ALL SOUND(35,-6,0):: NEXT I ::
    CALL SOUND(100,44000,30)
805 REM ** DISPLAY MONEY STATUS **
810 IF TOTAL=0 THEN DISPLAY AT(24,1
    ):RPT$(CHR$(30),5)&"Q0UwAREwC0W
    wEVEC"&RPT$(CHR$(30),7):: RETUR
    N
820 TOTAL$=STR$(ABS(TOTAL)):: LENGT
    H=LEN(TOTAL$):: COLUMN=6+(TOTAL
    >0)-INT(.5+LENGTH/2)
830 IF TOTAL>0 THEN DISPLAY AT(24,C
    OLUMN)SIZE(20+LENGTH):CHR$(30)&
    "Q0UwAREwW0CC0CGw>"&TOTAL$&RPT$
    (CHR$(30),4):: RETURN
840 IF TOTAL<0 THEN DISPLAY AT(24,C
    OLUMN)SIZE(18+LENGTH):CHR$(30)&
    "Q0UwAREwF0S0CGw>"&TOTAL$&RPT$(
    CHR$(30),4):: RETURN
```

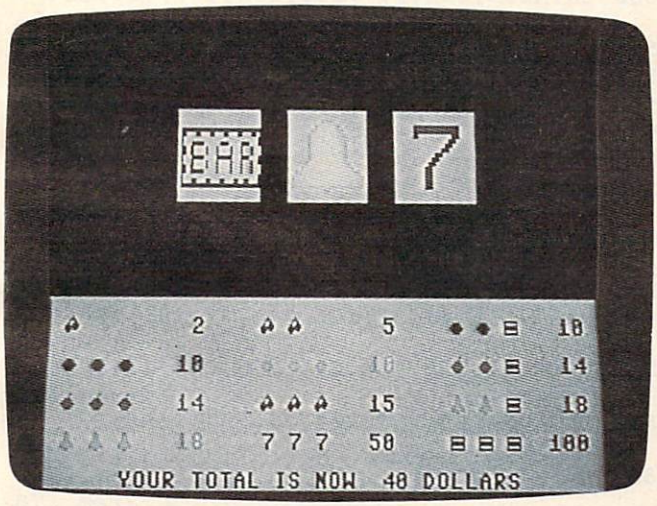
Program 2: VIC Jackpot

by Kevin Mykytyn, Editorial Programmer

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

```
10 POKE52,28:POKE56,28:POKE51,0:POKE55,0:
   GOSUB300                              :rem 158
20 PRINT"{CLR}";:FORA=1TO4:FORB=1TO22:PRI
   NT"@";:NEXT:NEXT                      :rem 248
30 FORA=1TO2:PRINT"@@@@@@{2 SPACES}@@
   {2 SPACES}@@{2 SPACES}@@@@@@";:NEXT
                                         :rem 141
40 FORA=1TO3:FORB=1TO22:PRINT"@";:NEXT:NE
   XT                                    :rem 32
50 PRINTB$V$B$V$B$" 100 "V$B$V$B$"
   {4 SPACES}50":PRINTS$V$S$V$S$"
   {2 SPACES}50 "V$S$V$S$"{4 SPACES}25"
                                         :rem 172
70 PRINTBE$V$BE$V$BE$"{2 SPACES}18 "V$BE$
   V$BE$V$B$"{2 SPACES}10"               :rem 101
80 PRINTC$V$C$V$C$"{2 SPACES}15 "V$C$V$C$
   "{5 SPACES}5 ":PRINTV$P$V$P$V$P$"
   {2 SPACES}14 "V$P$V$P$V$B$"{2 SPACES}1
   0"                                    :rem 158
95 PRINTL$V$L$V$L$"{2 SPACES}10 "V$C$"
   {7 SPACES}2"                          :rem 59
100 A=RND(1):A$="":GETA$:IFA$<>"P"ANDA$<>
    "E"THEN100                           :rem 75
110 IFA$="E"THENPRINT"{CLR}":POKE36869,24
    0:END                                :rem 43
115 T$="{DOWN}{2 SPACES}{UP}{2 LEFT}
    {2 SPACES}{DOWN}":GOSUB210:GOSUB220:G
    OSUB220:PRINTY$"{UP}{21 SPACES}"
                                         :rem 127
120 W=0:H=0:N=1:GOSUB200:GOSUB210:GOSUB26
    0:N=2:GOSUB200:GOSUB220:GOSUB260
                                         :rem 10
140 N=3:GOSUB200:GOSUB220:GOSUB260:rem 63
145 FORA=1TO24STEP2:H$=STR$(H):H$=RIGHT$(
    H$,(LEN(H$)-1))                      :rem 249
150 K=LEN(P$(A)):IFP$(A)=LEFT$(H$,K)THENW
    =VAL(P$(A+1))                        :rem 60
160 NEXT:IFW>0THENPRINTY$"{UP}{2 SPACES}Y
    OU WIN"W-1"DOLLARS":GOSUB280 :rem 187
170 TT=TT-1:IFTT>0THENTT$=STR$(TT)+"
    {2 SPACES}":PRINTY$"{4 SPACES}TOTAL N
    OW "TT$;:POKE198,0:GOTO100   :rem 145
180 PRINTY$"{UP}{4 SPACES}YOU ARE BROKE"
                                         :rem 192
190 PRINT"{3 SPACES}PLAY AGAIN{2 SPACES}Y
    /N ";                                :rem 18
195 GETA$:IFA$<>"Y"ANDA$<>"N"THEN195
                                         :rem 59
197 IFA$="Y"THENTT=50:GOTO20             :rem 189
198 PRINT"{CLR}":END                     :rem 23
200 A=INT(RND(1)*17)+1:B=G%(N,A):T$=F$(B)
    :H=H*10+B:RETURN                     :rem 214
210 PRINT"{HOME}{4 DOWN}{6 RIGHT}"T$;:RET
    URN                                  :rem 54
220 PRINT"{UP}{2 RIGHT}"T$;:RETURN
                                         :rem 253
260 POKEV,150:FORA=1TO30:NEXT:POKEV,0:IFN
    <3THENFORA=1TORND(1)*200:NEXT:rem 210
270 RETURN                               :rem 121
280 FORQ=1TOW:TT=TT+1:TT$=STR$(TT)+"
    {2 SPACES}":PRINTY$"{4 SPACES}TOTAL N
    OW "TT$;:POKEV1,220                  :rem 220
290 FORA=1TO110-W:NEXT:POKEV1,0:NEXT:RETU
    RN                                   :rem 66
300 PRINT"{CLR}{3 DOWN}{2 SPACES}LOADING
    {SPACE}CHARACTERS"                   :rem 8
305 DIMG%(3,17):FORA=1TO3:FORB=1TO17:READ
    C:G%(A,B)=C:NEXT:NEXT                :rem 41
310 DATA1,2,3,4,5,6,7,5,7,3,4,5,6,7,3,2,3
                                         :rem 231
320 DATA1,2,3,4,5,6,7,5,7,3,4,5,6,7,3,6,3
                                         :rem 236
330 DATA1,2,3,4,5,6,7,5,7,3,4,5,6,7,3,2,3
                                         :rem 233
340 DIMP$(24):FORA=1TO24:READP$(A):NEXT
                                         :rem 74
350 DATA4,3,44,6,444,16,555,11,661,11,666
    ,15,331,11,333,19,22,26,222,51,11,51,
    111,101                              :rem 103
400 A=7168:B=7679:C=25600:FORI=ATOB:POKEI
    ,PEEK(I+C):NEXT:POKE36869,255:rem 199
410 READB:IFB=-1THEN430                  :rem 95
420 FORI=0TO7:READC:POKE7168+B*8+I,C:NEXT
    :GOTO410                             :rem 24
430 B$="{RED}%&{DOWN}{2 LEFT}'(":S$="
    {RED}!-{DOWN}{2 LEFT}#$":L$="{GRN}↑←
    {DOWN}{2 LEFT}>?":C$="{RED}Z[{DOWN}
    {2 LEFT}:;":P$="{PUR}£]{DOWN}
    {2 LEFT}<="                          :rem 118
440 BE$="{YEL})*{DOWN}{2 LEFT}+,":LE$="
    {YEL}↑←{DOWN}{2 LEFT}>?":U$="{UP} ":V
```

```
    $="{UP}":Y$="{HOME}{22 DOWN}" :rem 54
450 F$(1)=B$:F$(2)=S$:F$(3)=BE$:F$(4)=C$:
    F$(5)=L$:F$(6)=P$:F$(7)=LE$  :rem 177
490 POKE36878,15:V=36877:V1=36876:TT=50:R
    ETURN                        :rem 244
500 DATA26,0,0,0,1,2,4,8,28      :rem 64
501 DATA27,0,0,128,128,128,64,56,124
                                 :rem 22
502 DATA28,0,0,0,3,15,31,63,63   :rem 218
503 DATA29,0,96,192,224,240,248,252,252
                                 :rem 182
504 DATA30,0,0,0,0,7,31,63,127   :rem 212
505 DATA31,0,0,0,0,224,248,252,254
                                 :rem 162
506 DATA33,0,0,15,15,0,0,0,0     :rem 103
507 DATA45,0,0,248,248,24,48,96,192
                                 :rem 248
508 DATA35,1,3,3,3,3,3,0,0       :rem 15
509 DATA36,128,0,0,0,0,0,0,0     :rem 108
510 DATA37,0,0,0,255,205,128,177,170
                                 :rem 15
511 DATA38,0,0,0,255,179,1,153,85:rem 128
512 DATA39,179,170,178,128,205,255,0,0
                                 :rem 133
513 DATA40,217,85,85,1,179,255,0,0
                                 :rem 185
514 DATA41,0,0,3,7,15,15,15,15   :rem 219
515 DATA42,0,0,128,192,224,224,224,224,43
    ,15,15,31,63,63,63,1,0       :rem 49
517 DATA44,224,224,240,248,248,248,128,0,
    58,62,127,127,127,127,62,28,0:rem 181
519 DATA59,254,254,254,254,124,56,0,0,60,
    63,63,63,63,31,15,3,0        :rem 24
521 DATA61,252,252,252,252,248,240,224,0,
    62,255,127,63,31,7,0,0,0     :rem 151
523 DATA63,255,254,252,248,224,0,0,0,0,25
    5,255,255,255,255,255,255,-1
                                 :rem 57
```



"Jackpot" for the 64 features smooth sprite movement to simulate realistic casino action.

Program 3: 64 Jackpot

by Kevin Mykytyn, Editorial Programmer

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

```
15 POKES+1,112:POKES+5,9:POKES+15,208:POK
   ES+24,15                      :rem 119
20 POKES+8,150:POKES+12,8:POKES+13,0
                                 :rem 94
25 PRINT"{CLR}";:FOR A=1TO5:FORB=1TO40:PR
   INT"{RVS} ";:NEXTB,A          :rem 6
30 FOR B=1TO5:PRINT"{RVS}{9 SPACES}{OFF}
   {6 SPACES}{RVS}{2 SPACES}{OFF}
   {6 SPACES}{RVS}{2 SPACES}{OFF}
   {6 SPACES}{RVS}{9 SPACES}";:NEXT
                                 :rem 143
35 FORA=1TO5:FORB=1TO40:PRINT"{RVS} ";:NE
   XTB,A:POKE53269,255           :rem 104
40 PRINT:PRINT"{RED} ({8 SPACES}2
   {4 SPACES}{RED}( ({6 SPACES}5
   {4 SPACES}{GRN}& & {RED}%{3 SPACES}10"
                                 :rem 31
45 PRINT:PRINT"{GRN} &{SHIFT-SPACE}&
   {SHIFT-SPACE}&{3 SPACES}10{4 SPACES}
   {YEL}&{SHIFT-SPACE}&{SHIFT-SPACE}&
   {3 SPACES}10{4 SPACES}{PUR}'
   {SHIFT-SPACE}' {RED}%{3 SPACES}14"
                                 :rem 19
50 PRINT:PRINT"{PUR} '{SHIFT-SPACE}'
   {SHIFT-SPACE}'{SHIFT-SPACE}{2 SPACES}1
   4{4 SPACES}{RED}({SHIFT-SPACE}(
   {SHIFT-SPACE}({3 SPACES}15{4 SPACES}
   {YEL}#{SHIFT-SPACE}#{SHIFT-SPACE}{RED}
   %{3 SPACES}18"                :rem 91
55 PRINT:PRINT"{YEL} #{SHIFT-SPACE}#
   {SHIFT-SPACE}#{3 SHIFT-SPACE}18
   {4 SPACES}{RED}${SHIFT-SPACE}$
   {SHIFT-SPACE}${3 SPACES}50{4 SPACES}
   {RED}%{SHIFT-SPACE}%{SHIFT-SPACE}%
   {2 SPACES}100":GOTO400        :rem 34
60 POKE 53281,1:POKE53275,255:D$="{HOME}
   {23 DOWN}"                    :rem 233
65 PRINT"{3 DOWN}{BLK}{RVS}{16 RIGHT}JACK
   POT"                          :rem 237
70 PRINT"{2 DOWN} YOU WILL BEGIN WITH $50
    AND TRY AND"                 :rem 94
75 PRINT"{DOWN}{8 SPACES}TURN IT INTO A F
   ORTUNE."                      :rem 0
80 PRINT"{DOWN} IT WILL COST YOU $1 FOR E
   ACH PULL."                    :rem 13
85 PRINT"{DOWN} TO PULL THE HANDLE USE TH
   E KEYS 1-4."                  :rem 134
90 PRINT"{DOWN} THE HIGHER THE NUMBER, TH
   E HARDER THE                  :rem 15
95 PRINT"{DOWN}{13 SPACES}PULL WILL BE."
                                 :rem 122
100 PRINT"{DOWN} TO STOP THE GAME AT ANY
    {SPACE}TIME PRESS (E)"       :rem 38
105 PRINT"{2 DOWN} PLEASE WAIT WHILE I LO
    AD THE SPRITES"              :rem 101
110 DIM WIN$(24)                 :rem 53
115 FOR A=1TO24:READWIN$(A):NEXT :rem 169
120 DATACHERRY,2,CHERRYCHERRY,5,LIMELIMEB
    AR,10,LIMELIMELIME,10        :rem 9
125 DATAPLUMPLUMBAR,14,PLUMPLUMPLUM,14,BE
    LLBELLBAR,18                 :rem 223
130 DATABELLBELLBELL,18,SEVENSEVENSEVEN,5
    0,BARBARBAR,100              :rem 72
135 DATALEMONLEMONLEMON,10,CHERRYCHERRYCH
    ERRY,15                      :rem 214
140 M=2047:NN=12288:OO=53248:S=54272:FORL
    =STOS+24:POKEL,.:NEXT        :rem 249
145 POKE S+5,9:POKES+6,0:POKES+24,15:POKE
    S+1,120                      :rem 16
150 FORI=49664TO49714:READB:B=B+239:POKEI
    ,B:NEXT                      :rem 177
155 DATA 4,2,3,7,5,6,7,6,3,1,5,5,7,4,6,7,
    6                            :rem 249
160 DATA 1,2,5,4,5,6,7,4,3,7,5,6,7,2,6,7,
    3                            :rem 241
```

```
165 DATA 1,2,3,4,5,6,7,7,3,4,5,6,7,5,6,7,
    4                                :rem 248
170 FORI=830TO833:POKEI,0:NEXT       :rem 104
175 AA=15360:BB=15807:CC=12568:DD=12615
                                     :rem 92
180 FOR A=AA TO BB:READB:IF B=>. THEN POK
    EA,B:GOTO 190                    :rem 13
185 D=ABS(B)-1:FORC=. TO D:POKE A+C,.:NEX
    T:A=A+D                          :rem 239
190 NEXT                             :rem 217
195 POKE56334,PEEK(56334)AND254:POKE1,PEE
    K(1)AND251:FOR I=. TO M          :rem 133
200 POKE NN+I,PEEK(OO+I):NEXT:POKE 1,PEEK
    (1)OR4:POKE56334,PEEK(56334)OR1
                                     :rem 39
205 PRINT"{CLR}":FOR I=CC TO DD:READB:POK
    E I,B:NEXT                       :rem 132
210 POKE53272,(PEEK(53272)AND240)OR12
                                     :rem 40
215 POKE53249,48:POKE53251,90:POKE53253,4
    8:POKE53255,90                   :rem 105
220 POKE53257,48:POKE53259,90        :rem 103
225 FORQ=2041TO2045STEP2:POKEQ,240:NEXT
                                     :rem 166
230 POKE53248,97:POKE53250,97:POKE53252,1
    60:POKE53254,160:POKE53256,223
                                     :rem 247
235 POKE53258,223:POKE53271,255:POKE53277
    ,255                             :rem 255
240 U=0:A=49152:B=49475:FORI=ATOB:READC::
    U=U+C:POKEI,C:NEXT:IFU=38200THENRETUR
    N                                :rem 11
245 PRINT "ERROR IN DATA STATEMENTS 645-8
    45!":END                         :rem 78
250 FORA=679TO685:POKEA,INT(RND(1)*16)+1:
    NEXT                             :rem 198
255 PRINTD$"{35 SPACES}":SYS 49152:POKE S
    +11,128                          :rem 35
260 SPIN$="":FORB=2041TO2045STEP2:Q=PEEK(
    B)-239                           :rem 5
265 ONQGOSUB365,370,375,380,385,390,395
                                     :rem 98
270 NEXT:WIN=0                       :rem 109
275 IFSPIN$="BARBARBAR"THENGOSUB 850
                                     :rem 215
280 FORA=1TO24:L=LEN(WIN$(A)):IFLEFT$(SPI
    N$,L)=WIN$(A)THENWIN=VAL(WIN$(A+1))
                                     :rem 126
285 NEXT:TT=TT-1                     :rem 3
290 IF WIN<>0THENPRINTD$"{10 SPACES}YOU W
    IN "WIN" DOLLARS";               :rem 147
295 IF WIN>5 THENPRINT"!";           :rem 236
300 IFWIN=0THEN325                   :rem 70
305 POKE S,80:POKES+1,112:POKES+15,208
                                     :rem 153
310 FORTT=TT+1TOTT+WIN-1:POKES+4,21:FORTD
    =1TO150-WIN:NEXT                 :rem 196
315 T$=STR$(TT):PRINTD$"{DOWN}{5 SPACES}Y
    OUR TOTAL IS NOW "T$" DOLLARS
    {2 SPACES}";                     :rem 83
320 POKES+4,20:FORT=1TO150:NEXT:NEXT:POKE
    S+1,0:POKES+15,0                 :rem 47
325 T$=STR$(TT):PRINTD$"{DOWN}{5 SPACES}Y
    OUR TOTAL IS NOW "T$" DOLLARS
    {2 SPACES}";                     :rem 84
330 IF TT>0 THEN 360                 :rem 3
335 PRINTD$"{8 SPACES}SORRY BUT YOU'RE BR
    OKE"                             :rem 140
340 PRINT"{3 SPACES}DO YOU WANT TO PLAY A
    GAIN?{2 SPACES}Y/N{4 SPACES}";
                                     :rem 187
345 GETA$:IFA$<>"Y"ANDA$<>"N"THEN345
                                     :rem 53
350 IF A$="Y"THEN 10                 :rem 246
355 POKE 53269,0:PRINT"{CLR}":END:rem 224
360 POKE 198,0:GOTO 400              :rem 205
365 SPIN$=SPIN$+"BAR":RETURN         :rem 245
370 SPIN$=SPIN$+"SEVEN":RETURN       :rem 157
375 SPIN$=SPIN$+"BELL":RETURN        :rem 64
380 SPIN$=SPIN$+"CHERRY":RETURN      :rem 234
385 SPIN$=SPIN$+"LIME":RETURN        :rem 73
390 SPIN$=SPIN$+"PLUM":RETURN        :rem 92
395 SPIN$=SPIN$+"LEMON":RETURN       :rem 158
400 GETA$:IFA$="E"THEN PRINT"{CLR}":POKE
    {SPACE}53269,0:END               :rem 63
405 A=RND(1)                         :rem 125
410 IF A$<"1" OR A$>"4" THEN 400     :rem 185
415 POKE 49238,VAL(A$):GOTO 250      :rem 162
420 DATA-12,255,255,255,204          :rem 90
425 DATA204,205,128,0,3,158,28,115
                                     :rem 179
430 DATA209,34,73,209,34,73,158,62
                                     :rem 195
435 DATA115,145,34,75,209,34,73,222
                                     :rem 240
440 DATA34,73,128,0,3,153,153,179:rem 136
445 DATA255,255,255,-22              :rem 160
450 DATA3,255,240,3,255,240,3        :rem 184
455 DATA0,112,0,0,224,0,1,192        :rem 165
460 DATA0,3,128,0,14,0,0,28          :rem 69
465 DATA0,0,56,0,0,112,0,0           :rem 12
470 DATA112,0,0,112,0,0,112,0        :rem 149
475 DATA0,112,-21,60                 :rem 248
480 DATA0,0,255,0,1,255,128,3        :rem 177
485 DATA255,192,3,255,192,3,255,192
                                     :rem 253
490 DATA3,255,192,3,255,192,3,255:rem 144
495 DATA192,7,255,224,15,255,240,63
                                     :rem 248
500 DATA255,252,127,255,254,127,255,254
                                     :rem 186
505 DATA127,255,254,0,24,-18,48      :rem 37
510 DATA0,0,80,0,0,136,0,1           :rem 7
515 DATA4,0,2,2,0,2,1,240            :rem 218
520 DATA15,129,248,31,195,252,63,227
                                     :rem 37
525 DATA252,63,227,252,063,225,248,63
                                     :rem 89
530 DATA224,240,31,192,0,15,128,-27
                                     :rem 219
535 DATA255,0,3,255,192,15           :rem 48
540 DATA255,240,31,255,248,63,255,252
                                     :rem 85
545 DATA127,255,254,255,255,255,127,255
                                     :rem 199
550 DATA254,63,255,252,31,255,248,15
                                     :rem 37
555 DATA255,240,3,255,192,0,255,-16,255
                                     :rem 178
560 DATA0,0,0,0,2,0,0,4              :rem 111
565 DATA0,0,8,0,0,60,0,0             :rem 172
570 DATA255,0,3,255,192,7,255,224:rem 140
575 DATA15,255,240,15,255,240,15,255
                                     :rem 33
580 DATA240,15,255,240,7,255,224,3
                                     :rem 183
585 DATA255,192,0,255,0,0,60,-16,255,-10
                                     :rem 208
590 DATA255,0,3,255,192,15           :rem 49
595 DATA255,240,31,255,248,63,255,252
                                     :rem 95
```

```
600 DATA127,255,254,255,255,255,127,255
                                    :rem 191
605 DATA254,63,255,252,31,255,248,15
                                     :rem 38
610 DATA255,240,3,255,192,0,255,-16,255
                                    :rem 170
615 DATA24,60,60,60,126,255,255,24
                                    :rem 185
620 DATA127,3,6,12,24,24,24,0     :rem 174
625 DATA0,255,129,255,255,129,255,0
                                    :rem 243
630 DATA0,0,60,126,255,126,60,0    :rem 17
635 DATA4,8,60,126,255,126,60,0    :rem 34
640 DATA8,20,38,111,255,246,96,0   :rem 87
645 DATA169,0,141,176,2,141,177,2:rem 138
650 DATA141,178,2,173,176,2,208,16
                                    :rem 189
655 DATA206,167,2,208,11,173,168,2
                                    :rem 190
660 DATA141,167,2,162,0,32,234,192
                                    :rem 179
665 DATA173,177,2,208,16,206,169,2
                                    :rem 198
670 DATA208,11,173,170,2,141,169,2
                                    :rem 180
675 DATA162,4,32,234,192,173,178,2
                                    :rem 196
680 DATA208,16,206,171,2,208,11,173
                                    :rem 231
685 DATA172,2,141,171,2,162,8,32   :rem 84
690 DATA234,192,238,173,2,208,188,238
                                    :rem 100
695 DATA61,3,173,61,3,201,2,208    :rem 32
700 DATA178,169,0,141,61,3,169,128
                                    :rem 192
705 DATA141,11,212,238,168,2,208,5
                                    :rem 180
710 DATA169,255,141,168,2,173,168,2
                                    :rem 245
715 DATA201,112,144,22,173,176,2,208
                                     :rem 20
720 DATA17,173,1,208,201,48,208,10
                                    :rem 175
725 DATA169,129,141,11,212,169,1,141
                                     :rem 28
730 DATA176,2,238,170,2,208,5,169:rem 142
735 DATA255,141,170,2,173,170,2,201
                                    :rem 225
740 DATA112,144,22,173,177,2,208,17
                                    :rem 232
745 DATA173,5,208,201,48,208,10,169
                                    :rem 242
750 DATA129,141,11,212,169,1,141,177
                                     :rem 25
755 DATA2,238,172,2,208,5,169,255:rem 149
760 DATA141,172,2,173,172,2,201,112
                                    :rem 219
765 DATA144,22,173,178,2,208,17,173
                                    :rem 247
770 DATA9,208,201,48,208,10,169,129
                                    :rem 245
775 DATA141,11,212,169,1,141,178,2
                                    :rem 183
780 DATA24,173,176,2,109,177,2,109
                                    :rem 195
785 DATA178,2,201,3,240,3,76,11    :rem 34
790 DATA192,96,160,2,254,1,208,189:rem 202
795 DATA1,208,201,130,208,62,169,194
                                     :rem 37
800 DATA133,252,152,72,138,72,74,141
                                     :rem 29
```

Notes For The Commodore And IBM Versions

The VIC-20 version of "Jackpot" (Program 2) plays the same as the TI version, but does not offer a choice of different levels at the beginning of the program. The faces of each of the three wheels are numerically represented in the DATA statements in lines 310–330. A 1 represents a bar, 2 is a seven, 3 a bell, 4 a cherry, 5 a lime, 6 a plum, and 7 a lemon. To change the odds, simply change the numbers in the DATA statements. For example, if you change all the numbers in the DATA statements to 1, you will always spin triple bar.

The Commodore 64 version (Program 3) is very different from the other games. Using a machine language subroutine and colored sprites, a smooth spinning effect is created. A total of six sprites are used (two for each window). The different shapes are displayed by changing the sprites' data pointers.

You can alter the odds, in the same way as the VIC version, by changing the numbers in the DATA statements in lines 155–165.

The IBM version of Jackpot (Program 4) uses the graphics PUT and GET commands to display various shapes on the screen. To run this program, therefore, requires either Cartridge BASIC (PCjr) or BASICA and the Color/Graphics Adapter (PC). The rules are the same as for the TI version, but the payoffs are slightly different. In the IBM version, you begin with $100. If you want to change the odds, change the numbers in the DATA statements in lines 1310–1330.

```
805 DATA80,3,74,170,189,65,193,133
                                    :rem 202
810 DATA251,254,62,3,189,62,3,201:rem 133
815 DATA17,208,5,169,0,157,62,3    :rem 43
820 DATA168,177,251,174,80,3,157,248
                                    :rem 48
825 DATA7,56,233,240,168,185,58,193:rem 4
830 DATA157,39,208,104,170,104,168,169
                                    :rem 141
835 DATA48,157,1,208,232,232,136,208
                                     :rem 36
840 DATA179,96,2,2,7,2,5,4          :rem 48
845 DATA7,0,17,34                   :rem 117
850 B=0:POKES+5,9:POKES+6,9          :rem 79
```

```
855 FORA=1TO130:POKE53281,A:POKE53280,256
    -A:B=-(B=0):POKE53271,255-255*B
                                    :rem 56
860 POKE 53277,255-255*B:POKES+1,A:POKES+
    4,33:FORTD=1TO20:POKES+4,32  :rem 220
865 FORTD=1TO20:NEXT:NEXT:POKES+4,32:POKE
    S+1,0:POKES-992,6:POKES-991,1:RETURN
                                    :rem 153
```

Program 4: PC/PCjr Jackpot

by Kevin Mykytyn, Editorial Programmer

```
10 DEFINT A-Z:SCREEN 1:KEY OFF
20 DEF SEG=0:POKE 1047,64
30 GOSUB 430:DIM A(E):A(0)=X:A(1)=Y:FOR
I=2 TO E:READ A(I):NEXT
40 GOSUB 430:DIM L(E):L(0)=X:L(1)=Y:FOR
I=2 TO E:READ L(I):NEXT
50 GOSUB 430:DIM B(E):B(0)=X:B(1)=Y:FOR
I=2 TO E:READ B(I):NEXT
60 GOSUB 430:DIM S(E):S(0)=X:S(1)=Y:FOR
I=2 TO E:READ S(I):NEXT
70 GOSUB 430:DIM CA(E):CA(0)=X:CA(1)=Y:F
OR I=2 TO E:READ CA(I):NEXT
80 GOSUB 430:DIM CH(E):CH(0)=X:CH(1)=Y:F
OR I=2 TO E:READ CH(I):NEXT
90 GOSUB 430:DIM BN(E):BN(0)=X:BN(1)=Y:F
OR I=2 TO E:READ BN(I):NEXT
100 COLOR 0,2:CLS:GOSUB 1300:T=100
110 LINE (76,20)-(112,50),2,B:LINE (142,
20)-(178,50),2,B:LINE (208,20)-(244,50),
2,B
120 PUT (10,75),S:PUT (50,75),S:PUT (90,
75),S:PUT (200,75),S:PUT (240,75),S
130 PUT (10,100),CH:PUT (50,100),CH:PUT
(90,100),CH:PUT (200,100),CH:PUT (240,10
0),CH
140 PUT (10,125),L:PUT (50,125),L:PUT (9
0,125),L:PUT (200,125),CH
150 PUT (10,150),A:PUT (50,150),A:PUT (9
0,150),A:PUT (200,150),A:PUT (240,150),A

160 PUT (10,175),CA:PUT (50,175),CA:PUT(
90,175),CA:PUT (200,175),CA:PUT (240,175
),CA
170 LOCATE 12,17:PRINT " 25":LOCATE 12,3
6:PRINT" 10"
180 LOCATE 15,17:PRINT " 15":LOCATE 15,3
6:PRINT "  5"
190 LOCATE 18,17:PRINT " 10":LOCATE 18,3
6:PRINT "  2"
200 LOCATE 21,17:PRINT " 18":LOCATE 21,3
6:PRINT " 10"
210 LOCATE 24,17:PRINT " 14";:LOCATE 24,
36:PRINT " 10";:GOSUB 420
220 LOCATE 8,5:PRINT "Press (P) to play
or (E) to end";
230 IF T<=0 THEN LOCATE 7,5:PRINT " Sorr
y,you are broke. Play again ? (y/n)":GOT
O 440
240 H=0:W=0:A$=INKEY$:A=RND(1):IF A$<>"E
" AND A$<>"P" THEN 240
250 IF A$="E" THEN CLS:END
260 X=79:Y=24:WH=1:GOSUB 320:X=145:WH=2:
GOSUB 320:X=211:WH=3:GOSUB 320
270 FOR A=1 TO 24 STEP 2:H$=STR$(H):H$=R
IGHT$(H$,(LEN(H$)-1))
280 L=LEN(P$(A)):IF P$(A)=LEFT$(H$,L) TH
EN W=VAL(P$(A+1))
290 NEXT:IF W>0 THEN GOSUB 400
300 T=T-1:GOSUB 420
310 POKE 1050,PEEK(1052):GOTO 230
320 FOR J=1 TO RND(1)*6+5:K=INT(RND(1)*1
7)+1:ON G(WH,K) GOSUB 330,340,350,360,37
0,380,390:SOUND 20*K+37,.1:FOR TD= 1 TO
J*40:NEXT:NEXT:H=H*10+G(WH,K):RETURN
330 PUT (X,Y),B,PSET:RETURN
340 PUT (X,Y),S,PSET:RETURN
350 PUT (X,Y),A,PSET:RETURN
360 PUT (X,Y),CH,PSET:RETURN
370 PUT (X,Y),L,PSET:RETURN
380 PUT (X,Y),CA,PSET:RETURN
390 PUT (X,Y),BN,PSET:RETURN
400 IF W=101 THEN PLAY SONG$:T=T+1:FOR A
=1 TO 25:T=T+4:GOSUB 420:NEXT:RETURN
410 FOR A=1 TO W:T=T+1:GOSUB 420:FOR B=1
531 TO 1540:SOUND B,.1:NEXT:NEXT:RETURN
420 LOCATE 1,5:PRINT "Winnings "T-100"
":LOCATE 1,25:PRINT "Total "T"  ":RETURN

430 READ X,Y:E=(4+INT((X+7)/8)*Y)/2:RETU
RN
440 A$=INKEY$:IF A$<>"Y" AND A$<>"N" THE
N 440
450 IF A$="Y" THEN LOCATE 7,5:PRINT "
        ONE MOMENT PLEASE            ":R
UN ELSE CLS:END
460 DATA &H40,&H17,&H0,&H1400,&H0,&H0,&H
0,&H500
470 DATA &H0,&H0,&H0,&H100,&H40,&H0,&H0,
&HA90A
480 DATA &HA06A,&H0,&H0,&HAA2A,&HA8AA,&H
0,&H0,&HAAAA
490 DATA &HAAAA,&H0,&H200,&HAAAA,&HAAAA,
&H80,&HA00,&HAAAA
500 DATA &HAAAA,&HA0,&H2A00,&HAAAA,&HAAA
A,&HA8,&H2A00,&HAAAA
510 DATA &HAAAA,&HA8,&HAA00,&HAAAA,&HAAA
A,&HAA,&HAA00,&HAAAA
520 DATA &HAAAA,&HAA,&HAA00,&HAAAA,&HAAA
A,&HAA,&HAA00,&HAAAA
530 DATA &HAAAA,&HAA,&HAA00,&HAAAA,&HAAA
A,&HAA,&H2A00,&HAAAA
540 DATA &HAAAA,&HA8,&H2A00,&HAAAA,&HAAA
A,&HA8,&HA00,&HAAAA
550 DATA &HAAAA,&HA0,&H200,&HAAAA,&HAAAA
,&H80,&H0,&HAAAA
560 DATA &HAAAA,&H0,&H0,&HAA2A,&HA8AA,&H
0,&H0,&HAA0A
570 DATA &HA0AA,&H0,&H0,&HA800,&H2A,&H0,
&H0
580 DATA &H40,&H17,&H0,&H0,&H0,&H0,&H0,&
H0
590 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0,&H5
501
600 DATA &H4055,&H0,&H0,&H5515,&H5455,&H
0,&H100,&H5555
610 DATA &H5555,&H40,&H1500,&H5555,&H555
5,&H54,&H5500,&H5555
620 DATA &H5555,&H55,&H5501,&H5555,&H555
5,&H4055,&H5505,&H5555
630 DATA &H5555,&H5055,&H5515,&H5555,&H5
555,&H5455,&H5555,&H5555
640 DATA &H5555,&H5555,&H5515,&H5555,&H5
555,&H5455,&H5505,&H5555
650 DATA &H5555,&H5055,&H5501,&H5555,&H5
555,&H4055,&H5500,&H5555
660 DATA &H5555,&H55,&H1500,&H5555,&H555
5,&H54,&H100,&H5555
670 DATA &H5555,&H40,&H0,&H5515,&H5455,&
H0,&H0,&H5501
```

```
680 DATA &H4055,&H0,&H0,&H0,&H0,&H0,&H0,
&H0
690 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0
700 DATA &H40,&H17,&HAAAA,&HAAAA,&HAAAA,
&HAAAA,&HAAAA,&HAAAA
710 DATA &HAAAA,&HAAAA,&HA2AA,&HAAA8,&H8
A2A,&HAAA2,&H80AA,&H28A0
720 DATA &H20A,&HAA80,&HA0,&H0,&H0,&HA00
,&HAAA2,&H2A80
730 DATA &HAA8,&HAAA,&HAAA2,&HAAA0,&HAAA
,&H8AAA,&H80A2,&HA0A0
740 DATA &HA0A,&H8A02,&H80A2,&HA0A0,&HA0
A,&H8A02,&H80A2,&HA0A0
750 DATA &HA0A,&H8A02,&HAAA2,&HAA80,&HAA
A,&HAAA,&HAAA2,&HAA80
760 DATA &HAAA,&H8AAA,&H80A2,&HA0A0,&HA0
A,&H8A02,&H80A2,&HA0A0
770 DATA &HA0A,&H8A02,&H80A2,&HA0A0,&HA0
A,&H8A02,&H80A2,&HA0A0
780 DATA &HA0A,&H8A02,&H82A2,&HA0A0,&HA0
A,&H8A02,&HAAA2,&HA080
790 DATA &HA0A,&H8A02,&HA0,&H0,&H0,&HA00
,&H80AA,&H28A0
800 DATA &H20A,&HAA80,&HA2AA,&HAAA8,&H8A
2A,&HAAA2,&HAAAA,&HAAAA
810 DATA &HAAAA,&HAAAA,&HAAAA,&HAAAA,&HA
AAA,&HAAAA,&H0
820 DATA &H40,&H17,&H0,&HAAAA,&HA8AA,&H0
,&H0,&HAAAA
830 DATA &HA8AA,&H0,&H0,&H0,&HA800,&H0,&
H0,&H0
840 DATA &HA002,&H0,&H0,&H0,&H800A,&H0,&
H0,&H0
850 DATA &H2A,&H0,&H0,&H0,&HA8,&H0,&H0,&
H200
860 DATA &HA0,&H0,&H0,&HA00,&H80,&H0,&H0
,&H2A00
870 DATA &H0,&H0,&H0,&HA800,&H0,&H0,&H0,
&HA002
880 DATA &H0,&H0,&H0,&HA002,&H0,&H0,&H0,
&HA002
890 DATA &H0,&H0,&H0,&HA002,&H0,&H0,&H0,
&HA002
900 DATA &H0,&H0,&H0,&HA002,&H0,&H0,&H0,
&HA002
910 DATA &H0,&H0,&H0,&HA002,&H0,&H0,&H0,
&HA002
920 DATA &H0,&H0,&H0,&HA002,&H0,&H0,&H0,
&HA002
930 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0
940 DATA &H40,&H17,&H0,&H0,&H0,&H0,&H0,&
H0
950 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0,&H0

960 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0,&H0

970 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0,&H0
980 DATA &H0,&H100,&H0,&HFF00,&HFFFF,&H1
FC,&H0,&HFFFF
990 DATA &HFFFF,&H4FF,&HFF00,&HFFFF,&HFF
FF,&HD5FF,&HFFFF,&HFFFF
1000 DATA &HFFFF,&HD4FF,&HFF00,&HFFFF,&H
FFFF,&HD5FF,&H0,&HFFFF
1010 DATA &HFFFF,&H4FF,&H0,&HFF00,&HFFFF
,&H1FC,&H0,&H0
1020 DATA &H0,&H100,&H0,&H0,&H0,&H0,&H0,
&H0
1030 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0,&H
0
1040 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0,&H
0
1050 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0
1060 DATA &H40,&H17,&H0,&H0,&H0,&H0,&H0,
&H0
1070 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0,&H
0
1080 DATA &H40,&H0,&H0,&H0,&H40,&H0,&H0,
&H0
1090 DATA &H40,&H0,&H0,&H100,&H10,&H0,&H
0,&H400
1100 DATA &H4,&H0,&H0,&H1000,&H5001,&H0,
&H0,&H4000
1110 DATA &H400,&H0,&H0,&H1,&H2A00,&H80,
&H0,&H802A
1120 DATA &HAA00,&HE0,&H0,&HA0AA,&HAB02,
&HE8,&H200,&HA8FA
1130 DATA &HAA02,&HA8,&H200,&HA8EA,&HAA0
2,&HA8,&H200,&HA8AA
1140 DATA &HAA02,&HA8,&H200,&HA8AA,&HAA0
0,&HA0,&H0,&HA0AA
1150 DATA &H2A00,&H80,&H0,&H802A,&H0,&H0
,&H0,&H0
1160 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0,&H
0
1170 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0
1180 DATA &H40,&H17,&H0,&H0,&H0,&H0,&H0,
&HC000
1190 DATA &H0,&H0,&H0,&H3003,&H0,&H0,&H0
,&HC0C
1200 DATA &H0,&H0,&H0,&HF30,&HC0,&H0,&H0
,&HF3C
1210 DATA &HFC,&H0,&H0,&H33F,&HFF,&H0,&H
0,&HC33F
1220 DATA &HC0FF,&H0,&H0,&HF03F,&HF0FF,&
H0,&H0,&HFC0F
1230 DATA &HFC3F,&H0,&H0,&HFF0F,&HFF0F,&
H0,&H0,&HFF03
1240 DATA &HFFC3,&HC0,&H0,&HFF03,&HFFF0,
&HF0,&H0,&HFF00
1250 DATA &HFFC,&HFC,&H0,&H3F00,&HFF,&HF
F,&H0,&HF00
1260 DATA &HC0FF,&HC003,&H0,&H300,&HF0FF
,&H0,&H0,&H0
1270 DATA &HFC3F,&H0,&H0,&H0,&HFF03,&H0,
&H0,&H0
1280 DATA &H300,&HC0,&H0,&H0,&H0,&H0,&H0
,&H0
1290 DATA &H0,&H0,&H0,&H0,&H0,&H0,&H0
1300 DIM G(3,17):FOR A=1 TO 3:FOR B=1 TO
 17:READ G(A,B):NEXT:NEXT
1310 DATA 1,2,1,4,5,6,7,5,7,3,5,5,6,7,3,
2,5
1320 DATA 1,2,3,7,5,6,7,5,7,3,4,5,6,7,3,
2,3
1330 DATA 1,2,3,4,5,6,7,5,7,3,4,5,6,7,5,
2,3
1340 DIM P$(24):FOR A=1 TO 24:READ P$(A)
:NEXT
1350 DATA 4,3,44,6,444,16,555,11,66,11,6
66,15,33,11,333,19,22,11,222,26,11,26,11
1,101
1360  SONG$="mb t150 o3 18 eg. 116 e 18
fg...eg.116e18fg...g o4ecdccdecdc ":RETU
RN
```

Program 5: Atari Jackpot by Ray Patrick

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

Atari Jackpot

Ray Patrick

"Jackpot" for the Atari is a random slot-machine simulation that uses a fancy technique to manipulate the image on the screen. Strings are used to hold the character images that are placed on the screen. A pointer to the string is used to specify which character is being displayed on the screen. The character codes on the screen are never changed, but the data that the character code references is changed to simulate the movement. This is done through the pointer.

This technique allows BASIC to appear a lot faster than it really is. If you were to change the character code on the screen instead of changing the pointer, the simulation would slow down considerably and the only thing that could speed it up would be machine language.

The game is very easy to play. All you really have to do is press the P key to play and the E key to end. You will begin each game with five dollars. Each bet is limited to a quarter to force you to be thrifty. You may be glad about this restriction after you realize how difficult it is to win. Payouts are based on odds.

```
HJ 260 CLR
NE 265 OPEN #1,4,0,"K:"
JM 270 GOTO 1540
AK 290 REM **** DRAW SLOT MACHINE **
       *
HF 300 PRINT #6;"{CLEAR}"
JF 310 POSITION 0,0
IF 320 ? #6;" '$$$$$$$$$$$&"
DI 330 FOR I=1 TO 10
FM 340 ? #6;" %          %"
CA 350 NEXT I
IN 360 ? #6;" ($$$$$$$$$$$)"
EE 370 POSITION 3,2:? #6;"'$&'$&'$&"
NB 380 POSITION 3,3:? #6;"% %% %% %"
FE 390 POSITION 3,4:? #6;"($)($)($)"
DJ 400 POSITION 3,7:? #6;"'$$$$$$$&"
DM 410 POSITION 3,8:? #6;"%
       {7 SPACES}%"
EB 420 POSITION 3,9:? #6;"($$$$$$$)"
DK 430 POSITION 5,8:? #6;MONEY
MJ 440 X=129
HO 450 FOR I=0 TO 6 STEP 3
MN 460 POSITION 4+I,3:? #6;CHR$(X)
OE 470 X=X+1
CE 480 NEXT I
NL 490 POSITION 13,9
CD 500 ? #6;"*$)"
AN 510 FOR I=1 TO 6
EE 520 POSITION 15,2+I
NE 530 ? #6;"%"
CB 540 NEXT I
ND 550 POSITION 15,2
NN 560 ? #6;"+"
NI 580 REM ****** MAIN LOOP ******
HL 600 GET #1,A:IF A<>ASC("P") AND A
       <>ASC("E") THEN 600
BK 605 IF A=ASC("E") THEN GRAPHICS 0
       :END
KA 610 X$="          "
HG 620 MONEY=MONEY-0.25:BANK=BANK+0.
       25
PI 630 POSITION 5,8:? #6;"
       {5 SPACES}"
DN 640 POSITION 5,8:? #6;MONEY
NL 650 FOR I=15 TO 0 STEP -1
GP 660 SOUND 0,50,10,I:SOUND 0,60,10
       ,I
CF 670 NEXT I
HD 680 SOUND 0,0,0,0:SOUND 1,0,0,0
LA 700 REM ******* PULL HANDLE *****
       **
AM 720 FOR I=0 TO 3
AN 730 CHSET$(HANDLE,FILL-1)=HANDLE$
       (9,16)
JJ 740 POSITION 15,2+I:? #6;" "
KF 750 POSITION 15,2+I+1
NP 760 ? #6;"+"
GE 770 CHSET$(HANDLE,HANDLE+8)=HANDL
       E$(1,8)
KK 780 SOUND 0,100-3*I,2,8
CI 790 NEXT I
GF 800 SOUND 0,0,0,0
KI 810 FOR I=4 TO 1 STEP -1
NG 820 CHSET$(HANDLE,FILL-1)=HANDLE$
       (1,8)
LO 830 POSITION 15,2+I:? #6;"%"
KH 840 POSITION 15,2+I-1
NP 850 ? #6;"+"
BB 860 CHSET$(HANDLE,FILL-1)=HANDLE$
       (9,16)
KK 870 SOUND 0,100-3*I,2,8
CI 880 NEXT I
NN 890 CHSET$(HANDLE,FILL-1)=HANDLE$
       (1,8)
GG 900 SOUND 0,0,0,0
AG 920 REM ***** SPIN THE WHEELS ***
       **
BB 940 FOR I=1 TO 3
EB 950 FOR J=1 TO 10
EK 960 INDEX(I)=INT(RND(0)*5)+1
BN 970 CHSET$(BEGIN+I*8,BEGIN+I*8+7)
       =IMAGE$(INDEX(I)*8-7,INDEX(I)
       *8)
NL 975 CHSET$(FILL,FILL+7)=FILL$(1,8
       )
FB 980 FOR R=15 TO 0 STEP -5.5
PD 990 SOUND 0,10,10,R
FC 1000 NEXT R
CB 1010 SETCOLOR 2,INT(RND(0)*16),8
JA 1020 SOUND 0,0,0,0
DB 1030 CHSET$(FILL,FILL+7)=FILL$(9,
        16)
EO 1040 NEXT J
ED 1050 CHSET$(BEGIN+I*8,BEGIN+I*8+7
        )=IMAGE$(INDEX(I)*8-7,INDEX(
        I)*8)
EP 1060 NEXT I
FC 1070 SETCOLOR 2,4,10
IO 1090 REM *** CHECK COMBINATIONS *
        **
```

```
CG 1110 IF INDEX(1)=2 AND INDEX(2)=2
        AND INDEX(3)=2 THEN X$=" JA
        CK POT":GOSUB 1200:GOTO 1250
CB 1120 IF INDEX(1)=1 AND INDEX(2)=1
        AND INDEX(3)=1 THEN X$="  4
        TO 1 ":GOSUB 1210:GOTO 1250
ME 1130 IF INDEX(1)=5 AND INDEX(2)=5
        AND INDEX(3)=5 THEN X$="  O
        OPS.. ":GOSUB 1220:GOTO 1250
HL 1140 IF INDEX(1)=INDEX(2) AND IND
        EX(2)=INDEX(3) THEN X$="  2
        TO 1 ":GOSUB 1230:GOTO 1250
BO 1150 IF INDEX(1)=INDEX(2) OR INDE
        X(2)=INDEX(3) OR INDEX(1)=IN
        DEX(3) THEN X$="  1 TO 1 ":G
        OSUB 1240:GOTO 1250
JF 1155 IF BANK<=0 THEN MSG$="MACHIN
        E EMPTY":GOTO 1165
FM 1160 IF MONEY=0 THEN MSG$="OUT OF
        MONEY":GOTO 1165
JJ 1162 GOTO 600
CP 1165 GRAPHICS 2+16:POSITION 4,4:?
        #6;MSG$:POSITION 5,6:? #6;"
        GAME OVER":POSITION 6,8:? #
        6;"TOTAL $";MONEY
EH 1167 GET #1,A:IF A=ASC(" ") THEN
        RUN
NI 1168 GOTO 1167
KE 1180 REM ** ACCOUNTING SUBROUTINE
        S **
KD 1200 MONEY=MONEY+INT(BANK/2):BANK
        =BANK-INT(BANK/2):RETURN
JE 1210 MONEY=MONEY+1:BANK=BANK-1:RE
        TURN
HN 1220 MONEY=MONEY-INT(MONEY/2):BAN
        K=BANK+INT(MONEY/2):RETURN
FK 1230 MONEY=MONEY+0.5:BANK=BANK-0.
        5:RETURN
LP 1240 MONEY=MONEY+0.25:BANK=BANK-0
        .25:RETURN
KH 1250 REM
AE 1280 REM *{4 SPACES}SPECIAL EFFEC
        TS{4 SPACES}*
DD 1340 FOR A=1 TO 2
FM 1350 POSITION 3,6:? #6;X$
AF 1360 SOUND 0,20,10,4:SOUND 0,0,0,
        0
GO 1370 FOR D=15 TO 4 STEP -2.5
FP 1380 POSITION 3,6:? #6;X$
GI 1390 K=36:FOR E=1 TO 3
LO 1400 SOUND 0,K,8,D:SOUND 0,K-10,1
        0,D
HC 1410 K=15:NEXT E
CO 1420 POSITION 3,6:? #6;"
        "
EL 1430 NEXT D
JO 1440 SOUND 0,0,0,0:SOUND 1,0,0,0
EK 1450 NEXT A
JK 1460 GOTO 600
NC 1500 REM INITIALIZATION
NI 1540 DIM CHSET$(1536),OUTLINE$(7*
        8),HANDLE$(2*8),FILL$(2*8),I
        MAGE$(5*8),INDEX(3),X$(9),MS
        G$(14)
ID 1550 CHSET$(1)=CHR$(0):CHSET$(102
        4)=CHR$(0):CHSET$(2)=CHSET$
OB 1560 GRAPHICS 2+16
IH 1570 SETCOLOR 3,12,10
FI 1580 SETCOLOR 2,4,10
CP 1590 SETCOLOR 4,7,2
FA 1600 SETCOLOR 1,10,4
AD 1610 POSITION 6,4:PRINT #6;"JACKP
        OT":POSITION 4,6:? #6;"PLEAS
        E WAIT"
LH 1620 MONEY=5:BANK=10
AL 1630 A=ADR(CHSET$)
PK 1640 START=INT(A/1024)*1024
OA 1650 IF START<A THEN START=START+
        1024
MH 1660 BEGIN=START-A+1
FH 1670 HI=INT(START/256):LO=START-H
        I*256
BG 1680 POKE 203,LO:POKE 204,HI
EI 1690 FOR X=0 TO 27:READ Y:POKE 15
        36+X,Y:NEXT X
IG 1700 DATA 104,169,0,133,205,168,1
        69,224,133,206,177,205,145,2
        03,200,208,249,230,204,230,2
        06,165,206,201,228
OC 1710 DATA 208,239,96
HK 1730 X=USR(1536)
CD 1740 POKE 756,START/256
JA 1750 SCRMEM=PEEK(88)+256*PEEK(89)
KG 1760 FOR I=1 TO 5*8
CM 1770 READ A
DG 1780 IMAGE$(I,I)=CHR$(A)
FJ 1790 NEXT I
EO 1800 DATA 0,28,18,56,124,124,56,0
DM 1810 DATA 0,102,102,0,129,66,60,0
AN 1820 DATA 0,0,24,60,126,126,24,0
EE 1830 DATA 0,24,60,126,126,60,24,0
DP 1840 DATA 0,102,102,0,60,66,129,0
EB 1850 REM OR I=1 TO 2*8
KJ 1860 FOR I=1 TO 7*8
CN 1870 READ A
PE 1880 OUTLINE$(I,I)=CHR$(A)
FK 1890 NEXT I
HA 1900 DATA 0,0,0,255,255,0,0,0
EJ 1910 DATA 24,24,24,24,24,24,24,24
BI 1920 DATA 0,0,0,248,248,24,24,24
KF 1930 DATA 0,0,0,31,31,24,24,24
KG 1940 DATA 24,24,24,31,31,0,0,0
BL 1950 DATA 24,24,24,248,248,0,0,0
DO 1960 DATA 31,31,31,31,31,31,31,31
KG 1970 FOR I=1 TO 2*8
CP 1980 READ A
PN 1990 FILL$(I,I)=CHR$(A)
EK 2000 NEXT I
CF 2010 DATA 170,85,170,85,170,85,17
        0,85
CG 2020 DATA 85,85,85,85,170,170,170
        ,170
JK 2030 FOR I=1 TO 2*8
CD 2040 READ A
HG 2050 HANDLE$(I,I)=CHR$(A)
FA 2060 NEXT I
HK 2070 DATA 126,60,60,60,24,24,24,2
        4
KD 2080 DATA 0,0,0,0,126,60,60,60
DD 2090 IMAGE=BEGIN+8
ED 2100 OUTLINE=IMAGE+3*8
JB 2110 HANDLE=OUTLINE+7*8
DI 2120 FILL=HANDLE+8
MG 2130 CHSET$(IMAGE,OUTLINE-1)=IMAG
        E$(1,3*8)
IK 2140 CHSET$(OUTLINE,HANDLE-1)=OUT
        LINE$
AE 2150 CHSET$(HANDLE,FILL-1)=HANDLE
        $(1,8)
DG 2160 CHSET$(FILL,FILL+7)=FILL$(9,
        16)
JG 2170 GOTO 300
```

The *Complete Personal Accountant* For The Commodore 64

Richard DeVore

The *Complete Personal Accountant* for the Commodore 64 is a powerful personal finance package with many useful options and features. It comes with two diskettes and a 190-page manual. The diskettes contain ten programs which include all the necessary functions and a tutorial to help you get started. The 5½ × 8½-inch bound manual is thorough, but the binding makes it a bit awkward to use while working at the computer—it will not lie open.

Twenty Dollar Insurance

There's an unpleasant surprise in a letter that comes with the package. It reads as follows: "IMPORTANT: If you wish to obtain service from our Technical Support Staff and be advised of any enhancements, program changes, helpful hints, or new products, the information on the next page of this letter must be completed and returned immediately to Futurehouse, Inc. with $20." This policy means that you should add $20 to the cost of the package when making your value/cost comparison.

The software package consists of ten programs which work together. They allow setting up a chart of accounts, keeping track of expenditures, setting up a budget, and trying to keep within it, in addition to computing net worth statements. There are payment and appointment calendar functions as well as graphing and mailing list management programs.

Lightning Demo

The onscreen tutorial, which looks more like a demonstration, covers each of the programs of the *Complete Personal Accountant*.

Starting with the Chart of Accounts, you are shown what the various menu items allow you to accomplish. The speed of the self-paced demonstration is quicker than I was able to keep up with. In most cases, it didn't give me sufficient time to read the complete screen. This offers an impression of what the programs do, but doesn't really teach *how* it is done. The first screens cover most of the menu functions, but as you progress there is a tendency to skip some functions.

The concept of the tutorials is good, and if you run each one several times or have quicker reading and retention than I, you may derive more benefit from them. If Futurehouse slows down the screen-flipping speed in later releases, the tutorial will be more effective.

To use the software you will need a Commodore 64 computer with a 1541 disk drive and several formatted disks on which to store your files. The manual states that a printer is optional, but a printer really is almost mandatory. Even the manual recommends having a printout of the Chart of Accounts available when inputting checkbook information. The Chart of Accounts provided with the program contains 66 different accounts which I found extremely difficult to follow until I had a printout to scan for the proper account number.

To use the *Complete Personal Accountant*, it is necessary to set up your work files first. The Chart of Accounts is the main one, and the manual leads you through its initialization on a step-by-step basis. For your convenience there is already a standard Chart of Accounts set up and numbered. Using this as a model, it was quite easy to configure another to suit my needs. It is not necessary to use the chart provided, but it is necessary to maintain the five major types of accounts within the setup account numbers.

This is clearly shown in the manual and is not restrictive but merely reveals the power of the program. Thought should be given to the accounts and subaccounts that may be most needed or useful prior to doing your Chart of Accounts. This will allow you to make the best use of the program. A separate chart has to be made up for each checking account that you are working with.

Bouncing Checks

Before inputting your checkbook information, you should go

through several months in your checkbook to ascertain the number of transactions required. The program requires that the maximum number of records needed for a month be input so that disk space may be allotted. If you designate too few, it will be necessary to start over when the space is used up. I found this to be awkward, but with the proper forethought it should not pose problems for the user. The checkbook program can handle up to 400 records. This should be sufficient for personal accounts and for all but a few small businesses.

The program is menu operated and reasonably self-explanatory. It appears to be well error-trapped. In working with the checkbook maintenance program, as well as the other sections, it was not possible to lose information without deliberately going against what the manual stated. When I attempted to input information that was not in the context that the screen prompts asked for, the program simply requested that I try again. Should a data entry error be made, the records can be scanned and edited.

While entering checks or deposits, there is a simple method of spreading them over several accounts. This is good when you use one check to pay a credit card and there are purchases that should be applied to different accounts or when a deposit is made that includes income from several sources.

Other handy features of this section include the ability to print your checks from the program. Therefore, the checks and the records have to match. This requires ordering the checks from Futurehouse and having access to a printer, but it may be worth it to you to eliminate extra work. Another feature that speeds up check information input is the ability to simply press the RETURN key on a field where the information is the

same as the previous check. When this is done, the program automatically brings the information forward and inserts it for you.

Once you have set up the Chart of Accounts and input your checking information, the program is ready to work for you. Using the menu-driven format, it is quite easy to establish a budget and compare your monthly expenses to your budgeted expenses. You may also change your budget at any time in order to make it more realistic. With a printer connected, the figures may be printed out for examination at your leisure.

The financial statement portion of the program is a method of ascertaining your net worth. All the input is done through menus. This would include such items as outstanding loans, home mortgages, value of investments, and anything else that pertains to value, whether you own it or owe it.

Financial Records

Once the information has been entered, the program provides both net worth and the ratio of income to expense. These may be printed out on your screen or on paper. By keeping the information updated, you will always be able to determine your financial status. This could be quite useful should you need to take out a loan, because all banks like financial records that they can both read and understand.

The rest of the programs included in the *Complete Personal Accountant* are not directly related to your financial record keeping. They allow you to set up a payment schedule that may be accessed to determine which bills should be paid on a given date, or an appointment calendar for keeping up with your luncheon dates and when to be at the IRS office.

They even include a program that allows graphing your expenses and income along with assets and liabilities. Each of these can be done singly or all on one graph. Just like other portions of the program, these can be printed to screen or paper. This function would be useful if a quick analysis were needed, and besides, the shock value of a graph of your financial situation may be what you need to adhere to a budget.

A more useful portion of the package is the mailing list program. This is saved on its own disk and can contain up to 1200 addresses. There are provisions for updating and sorting. Just like the other programs, it is all menu-driven and very easy to use. Once this is set up, the information may be retrieved in any amount or order desired. Although the manual does not give specific instructions for doing so, mailing labels can be printed.

All things considered, the *Complete Personal Accountant* is a powerful and easy-to-use set of programs. If you need a program to help you keep your finances straight, you should consider this one.

Complete Personal Accountant
Futurehouse, Inc.
P.O. Box 3470
Chapel Hill, NC 27514
$79.95

Star League Baseball

Shay Addams

Writing a sports simulation must be really tough on a programmer because he or she doesn't get to invent the rules of the game. At the same time, the positions, actions, and interactions of the members of the opposing teams have to be smoothly coordinated—according to those rules—and convincingly animated. On top of that, the pro-

grammer's work is ultimately judged on how effectively those rules are implemented in the game without sacrificing playability.

It's even possible to err on the side of authenticity. *Starbowl Football*, the previous effort from the same company that produced this entry, was *too* realistic: Tossing and receiving a pass required moving the receiver exactly under the flying ball and pressing the fire button at precisely the right instant—a near-impossible, frustrating maneuver that took even the most adept joystick maestros a long time to master. Fortunately for the sports-minded, the ball handling techniques in *Star League Baseball* are more accommodating.

Grandstand Viewpoint

In fact, it's one of the most enjoyable sports simulations ever, offering an unusual perspective on the diamond—the view you'd get if you were sitting up in the grandstands behind first base. Joysticks control the action with logically designed patterns. When you're in the field, the ball can be thrown to any of the infielders by pressing the fire button once and moving the stick in the direction of the base's actual position. The location of the man throwing the ball is irrelevant; this makes it easy to learn and execute the moves.

Hit the button twice to return the ball from any player to the pitcher. When he's got the ball, the same action puts him in pitching mode, and he crouches to look for the catcher's signal. Then you hold the button down and move the stick in one of eight directions, each indicating a different type of pitch, to send the ball flying across the plate. The pitcher has the option of changing his mind by releasing the button. This enables him to try to pick off a base runner who looks eager to steal second or third.

At the start of the game, you select from three pitchers, each with his own specialties, which include sinkers, curves, fast balls, and sliders of varying speeds and height. You're better off holding "Knuckles" in the bullpen as a relief pitcher, though. That's right, there's a seventh inning stretch that allows for this option.

Striking Out

To swing the bat, just press the fire button. In addition to visualizing the ball's trajectory, it helps if you glance at its shadow. The distance between the two provides a fair gauge of whether the ball's high, low, or in the strike zone. A batting practice option is convenient for honing this skill to perfection.

You can also bunt, and then control the direction in which the ball travels. After each pitch, big block letters display the results (strike, out, ball, home run, etc.) at the top of the screen. When the catcher tosses the ball back to the pitcher, this display is replaced by the number of strikes, balls and outs, the current inning, and other vital information. A scoreboard also appears between innings, posting the runs scored in each inning.

The batter automatically runs to first upon hitting a fair ball, but you'll soon learn that placement—where the ball lands—makes a big difference in whether you get thrown out or not. Infield hits generally result in failure. Hit to the outfield, and you'll have more time to make it to first; the offense gets joystick control of the outfielder nearest the ball, and must race after the ball. He can snare a fly ball by watching its shadow to figure out where it will land.

Stealing Bases

A runner won't advance to the next base unless you move the stick to the right. This allows you to lead off the base, or even steal. But watch out, because it's easy to get caught in a rundown between a pair of infielders. Episodes like this spark genuine excitement when you're playing the computer or a friend, but the two-player games are definitely more fun. Strategy is as important a role as eye-hand coordination, because it pays to figure out the pitcher's pattern. If he just tossed a ball right down the middle and the count's now three and two, will he repeat himself, or try to fake you out with a high slider? You have only split seconds to make the same decisions you would in the batter's box.

The SID chip recreates the smack of a ball connecting with

a piece of ash, or plopping into a leather glove. And you'll hear some familiar ballpark sounds when the bases are loaded or one of the heavy hitters approaches the plate. The crisply defined characters wear clearly recognizable hats, and are well animated when you put them through their paces. It's impossible to forget which player has the ball, because he's always black instead of his team's color of white or yellow. Until you've learned the ropes, taking on the computer is only good for humiliation, but the satisfaction of pulling off a successful double play or hitting a grand slam against a human opponent is infinitely more exhilarating than shooting down a thousand flying saucers from the planet Mongo.

Star League Baseball
Gamestar, Inc.
1302 State Street
Santa Barbara, CA 93101
Disk or Tape
$31.95 Atari; $29.95 Commodore 64 ©

INSIGHT: Atari

Bill Wilkinson

The assembler listing which accompanies this article is a set of patches to Atari DOS 2.0s. If you own an Atari 1050 drive, these patches will allow you to use it in "enhanced density" mode.

Before we get started with the listing and its explanation, though, let's look at a new tidbit.

Bye-Bye BASIC

Are you an 800XL owner? Do you have an unprotected diskette which boots a machine language program via an AUTORUN.SYS file? Would you like to avoid pushing the OPTION button? Are you willing to follow a few simple steps to do so?

Your 800XL enables and disables the built-in BASIC by changing the contents of location $D301 (54017). In Atari 400s and 800s, this location is usually used to input the state of joysticks 3 and 4. In an 800XL, this port controls various system hardware configurations.

For example, bit 0 of $D301 controls whether the OS ROM is active or whether you are using the RAM underneath it. And—guess what—bit 1 of $D301 controls whether the built-in BASIC is active or not. Specifically, the following table applies:

```
Bit 0  = 1   OS ROM enabled
         0   OS ROM disabled, RAM enabled
Bit 1  = 1   Atari BASIC disabled, RAM enabled
         0   Atari BASIC enabled
```

At least one of the other bits in $D301 is used (to control whether or not the diagnostic ROM is enabled), but the "normal" values for $D301 are either $FF (BASIC disabled) or $FD (BASIC enabled).

No Option Button

So all we need to do is add a couple of instructions to our AUTORUN.SYS file, to select RAM instead of BASIC, and we will no longer have to hold down the OPTION button. For example, we might add:

```
LDA #$FF
STA $D301
```

And, yet, there is an easier way. Remember, Atari LOAD files may consist of multiple segments, each of which starts with a start address and an end address. The entire file starts with a pair of $FF bytes, but it doesn't hurt if there are extra $FF header bytes in front of other segments.

So consider: If we specify that we have a LOAD file which starts at location $D301 *and* ends at location $D301, the DOS file loader will try to load (and thereby *store*) a single byte at location $D301. This is equivalent to storing a byte via our program.

Disabling BASIC

So simply use the following steps to modify your AUTORUN.SYS to disable the built-in BASIC:

Under Atari DOS 2.0s:

1. Boot your DOS disk while holding down the OPTION button.
2. Put the disk containing the AUTORUN.SYS you want to modify into drive 1.
3. Use the E option from the DOS menu. When prompted for old and new filenames, respond:

   D:AUTORUN.SYS,AUTORUN.OLD
4. Use the K option from the DOS menu. When prompted for filename, starting address, etc., respond:

   D:AUTORUN.SYS,D301,D301
5. Use the C option from the DOS menu. When prompted for from and to filenames, respond:

   D:AUTORUN.OLD,AUTORUN.SYS/A

Under OS/A+ or DOS XL:

1. Boot your DOS disk while holding down the OPTION button. If the DOS XL menu appears, use the Q option.
2. Put the disk containing the AUTORUN.SYS you want to modify into drive 1.
3. Type the command:

   RENAME AUTORUN.SYS AUTORUN.OLD
4. Type the command:

   SAVE AUTORUN.SYS D301 D301
5. Type the command:

   COPY −AF AUTORUN.OLD AUTORUN.SYS

And that's it. Your AUTORUN.SYS file should now be ready to use.

ATARI

ATARI is a trademark of ATARI, INC.

Atari 800XL . . . . . Call
1010 Program Recorder 77.00
1027 Letter Quality Printer . . . Call
1050 Disk Drive . . . . . Call
Atari 850 Interface . . . . . Call
Atari Light Pen 82.95
Numerical Keypad 104.95

INTERFACES
Aid Interfast I . . . Call
Ape Face . . . . . Call
R-Verter Modem Adaptor 39.95

DISK DRIVES
Indus GT . . . . . Call
Percom . . . . . Call
Rana 1000 . . . . . Call
Trak . . . . . Call

MOSAIC
48K RAM 94.00
64K RAM/400 . 149.00
64K RAM/800 + Cable Kit #1 . 169.00
Handyman-D . . 23.95
Cable Kit #2 . . . 14.95
Superdrive-D . . 23.95

MODEMS
Hayes Smart Modem 300 . . . . Call
Mark II 79.00
Mark VII/Auto Ans/ Auto Dial . . . . . Call
Mark XII/1200 Baud . . . . . Call
MPP 1000 C . . . . Call
Novation . . . . . Call
R-Verter Modem Adaptor 39.95

MONITORS
AMDEK Color 1 + Call
USI 12" Amber . 149.00
USI 12" Green . 139.00
USI 14" Color . . 269.00
BMC 12" Color . 229.00
BMC 12" Amber 99.00
BMC 12" Green . 95.00
Phillips 12" Green w/Sound 129.00
Phillips 12" Amber w/Sound 139.00

Triangle Replacement Keyboard for 400 54.95

Alien Voice Box II 99.95

Expansion Memory for 600XL . . . . . Call

SUPER PRINTER PACKAGES
Gemini 10X and Apeface 339.00
Prowriter and Apeface 409.00
Prowriter + Aid Interf. + Cable . . 485.00
Gemini 10X and Cardco + G 349.00
Prowriter and Cardco + G 419.00
No additional ship. charges on printer packages in Continental USA

COMMODORE 64

CBM 64 . . . . . Call
SX-64 Portable . . . . . Call
1541 Disk Drive . . . . . Call
1526 Printer 279.00
1530 Datasette 66.00
1702 Monitor . . . . . Call
1650 AD/AA Modem 89.00
RS 232 Interface . . . . . Call

TOUCH TABLETS
Koala Pad-D . . . 69.95
Koala Pad-Cart . 74.95

PRINTERS
Alphacom 40C w/Interface 99.95
Alphacom 80C w/Interface . . . 189.95
Axiom AT-550 . . 329.00
Epson . . . . . Call
Gemini 10X . . . 269.00
Okidata . . . . . Call
Prowriter I . . . . . Call
Riteman . . . . . Call

☆ **Call for prices on joysticks, printer cables, blank floppy disks, books and other computer accessories.** ☆

ATARI SOFTWARE

ATARI
Caverns of Mars-Cart . 32.95
Donkey Kong-Cart . . . 37.95
Donkey Kong Jr.-Cart . 39.95
Eastern Front-Cart . . . 32.95
Joust-Cart 39.95
Jungle Hunt-Cart 35.95
Mario Bros.-Cart 35.95
Ms. Pac-Man-Cart 39.95
Pengo-Cart 35.95
Pole Position-Cart 39.95
Qix-Cart 32.95
Realsports Football-Cart . 35.95
Robotron: 2084-Cart . . 35.95
Star Raiders-Cart 32.95
Tennis-Cart 35.95
Atariwriter-Cart 74.95
Family Finances-D . . . 37.95
Home Filing Manager-D . 37.95
Mailing List-T 19.95
Syncalc-D 74.95
Synfile-D 74.95
Syntrend-D 74.95
Logo-Cart 79.95
My First Alphabet-D . . 26.95
Atarimusic I or II-D . . 29.95
Conversational Lang.-T 44.95
Final Legacy-Cart . . . . Call
Ballblazer-Cart . . . . . Call
Plato-Cart . . . . . Call
Paint-D 33.95
Microsoft Basic II-Cart . 67.95
Atari Macro-Assembler-Cart 67.95
Atari Basic-Cart 41.95
Assembler Editor-Cart . 46.95

BRODERBUND
AE-D 23.95
Arcade Machine-D . . . 41.95
Bank Street Writer-D . . 49.95
Choplifter-D 23.95
Drol-D 23.95
Gumball-D 20.95
Loderunner-D 23.95
Mask of the Sun-D . . . 27.95
Operation Whirlwind-D . 27.95

DATASOFT
Basic Compiler-D 55.95
Bruce Lee-D/T 23.95
Dallas-D 23.95
Heathcliff-D/T 23.95
Letter Wizard . . . . . Call
Micropainter-D 23.95
Money Wizard . . . . . Call

DATASOFT (cont'd.)
Nibbler-D/T 20.95
Spell Wizard-D 34.95
Teletalk-D 34.95
Zaxxon-D/T 27.95

EPYX
Dragons/Pern-D/T . . . 27.95
Fun With Art-Cart 27.95
Gateway to Apshai-Cart 27.95
Jumpman Jr.-Cart 27.95
Jumpman-D/T 27.95
Pitstop-Cart 27.95
Puzzlemania . . . . . Call
Summer Games . . . . . Call
Temple of Apshai-D/T . 27.95

INFOCOM
Deadline-D 34.95
Enchanter-D 34.95
Infidel-D 34.95
Planetfall-D 34.95
Sorcerer-D 34.95
Starcross-D 27.95
Suspended-D 34.95
Witness-D 34.95
Zork I, II, or III-D . . . 27.95
Sea Stalker-D 34.95

MISCELLANEOUS ATARI
Omnimon 82.95
Pitfall-Cart 31.95
River Raid-Cart 31.95
Diskey-D 34.95
Ultra Disassembler-D . 34.95
Zombies-D/T 23.95
Codewriter-D 69.95
Star League Baseball-D/T 22.95
Star Bowl Football-D/T . 22.95
Death in the Caribbean-D 27.95
Dino Eggs-D 27.95
The Heist-D 23.95
Gyruss-Cart 34.95
James Bond-Cart 34.95
Popeye-Cart 34.95
Q-Bert-Cart 34.95
Star Wars-Cart 34.95
Return of Heracles-D . . 26.50
Master Type-D/Cart . . 27.95
Songwriter-D 27.95
Flight Simulator II-D . . 37.95
Chatterbee-D 27.95
S.A.M.-D 41.95
Castle Wolfenstein-D . . 20.95
Compuserve Starter Kit 34.95
Home Accountant-D . . 52.95
Ken Uston's Blackjack-D 49.95

MISC. ATARI (cont'd.)
Megafont-D 19.95
Monkey Wrench II-Cart 49.95
Movie Maker-D 41.95
Pogo Joe-D 20.95
Spelunker-D 27.95
Ultima III-D 41.95
Jupiter Mission-D 34.95
Dreadnaught Factor-Cart . 31.95
Keystone Kapers-Cart . . 31.95
Antica 4-D 37.95
Boulder Dash-D/T . . . 20.95
Scraper Caper-Cart . . . 34.95
Miner 2049'er-Cart . . . 34.95
Beachead-D/T 23.95

OPTIMIZED SYSTEMS
Action-Cart 74.95
Basic XL-Cart 74.95
MAC/65-Cart 74.95
DOS XL-D 27.95
Action Aid-D 27.95

SIERRA ON-LINE
Aquatron-D 20.95
Dark Crystal-D 27.95
Homeword Speller-D . . 34.95
Homeword-D 49.95
Oil's Well-D 23.95
Prisoner-D 23.95
Quest for Tires-D 23.95
Ultima II-D 41.95
Ultima I-D 23.95
Wiztype-D 23.95

SSI
Carrier Force-D 41.95
Combat Leader-D/T . . . 27.95
Cosmic Balance II-D . . 27.95
Cosmic Balance-D 27.95
Broadsides-D 27.95
War in Russia-D 55.95
50 Mission Crush-D . . . 27.95
Questron-D 34.95
Rails West-D 27.95

COMMODORE 64 SOFTWARE

ACCESS
Neutral Zone-D/T 23.95
Spritemaster-D/T 23.95
Beachhead-D/T 23.95
Master Composer-D . . . 27.95

BATTERIES INCLUDED
Consultant-D 69.95
Paperclip w/Spellpak-D . 84.95
Super Busscard II . . . . Call
Home Inventory-D 23.95
Recipe-D 23.95
Audio/Video Cat-D 23.95
Mail List-D 23.95
Stamps-D 23.95

CARDCO
Cardprint/B 47.95
Cardco + G 69.95
Cardboard/5 . . . . . 59.95
Cardkey 39.95
Cassette Recorder 47.95
Printer Utility-D/T . . . 19.95
Write Now-Cart 34.95
Mail Now-D 29.95

COMMODORE
Program Ref. Guide . . . 19.95
Assembler-D 39.95
Easy Finance I,II, III,IV-D 19.95
Easy Calc-D 64.95
Easy Mail-D 19.95
Easy Script-D 44.95
Easy Spell-D 19.95
Logo-D 57.95
The Manager-D 39.95
General Ledger-D 39.95
Accts. Rec.-D 39.95
Accts. Pay.-D 39.95
Magic Desk-D 52.95
Zork I, II or III-D 32.95
Suspended-D 32.95
Starcross-D 32.95
Deadline-D 32.95
Soccer-Cart 29.95

HANDIC
64 Forth-Cart 29.95
64 Graf-Cart 23.95
Stat 64-Cart 23.95
Calc Result Easy-Cart . 34.95
Calc Result Advanced-D . 74.95
Calc Result Adv.-Cart . . 69.95
The Diary-Cart 23.95
The Tool-Cart 29.95

INSTA (CIMMARON)
Insta-Writer-Cart 39.95
Insta-Mail-D 24.95
Insta-File-D 49.95
Management Combo . . . 64.95
Insta-Calc-Cart/D 31.95
Insta-Graph-D 24.95
Insta-Vestor-D 31.95
Insta-Speed-D 99.95
Insta Sched-Cart/D . . . 49.95
Insta-Music-Cart/D . . . 79.95
Invest Combo 74.95

MISC. COMMODORE
Ken Uston's Blackjack-D 49.95
Quick Brown Fox-D/Cart. 34.95
Ultima III-D 41.95
Flight Simulator II-D . . 37.95
Night Mission/ Pinball-D/T 20.95
Praticaic PS-D 59.95
M-File-D 64.95
Word Pro 3+/Spell-D . . 74.95
Home Accountant-D . . . 52.95
Step By Step-D/T 44.95
Barron's Sat.-D 67.95
Bristles-D/T 20.95
Telestar 64-Cart 37.95
Star League Baseball-D/T 20.95
Castle Wolfenstein-D . . 20.95
Mastertype-D/Cart . . . 27.95
Vic Switch 124.95
First Class Mail-D 34.95
Aztec-D 27.95
Miner 2049er-Cart 27.95
Sea Dragon-D/T 23.95
Diskey-D 34.95
Hodge Podge-D/T 19.95
Strip Poker-D 23.95
Mr. Robot-D 23.95
Paint Magic-D 34.95
Pooyan-D/T 20.95
Astro Chase-D/T 20.95
Flip Flop-D/T 20.95

MISC. COMM. (cont'd.)
Basic Building Blks-D . 54.95
Critical Mass-D 27.95
Rescue Squad-D 20.95
Super Text Word Pro.-D . 69.95
Beyond Wolfenstein-D . 23.95
Sam-D 41.95
Chatterbee-D 27.95

SPINNAKER
Adventure Creator-Cart . . 27.95
Aerobics-D 30.95
Aegean Voyage-Cart . . . 27.95
Alf in the Color Caves-C . 27.95
Alphabet Zoo-Cart 23.95
Bubble Burst-Cart 27.95
Cosmic Life-Cart 23.95
Delta Drawing-Cart . . . 27.95
Facemaker-Cart 23.95
Fraction Fever-Cart . . . 23.95
Grandma's House-D . . . 23.95
Jukebox-Cart 27.95
Kids on Keys-Cart 23.95
Kidwriter-D 23.95
Kindercomp-Cart 20.95
Ranch-Cart 27.95
Rhymes/Riddles-D 20.95
Search/ Amazing Thing-D 27.95
Snooper #1-D 30.95
Snooper #2-D 30.95
Story Machine-Cart . . . 27.95
Trains-D 27.95
Up For Grabs-Cart 27.95

SYNAPSE
Blue Max-D/T 23.95
Drelbs-D/T 23.95
Fort Apocalypse-D/T . . 23.95
Necromancer-D/T 23.95
New York City-D/T . . . 23.95
Pharoah's Curse-D/T . . 23.95
Protector II-D/T 23.95
Quasimodo-D/T 23.95
Rainbow Walker-D/T . . 23.95
Relax Stress Reduction Sys. 64.95
Shamus Case II-D/T . . . 23.95
Shamus-D/T 23.95
Slam-Ball-D/T 23.95
Survivor-D/T 23.95
Zaxxon-D/T 27.95
Zepplin-D/T 23.95

To Order Call Toll Free
800-558-0003
For Technical Info, Order Inquiries, or for Wisc. Orders -
414-351-2007

D-Disk T-Cassette Cart-Cartridge

NEW LOWER PRICES

ComputAbility™

NEW PRODUCTS

NO SURCHARGE FOR MASTERCARD OR VISA

ORDERING INFORMATION. Please specify system. For fast delivery send cashier's check, money order or direct bank transfers. Personal and company checks allow 2 weeks to clear. Charges for COD are $3.00. School Purchase Orders welcome. In CONTINENTAL USA, include $3.00 shipping per software order. Include 3% shipping on all Hardware orders, minimum $3.00. Mastercard & Visa please include card # and expiration date. WI residents please add 5% sales tax. HI, AK, FPO, APO, Canadian orders — add 5% shipping, minimum $5.00. All other foreign orders, please add 15% shipping, minimum $10.00. All goods are new and include factory warranty. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call 414-351-2007 to obtain an RA# or your return will NOT be accepted for replacement or repair. Prices and availability are subject to change without notice.

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

ORDER LINES OPEN
Mon-Fri 11 AM - 7 PM CST
Sat 12 PM - 5 PM CST

Check The Pointers

Caution! Even though the built-in BASIC is now disabled, HIMEM (the contents of location $2E5) and RAMTOP (contents of location $6A) will still reflect the 40K byte configuration where BASIC is present. If your program pays attention to one or both of these two values, it would also be worth performing the following steps:

1. Change RAMTOP to reflect the full 48K bytes.
2. Close channel zero (the screen editor).
3. Open channel zero for the E: device.

These steps will insure that all 48K bytes of accessible RAM are in use by your program. I won't go into how to accomplish these here and now. Write if you would like me to show how to code those steps in machine language.

Coming Attractions

A project related to this, which I hope to implement in an upcoming column, would be an "M:" device driver. Once upon a lifetime ago, in this column, I presented such a driver. It used the "excess" memory (between the top of a BASIC program and the bottom of the graphics screen) as a pseudodevice.

I would like to do the same thing again, but this time use the extra memory under the OS ROMs or under the built-in BASIC as a superfast RAM disk. Stay tuned for further developments.

DOS 2.0s For Enhanced Density 1050s

First, I would like to point out that the task of reconfiguring Atari DOS 2.0s for an enhanced density 1050 is difficult. I would also like to note that it is *extremely* difficult (if not impossible) to finish the task if you have only one drive.

So, may I suggest that you cooperate with a friend and his drive if you have only one of your own. If your friend's drive is an 810 or a non-Atari drive, it should be set up as drive 1. Your 1050 should be set up as drive 2.

Also, you should use an assembler capable of placing its object code directly in memory. (For example, AMAC—the Atari Macro Assembler—cannot be used for this job.) This is because loading the DOS-modifier code from a disk will use DOS itself, and you are almost guaranteed to run into conflicts. Atari's Assembler Editor cartridge, the old OSS EASMD, OSS's MAC/65, and (I believe) SYNASSEMBLER will all work properly (though the syntax for SYNASSEMBLER may vary a bit from what I show here).

You should boot a normal Atari DOS 2.0s disk, making sure that you can access a normal single diskette in drive 2 (at least to the point of making sure you can list its directory). Be sure you have at least two (2) blank or junk disks ready and at hand. Then begin.

Patching DOS

Type in the program, as shown herein. You may use automatic line numbering if you wish. Type in just the part from the right of the line numbers. LIST or SAVE the source code to disk and then assemble it. Check it against the listing given here. Do not proceed until you are reasonably sure that you have typed it in and assembled it correctly.

Then change line 1000 to read:

```
1000  .OPT NOLIST,OBJ
```

and assemble the code once more.

Voilà! DOS has been patched!

But, because DOS's DRVTBL has changed format, you *must* now hit the SYSTEM RESET key. Then give the DOS command from your assembler. Assuming that you get to the DOS menu (and if you don't, you did something wrong), it would probably be a good idea to immediately format (menu option I) a blank disk in drive 1 and write the DOS files (option H).

Implementing Enhanced Density

Now comes the tricky part. The way we have patched DOS 2.0s, DOS automatically checks each drive at power-on (or SYSTEM RESET) time to find its current configuration (single density, double density, or enhanced density). But the 1050 assumes it is in single-density mode unless you have inserted an enhanced-density diskette. So, up until now, DOS thinks it is working with all single-density disk drives. How do we change its mind?

The easy way: Turn your power off, put your BASIC (or BASIC XL) cartridge into your machine, and turn the power back on, thus booting the disk we just formatted and wrote DOS files to. Insert a blank disk into the second drive (your 1050). From BASIC, give the following command:

```
XIO 254,#1,0,34,"D2:"
```

If you are a faithful reader, you will recognize that as the format command, given from BASIC. But the 34 in the next-to-last position is new! That's right. As we have patched DOS, a nonzero value given in AUX2 is assumed to be the format command value to be sent to the disk drive. The *only legal values* here are 33 (for single density, à la 810 drives) and 34 (for 1050 enhanced density)!

Now drive 2 contains what we hope is an enhanced-density diskette. Once more, hit SYSTEM RESET so that DOS will recognize the new density. Then give the DOS command from BASIC. Once in DOS, use the H menu option to write the DOS files to drive 2.

If you have performed all these steps correctly, you should now have a bootable enhanced-density diskette in drive 2. You might wish to change your 1050 back to being drive 1 and try to boot from it with this new diskette.

Simpler Commands

The beauty of this system is that, once you have created this one enhanced-density master, you may make new enhanced-density masters by using just the I and H commands from the DOS menu.

There is, however, one potential problem. How do you copy files from an old single-density disk to a new enhanced-density disk? For now, the only practical way is to borrow a second drive and have one of each type of disk on your system. There may be ways around even this problem. We'll see.

Patching Other DOS Versions

The patch program given here will also work on all versions of OS/A+ and DOS XL from 1.2 to 2.3 (except that it will *not* patch the DOSXL.SYS versions).

The procedures are almost the same, but it is significantly easier to use a single drive. Try the following if you have only a single disk, on which you boot OS/A+ or DOS XL:

1. Type in, save, and check out the patch listing as described above.
2. Hit SYSTEM RESET. If you end up back in an assembler cartridge, type a DOS command.
3. From the D1: prompt, use an INIT command. Or use the I option from the DOS XL menu.
4. Use Option 1 (on a blank disk) or 3 (on an existing disk) of INIT. Use Option 4 to return to DOS.
5. Insert a BASIC cartridge. Reboot from the disk you just INITed.
6. Type the following BASIC command:
   XIO 254,#1,0,34,"D1:"
7. Hit SYSTEM RESET after the formatting is finished. If you are not then in the BASIC cartridge, use the CAR command.
8. Type the following BASIC command line:
   OPEN #1,8,0,"D1:DOS.SYS" : CLOSE #1

The reason the procedure works on a single drive is that neither OS/A+ nor DOS XL requires the DUP.SYS file of Atari DOS. The disk initialization can thus be performed entirely from BASIC.

Patches To Atari DOS 2.0s

```
0000          .OPT LIST,NO OBJ
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ; PATCHES TO ATARI DOS 2.0S
        ;
        ; THESE PATCHES ALLOW AN ATARI 1050 DRIVE
        ;    TO UTILIZE ENHANCED DENSITY UNDER
        ;    DOS 2.0S, TO A MAXIMUM OF 965 FREE SECTORS
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ;
        ; EQUATES -- TAKEN FROM THE LISTING OF
        ;                 ATARI DOS AS PUBLISHED IN
        ;                    "INSIDE ATARI DOS"
        ;                 FROM COMPUTE! BOOKS
        ;
=1311   DRVTBL = $1311
=1301   CURFCB = $1301
=0048   ZSBA =  $48
=11DB   DERR1 = $11DB
=12FE   DRVTYP = $12FE
=0021   DCBCFD = '!
=0302   DCBCMD = $0302
=0045   ZDRVA = $45
=0A4A   NOBURST = $0A4A
=0A4C   WRBUR = $0A4C
=0D18   XFORMAT = $0D18
=0BD6   XFV =   $0BD6
=1372   Z   =   $1372
=034B   ICAUX2 = $034B
=1382   FCBOTC = $1382
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

The rest of this listing will appear in "Insight: Atari" next month. ©

ML Tracer

Thomas G. Gordon

Attempting to debug a machine language program can sometimes be a trying experience, especially when the program always seems to exit into the twilight zone. And trying to study a program in ROM can be just as frustrating, even with a disassembler (where do branch instructions go?). Here's an excellent programming utility: a single-stepper for Atari, Apple, and all Commodore computers.

Anyone who has ever worked with machine language knows how helpful it can be to be able to single-step through a program. "ML Tracer" allows you to step through a machine language routine one event at a time and print out the contents of all of the microprocessor registers after each instruction. It also allows you to follow all branches, jumps, and returns. The program will display the address, opcode, mnemonic, and operand of each instruction.

Three versions are included. Program 1 runs on all Commodore computers (for the VIC, 8K or more expansion memory is required). Program 2, for the Apple II, is only slightly different from the Commodore version. The Atari version, Program 3, has more substantial changes, but its structure is still quite similar. Since all the versions have the same line numbers, references in this article apply to all versions unless otherwise stated.

When Tracer is run, there will be a ten-second delay while the DATA statements are read. You'll then be asked for the hex address of the ML program you wish to examine. You can change the contents of any register, before each instruction is executed. Press *a* for the accumulator, *x* for the x register, *y* for the y register, *s* for the stack pointer, *p* for the processor status, or *i* for the instruction pointer (program counter). On the Atari, also press RETURN. When you're through loading registers, press RETURN once more to execute the next instruction.

Hexadecimal numbers are used for all input and output. If you enter an address as a one-, two-, or three-digit hexadecimal number, zeros will be added on the left to make a four-digit number. If too many digits are entered, the rightmost four digits will be used. The same applies to changing the value in a register. The number that you enter will be converted to a two-digit hexadecimal number using the same rules.

The Execution Subroutine

The program is written mostly in BASIC, but contains two machine language subroutines. The first, the initialization subroutine, copies the lowest three pages (768 bytes) of RAM, which are used by BASIC, to a location above the BASIC program. The other, the execution subroutine, exchanges the two three-page blocks of data and loads all the registers with their saved values, then executes one instruction (which has been POKEd in from BASIC). When the instruction has been executed, the registers are saved and BASIC's original lower three pages of memory are restored.

The same technique was used to identify addressing modes as in my disassembler ("A 6502 Disassembler," COMPUTE!, January 1981, p. 81). Lines 10000–10031 contain four-character extended mnemonics for the 6502's instruction set. The fourth character is a tag code identifying the addressing mode of the instruction. In lines 110–120, the mode is identified and the proper subroutine is called.

There are several instructions which cannot be allowed to actually execute in the machine language subroutine. If any control transfer instructions (JMP, JSR, RTS, RTI, or a conditional branch) were executed, control would not be returned properly to the BASIC program. These instructions are simulated in BASIC instead, so that they appear to execute successfully. The SEI and CLI instructions are ignored, since interrupts are always disabled during the execution subroutine.

ATARI®

600XL ..... **CALL**

800XL ..... **CALL**

DISK DRIVES

Indus GT	Call
Trak AT-D2	$388
Trak AT-1	$319
Trak AT-D4	Call
Astra 1620 (Dual)	$499
Percom	Call
Atari 1050	$349
Rana 1000	Call

MEMORIES

Microbits 64K (XL)	$126
Mosaic 48K (400)	$98
Mosaic 64K (400/800)	Call
Mosaic 32K	$68
Atari 64K (600XL)	Call

INTERFACES

Axiom 846	Call
Ape Face	Call
Atari 850 (In Stock)	$169
Interfast 1	$150
Microbits 1150	Call
Axiom Buffer	Call

DIRECT PRINTERS

Axiom AT-100	$219
Atari 1027	$285
Axiom 550 AT	$319
Axiom 700 AT	$469
Atari 1025	$395

DIRECT MODEMS

Microbits 1000C	$128
Atari 1030	$114

OTHER ATARI

400 Keyboard	Call
Koala Pad	$67
Chalkboard Pad	$75
Bit-3 80 Column	$228
ATR-8000 (64K)	$448
ATR-8000 (16K)	$295
Alien Voice Box	$98
1010 Recorder	$74

ATARI SOFTWARE

ADVENTURE INT'L	
Ultra Disassembler (D)	$33
Diskey (D)	$33
Adv. 1-12 (each) (C)	$18
Saga 1-12 (each) (D)	$27
ATARI	
Atari Writer (R)	$68
Paint (D)	$30
Microsoft Basic II (R)	$64
Visicalc (D)	$139
Home File Mgr (D)	$36
Assembler Editor (R)	$44
Qix (R)	$32
Dig Dug (R)	$32
Atari Logo (R)	$72
Robotron (R)	$35
AVALON HILL	
Telengard (C) $16 (D)	$19
Close Assault (C) 20 (D)	23
TAC (D)	$27
BRODERBUND	
Arcade Machine (D)	$39
Bank St. Writer (D)	$46
Oper. Whirlwind (D)	$27
Choplifter (D) $23 (R)	$29
David's Midnight (D)	$23
CBS SOFTWARE	Call
CONTINENTAL	
Home Accountant (D)	$48
Tax Advantage (D)	$45
COUNTERPOINT SW	Call
DATASOFT	
Pooyan (C/D)	$20
Teletalk (D)	$33
Basic Compiler (D)	$55
Graphic Gen (D)	$17
Graphic Master (D)	$23
Micropainter (D)	$23
Zaxxon (C/D)	$27
Text Wizard/Spell (D)	$55
EASTERN HOUSE	
Monkey Wrench II	$51
EDUCATIONAL SW	
Tricky 1,2,3 or 4	$15
Tricky 5-13	$22
EPYX	
Dragon Riders (C/D)	$27
Temple APS (C/D)	$27
Jumpman (C/D)	$27
Pit Stop (R)	$27
FIRST STAR	
Boulder Dsh (C/D) 20 (R)	27
Bristles (C/D)	$20
Flip Flop (C/D)	$20
GAMESTAR	
Football (C/D)	$21
Baseball (C/D)	$21
INFOCOM	
Zork I, II or III (D)	$27
Deadline (D)	$34
Starcross (D)	$27
Suspended (D)	$34
Witness (D)	$34
Planetfall (D)	$34
Enchanter (D)	$34
Infidel (D)	$34
KRELL SAT	Call
INTELL. STATEMENTS	
Prof. Blackjack (D)	$46
LJK	
Letter Perfect (D)	$74
Data Perfect (D)	$74
Spell Perfect (D)	$56
Letter Perfect (R)	$74
MICROPROSE	
Solo Flight (D)	$26
Hellcat Ace (C/D)	$23
MONARCH	
ABC Compiler (D)	$55
OPTIMIZED SYSTEMS	
Action (R)	$65
Basic XL (R)	$65
Mac 65 (D)	$58
C-65 (D)	$58
Bug 65 (D)	$23
PARKER BROS	
Astrochase (R)	$33
Death Star (R)	$33
Q-Bert (R)	$33
Popeye (R)	$33
PENGUIN	
Graphics Magician (D)	$39
Spy's Demise (D)	$21
QUALITY SW	
Return of Hercules (D)	$22
Ali Baba (D)	$22
RESTON	
Moviemaker (D)	$45
SCARBOROUGH SYS.	
Mastertype (D/R)	$27
Songwriter (D)	$27
SCHOOL WIZWARE	Call
SEGA	
Star Trek (R)	$27
Buck Rogers (R)	$27
Congo Bongo (R)	$27
SIERRA ON-LINE	
Homeword (D)	$46
Ultima II (D)	$39
Dark Crystal (D)	$26
Wiz. & Princess (D)	$22
SPINNAKER	
Snooper Troop 1,2 (D)	$30
Most Amazing (D)	$27
Kids on Keys (D)	$20
Trains (D)	$27
Delta Drawing (R)	$27
Aerobics (D)	$34
STRATEGIC SIM.	
Broadsides (D)	$27
Carrier Force (D)	$39
Combat Leader (D)	$27
Rails West (D)	$27
Epidemic (D)	$23
Eagles (D)	$27
Cosmic Bal I or II (D)	$27
SUBLOGIC	
Flight Simulator II (D)	$36
Pinball (C/D)	$20
SYNAPSE	
Syn File + (D)	$65
Syn Text (D)	$65
File Manager (R)	$54
Fort Apocalypse (C/D)	$23
Dimension X (C/D)	$23
Blue Max (C/D)	$23
Encounter (D/R)	$23
Zepplin (C/D)	$23
Pharoah's Curse (C/D)	$23
TRONIX	
S.A.M. (D)	$39
P.M. Animator (D)	$29
Juice (C/D)	$20
Chatterbee (D)	$27
MISCELLANEOUS	
Miner 2049 (R)	$34
Millionaire (C/D)	$45
Zombies (C/D)	$23
Prisoner 2 (D)	$27
Sargon II (D)	$23
Gridrunner (R)	$20
B-Graph (D)	$65
Miles Invoices (D)	$57
Castle Wolfenstein (D)	$20
Odesta Chess (D)	$46
Financial Wizard (D)	$41
Ultima III (D)	$39

SPECIALS

Wico Joystick	$21	Maxells	$25
Wico Trackball	$36	Memorex	$24
Compuserve	$26	Elephants	$18
Covers	$7	Flip 'N File	$20

CBM 64 ..... **CALL**

SX64 ..... **CALL**

1701 MONITOR **$249**

*ASTRA 1620
DISK DRIVE SYSTEM

MORE FOR YOUR MONEY
DOUBLE OR SINGLE DENSITY
TWO DRIVES

FOR ATARI

SPECIAL **$499**

Printers/Etc.

STAR	
Gemini 10X	$268
Gemini 15X	$378
Delta 10	$378
Star Letter Qual.	Call
EPSON	
FX80	Call
MX100	Call
RX 80 F/T	Call
SMITH CORONA	
TP II	$438
JUKI	Call
CITOH	
Prowriter	$335
Prowriter II	$648
Starwriter	$1098
Printmaster	$1448
OKIDATA	
82A	Call
84P	Call
92	Call
93	Call
MANNESMAN	
160L	$588
Spirit	Call

MONITORS

AMDEK	
Color I	$274
V300	$139
V300A	$149
Color II	$449
SAKATA	Call
NEC	
GRN (JB1260)	$115
GRN (JB1201)	$149
Color Composite	$298
RGB Color	$598
PANASONIC	Call

MODEMS

NOVATION	
J-Cat	$99
Apple Cat II	$259
D-Cat	$149
HAYES	
Smartmodem	$209
Smartmodem 1200	$498
Micromodem II	$259
Micromodem IIe	$239

D = Disk
R = ROM (Cartridge)
C = Cassette

COSMIC COMPUTERS
UNLIMITED

727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789

ORDER LINES OPEN MON-SAT 8 am - 8 pm

(800) 626-7642

PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

IN CALIF. (714) 594-5204

FOR TECHNICAL INFO, ORDER INQUIRIES, OR FOR CALIFORNIA ORDERS

Add $2.50 shipping per software order in continental U.S. Add $5.00 shipping per software order for AK, HI, FPO-APO. Add $10.00 or 15% (whichever is greater) per software order for non-U.S. Call for cost of hardware shipping. Calif. residents add 6½% sales tax. Cashiers checks or money orders filled within 24 hours for items in stock. Personal checks require 4 weeks to clear. MasterCard and Visa OK for software only within continental U.S., add 3% surcharge. Include card no., expiration date and signature. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call to obtain one before returning goods for replacement or repair. Prices & availability subject to change.

COMMODORE

DISK DRIVES

MSD (170K)	$349
MSD (Dual) (170Kx2)	$539
Laser (170K)	$325
Commodore 1541	$239
Concord (170K)	Call

80 COLUMN BDS

Batteries 80 Col	$149
Video Pak 80	$129
Z80 Video Pak	$209

DIRECT MODEMS

Hesmodem	$53
1650 Automodem	$99
1600 Modem	Call

INTERFACES

The Connection	$85
Bus Card	$149
Cardco G +	$69
Cardco B	Call
MSD (IEEE)	$98
Cardco 5 Slot	$48
RS-232	Call

DIRECT PRINTERS

MPS 801	$219
Commodore 1526	$288
Cardco LQ/I	$498
1520 Color Printer	$129

RECORDERS

Cardco Recorder	$48
1530 Commodore	Call
Cassette Interface	$29

64 SOFTWARE 64

ACCESS SOFTWARE	
Beached (D)	$24
Neutral Zone (C/D)	$24
Spritemaster (D)	$25
AVALON HILL	
Nuke War (C)	$12
Androm. Conquest (C)	$14
Midway Campaign (C)	$13
Computer Football (C)	$13
Telengard (C) $16 (D)	$19
Flying Ace (C)	$15
Moon Patrol (C)	$17
BATTERIES INCLUDED	
Paper Clip (D)	$69
Consultant (D)	$75
Bus Card	$138
80 Column BD	$138
Paper Clip w/Spell (D)	$85
Spell Pack (D)	$37
Organizer Series (Ea)	$22
BLUE SKY	
Calc Result Adv.	$99
Calc Result Easy	$57
BRODERBUND	
Bank St. Writer (D)	$46
Operat. Whrlwnd (D)	$27
Choplifter (R)	$27
Lode Runner (D) 23 (R)	27
CBS SOFTWARE	
Success with Math (D)	$17
Wbstr Word Game (D)	$20
Learning Bridge (D)	$55
COMMODORE	
Simon's Basic (R)	CALL
Magic Desk (R)	$48
Logo (D)	$45
Assembler 64 (D)	$15
Easy Script 64 (D)	$32
Accounting Pkgs (Ea)	$32
COMPUTER ADV. IDEAS	
Game Show (D)	$27
Win W/Words I or II (D)	$15
Sports Facts (D)	$15
CONTINENTAL S.W.	
Home Accountant (D)	$48
Tax Advantage (D)	$45
FCM (D)	$34
COUNTER POINT SW	
Early Games (Ea)	$20
CREATIVE SOFTWARE	
Moondust (R)	$23
Save New York (R)	$23
Pipes (R)	$23
Household Finance (C)	$20
DATASOFT	
Pooyan (C/D)	$20
Moon Shuttle (D)	$20
ENTECH	
Studio 64 (C/D)	$28
Database 64 (D)	$45
EPYX	
Temple of APS (C/D)	$27
Jumpman (C/D)	$27
Dragonriders (C/D)	$27
Pitstop (R)	$27
Gateway to APS (R)	$27
FIRST STAR	
Astrochase (C/D)	$20
Bristle (C/D)	$20
Flip Flop (C/D)	$20
FUTURE HOUSE	
Comp. Pers. Account.	$56
HES	
Synthesound 64 (D)	$23
Omnicalc (D)	$34
Omniwriter (D)	$45
Hesmon (D)	$27
64 Forth (R)	$39
Multiplan (D)	$65
Turtle Graphics (R)	$39
INFOCOM	
Planet Fall (D)	$34
Enchanter (D)	$34
Infidel (D)	$34
JINSAM	
Mini Jini (R)	$75
MICROSOFTWARE INT'L	
Practicalc 64 (C) 34 (D)	36
Spreadsheet (C) 49 (D)	52
Practifile 64 (D)	$36
MIRAGE CONCEPTS	
Data Base (D)	$68
Word Processor (D)	$68
MICROSPEC	
Database (D)	$44
Mailing List (D)	$32
Checkbook Mgr (D)	$47
G/L (D)	$44
A/P (D)	$47
A/R (D)	$47
PARKER BROS	
Q-Bert (R)	$33
Popeye (R)	$33
Frogger (R)	$33
PROFESSIONAL SW	
Wordpro W/Spell (D)	$68
Spellright (D)	$45
QUIKTEX	
Quick Br. Fox (R)	$49
RAINBOW	
File Assistant (D)	$46
Writers Assistant (D)	$46
Spreadsht Assist. (D)	$56
SCARBOROUGH	
Mastertype (D/R)	$27
Song Writer (D)	$27
SEGA	
Star Trek (R)	$27
Buck Rogers (R)	$27
Congo Bongo (R)	$27
SOFTSMITH	
Touch Typing (C/D)	$21
SOUTHERN SOLUTIONS	
Businessman (D)	$48
Bill Payer (D)	$48
Bill Collector (D)	$48
Paymaster (D)	$48
SPINNAKER	
Snooper 1 or 2 (D)	$27
Aerobics (D)	$34
Kids on Keys (D)	$20
Most Amazing (D)	$27
Kindercomp (D)	$20
Alphabet Zoo (D)	20
Trains (D)	$27
Delta Drawing (R)	$27
Delta Music (R)	$27
SUBLOGIC	
Flight Simulator II (D)	$36
Pinball (C/D)	$22
SYNAPSE	
Ft. Apocalypse (C/D)	$23
Necromancer (C/D)	$23
Zaxxon (D)	$23
Blue Max (D)	$23
Pharoah's Curse (C/D)	$23
Zepplin (D)	$23
Quasimodo (D)	$23
SSI	
Combat Leader (C/D)	$27
Computer Baseball (D)	$27
Eagles (D)	$27
Ringside Seat (D)	$27
Tigers In Snow (C/D)	$27
Battle Normandy (C/D)	$27
TIMEWORKS	
Dungeons of Alg. (C/D)	$17
Robbers Lost (C/D)	$17
Money Mgr. (C/D)	$17
Wall Street (C/D)	$17
Data Manager (C/D)	$17
Elec. Checkbook (C/D)	$17
TOTL	
Totl Text (C) $32 (D)	$34
Label (C) $15 (D)	$17
Time Mgr. (C) $24 (D)	$27
Rsrch Asst. (C) $24 (D)	$27
TRONIX	
S.A.M. (D)	$39
Juice (D)	$23
Chatterbee (D)	$27
MISCELLANEOUS	
Diskey (D)	$33
Barrons Sat (D)	$59
Millionaire (D)	$39
Sargon II (D)	$23
B-Graph (D)	$59
Castle Wolfenstein (D)	$20
Odesta Chess (D)	$46
Ultima III (D)	$39
Prof. Blackjack (D)	$46
Homeword (D)	$46
Pers. Accountant (D)	$23
Karate Devils (D)	$27
Final Flight (D)	$22
MAE Assembler (D)	$47

How Does It Work?

The simplest way to see how the program works is to trace through an example. Suppose the instruction LDA #$20 resides at addresses $03C0–$03C1. For this instruction, the extended mnemonic is LDAB, where LDA stands for LoaD Accumulator, and B is the tag code for immediate addressing. The hexadecimal representation for LDA immediate is $A9, which is equivalent to decimal 169.

Line 50, the top of the main loop, calls the keyboard pause routine at line 7000, which also handles changing registers. In line 55, the variable C is loaded with 169 by PEEKing the memory addressed by B, the instruction pointer. The value of B, 960 in this example, is then converted to hexadecimal characters in line 2000 and PRINTed.

In line 60, NOP instructions are POKEd into the execution routine to take up space after one- or two-byte instructions. The hexadecimal value of the opcode is printed next, and then the mnemonic is retrieved from the array R$(). (In the Atari version, mnemonics are stored in the string R$.) If the mnemonic is a blank, this instruction is undefined and an error message is displayed. Otherwise, the standard (three-character) mnemonic is PRINTed, the opcode is POKEd into the execution routine at OP, and the program counter is incremented to 961.

The ASCII code for B is 66, so the ON GOSUB in line 120 transfers control to line 400. Here, the symbol for the addressing mode, #$ is printed. The one-byte operand routine, at line 3000, PEEKs location 961, pointed to by the program counter. This number is POKEd into OP+1, then converted to hexadecimal and PRINTed. After incrementing the program counter to point to the start of the next instruction, a RETURN is executed at line 3000.

At line 5000, the execution routine is SYSed, CALLed, or USRed depending on which computer you have. The contents of the registers are displayed, and control passes back to line 120. Here, a GOTO 50 takes us back to the top of the loop, where the instruction at $3C2 will be executed.

Tracing Is Educational Too

You will find that this program is most useful for testing small ML programs, such as those called as subroutines from BASIC. It's also good for examining sections of larger programs when you're not sure how a particular routine works. If you're learning machine language, you'll find that the register display is an enormous help in understanding the effects and side effects of each instruction, especially the bits (flags) of the processor status register.

Do be careful, though. Any program is vulnerable when dealing with something as powerful as machine language, and this one is no exception. There are more ways to kill a BASIC program from ML than anyone can name in one sitting, so always be conscientious about saving your programs. After you type this one in, SAVE it before you even think about running it. One typographical error could cause the program to erase itself, or at least lock up the computer.

There are also some ML programs that this tracer can't follow, such as those which disconnect the keyboard or video display (whether intentionally or accidentally). If everything is saved on disk or tape (for real security, take the diskette or cassette out of the drive), you can experiment as much as you want, and then if disaster struck all you'd have to do is just turn the computer off and reload the program.

Program 1: Commodore ML Tracer

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

```
10 GOSUB6000                              :rem 167
35 POKEA,0:POKEX,0:POKEY,0:POKEP,52:POKES
   ,255                                   :rem 63
40 PRINT"START ADDRESS (HEX)";:H$="C000":
   INPUTH$                                :rem 106
45 H$=RIGHT$(H$,4):GOSUB1500:B=D:PRINT"AN
   Y KEY TO STEP"                          :rem 9
50 GOSUB7000:D=FRE(0)                     :rem 197
55 PRINT:C=PEEK(B):D=B:GOSUB2000:PRINTH$"
    ";                                    :rem 148
60 POKEOP+1,234:POKEOP+2,234              :rem 127
70 D=C:GOSUB2000:PRINTRIGHT$(H$,2)" ";
                                          :rem 170
80 IFR$(C)=""THENPRINT"INVALID OPCODE":PR
   INT:GOTO35                             :rem 229
90 R$=LEFT$(R$(C),3):PRINTR$" ";:POKEOP,C
   :B=B+1                                 :rem 175
100 IFR$="BRK"THENPRINT:GOTO35            :rem 141
110 U$=RIGHT$(R$(C),1):IFU$=" "THENGOSUB2
    00:GOTO50                             :rem 126
120 ONASC(U$)-64GOSUB300,400,500,600,700,
    800,900,1000,1100,1200,1300:GOTO50
                                          :rem 156
199 REM{4 SPACES}>IMPLIED MODE<           :rem 42
200 IFR$="RTS"THENGOSUB4000:B=D:GOSUB4000
    :B=D*256+B+1:GOSUB5005:RETURN         :rem 42
203 IFR$<>"RTI"THEN208                    :rem 16
205 GOSUB4000:POKEP,D:GOSUB4000:B=D:GOSUB
    4000:B=D*256+B:GOSUB5005:RETURN
                                          :rem 204
208 IFR$="SEI"ORR$="CLI"THENGOSUB5005:RET
    URN                                    :rem 4
210 GOSUB5000:RETURN                      :rem 242
299 REM{4 SPACES}>ABSOLUTE MODE<          :rem 134
300 PRINT"$";:GOSUB2500                    :rem 68
310 IFR$="JMP"THENB=PEEK(OP+1)+PEEK(OP+2)
    *256:GOSUB5005:RETURN                 :rem 34
320 IFR$<>"JSR"THEN340                    :rem 13
330 B=B-1:D=INT(B/256):GOSUB3500:D=B-INT(
    B/256)*256:GOSUB3500                  :rem 249
335 B=PEEK(OP+1)+PEEK(OP+2)*256:GOSUB5005
    :RETURN                               :rem 141
340 GOSUB5000:RETURN                      :rem 246
```

```
399 REM{4 SPACES}>IMMEDIATE MODE<:rem 183
400 PRINT"#$";:GOSUB3000:GOSUB5000:RETURN
                                    :rem 253
499 REM{4 SPACES}>ZERO PAGE MODE<:rem 134
500 PRINT"$";:GOSUB3000:GOSUB5000:RETURN
                                    :rem 219
599 REM{4 SPACES}>ABSOLUTE,X<       :rem 232
600 PRINT"$";:GOSUB2500:PRINT",X";:GOSUB5
    000:RETURN                      :rem 170
699 REM{4 SPACES}>ABSOLUTE,Y<       :rem 234
700 PRINT"$";:GOSUB2500:PRINT",Y";:GOSUB5
    000:RETURN                      :rem 172
799 REM{4 SPACES}>(INDIRECT,X)<     :rem 46
800 PRINT"($";:GOSUB3000:PRINT",X)";:GOSU
    B5000:RETURN                    :rem 249
899 REM{4 SPACES}>(INDIRECT),Y<     :rem 48
900 PRINT"($";:GOSUB3000:PRINT"),Y";:GOSU
    B5000:RETURN                    :rem 251
999 REM{4 SPACES}>ZERO PAGE,X<      :rem 234
1000 PRINT"$";:GOSUB3000:PRINT",X";:GOSUB
     5000:RETURN                    :rem 209
1099 REM{3 SPACES}>ZERO PAGE,Y<     :rem 19
1100 PRINT"$";:GOSUB3000:PRINT",Y";:GOSUB
     5000:RETURN                    :rem 211
1199 REM{3 SPACES}>RELATIVE JUMP<:rem 202
1200 PRINT"TO ";:D=PEEK(B):B=B+1:D=D+(D>1
     27)*256:D=B+D:B1=D             :rem 52
1210 GOSUB2000:PRINT"$"H$;:BM=BM(INT(C/64
     )):BC=BMANDPEEK(P)             :rem 254
1220 IFBC=(INT(C/32)ANDl)*BMTHENB=B1
                                    :rem 88
1230 GOSUB5005:RETURN               :rem 42
1299 REM{3 SPACES}>INDIRECT JUMP<:rem 193
1300 PRINT"(";:GOSUB2500:PRINT")";:B=PEEK
     (OP+1)+PEEK(OP+2)*256          :rem 118
1310 B=PEEK(B)+PEEK(B+1)*256:GOSUB5005:RE
     TURN                           :rem 160
1499 REM{3 SPACES}> HEX TO DEC <    :rem 137
1500 D=0:FORI=1TOLEN(H$):J=ASC(MID$(H$,I,
     1))-48:D=D*H+J+7*(J>9):NEXT:RETURN
                                    :rem 180
1999 REM{3 SPACES}> DEC TO HEX <    :rem 142
2000 H$="":FORI=1TO4:E=INT(D/H):J=D-E*H:H
     $=CHR$(J+48-7*(J>9))+H$:D=E:NEXT
                                    :rem 192
2005 RETURN                         :rem 167
2499 REM{3 SPACES}> 2BYTE OPERAND <
                                    :rem 165
2500 D=PEEK(B+1):POKEOP+2,D:GOSUB2000:PRI
     NTRIGHT$(H$,2);:GOSUB3000:B=B+1:RETU
     RN                             :rem 90
2999 REM{3 SPACES}> 1BYTE OPERAND <
                                    :rem 169
3000 D=PEEK(B):POKEOP+1,D:GOSUB2000:PRINT
     RIGHT$(H$,2);:B=B+1:RETURN     :rem 124
3499 REM{3 SPACES}> PUSH <          :rem 119
3500 J=PEEK(S):POKEML+512+J,D       :rem 194
3505 IFJ=0THENPRINT:PRINT"WARNING: STACK
     {SPACE}OVERFLOW":J=256         :rem 114
3510 POKES,J-1:RETURN               :rem 57
3999 REM{3 SPACES}> POP <           :rem 43
4000 J=PEEK(S):D=PEEK(ML+513+J)     :rem 23
4005 IFJ=255THENPRINT:PRINT"WARNING: STAC
     K UNDERFLOW":J=-1              :rem 221
4010 POKES,J+1:RETURN               :rem 51
4999 REM{3 SPACES}> EXECUTE ONE INSTRUCTI
     ON <                           :rem 148
5000 SYSML+23                       :rem 237
5005 PRINT:FORK=0TO4:D=PEEK(A+K):GOSUB200
     0                              :rem 107
5010 PRINTMID$(" A= X= Y= S= P=",3*K+1,3)
     ;:PRINTRIGHT$(H$,2);:NEXT:PRINT:RETU
     RN                             :rem 143
5999 REM{3 SPACES}> INITIAL STUFF <
                                    :rem 208
6000 ML=2*4096+8*256                :rem 245
6001 A=ML+240:X=A+1:Y=X+1:S=Y+1:P=S+1:H=1
     6:OP=ML+92                     :rem 239
6002 DIMR$(255):DIMBM(3):FORI=0TO3:READB:
     BM(I)=B:NEXT                   :rem 204
6003 FORT=0TO255:READR$(T):NEXT     :rem 154
6004 READR$:IFR$<>"END"THENPRINT"ERROR IN
      OPCODES":PRINT"CHECK FOR TYPO'S":EN
     D                              :rem 133
6005 I=0:FORT=MLTOML+164:READB:POKET,B:I=
     I+B:NEXT                       :rem 128
6008 IFI<>17737THENPRINT"ERROR IN ML DATA
     ":PRINT"CHECK FOR TYPO'S":END:rem 36
6010 SYSML                          :rem 95
6015 PRINT"{CLR}{7 DOWN}{5 RIGHT}6502 ML
     {SPACE}TRACER{4 DOWN}"         :rem 163
6020 RETURN                         :rem 168
6999 REM{2 SPACES}> PAUSE <         :rem 189
7000 GETA$:IFA$=""THEN7000          :rem 177
7010 IFA$="I"THEND=B:L=4:GOSUB7100:B=D:GO
     TO7000                         :rem 40
7020 IFA$="A"THEND=PEEK(A):L=2:GOSUB7100:
     POKEA,D:GOTO7000               :rem 177
7030 IFA$="X"THEND=PEEK(X):L=2:GOSUB7100:
     POKEX,D:GOTO7000               :rem 247
7040 IFA$="Y"THEND=PEEK(Y):L=2:GOSUB7100:
     POKEY,D:GOTO7000               :rem 251
7050 IFA$="S"THEND=PEEK(S):L=2:GOSUB7100:
     POKES,D:GOTO7000               :rem 234
7060 IFA$="P"THEND=PEEK(P):L=2:GOSUB7100:
     POKEP,D:GOTO7000               :rem 226
7070 RETURN                         :rem 174
7100 PRINTA$"=";:GOSUB2000:INPUTH$:H$=RIG
     HT$(H$,L):GOSUB1500:RETURN     :rem 124
9000 DATA128,64,1,2                 :rem 207
10000 DATABRK ,ORAF,,,,ORAC,ASLC,:rem 142
10001 DATAPHP ,ORAB,ASL ,,,ORAA,ASLA,
                                    :rem 112
10002 DATABPLJ,ORAG,,,,ORAH,ASLH,:rem 228
10003 DATACLC ,ORAE,,,,ORAD,ASLD,:rem 133
10004 DATAJSRA,ANDF,,,BITC,ANDC,ROLC,
                                    :rem 244
10005 DATAPLP ,ANDB,ROL ,,BITA,ANDA,ROLA,
                                    :rem 148
10006 DATABMIJ,ANDG,,,,ANDH,ROLH,:rem 209
10007 DATASEC ,ANDE,,,,AMDD,ROLD,:rem 128
10008 DATARTI ,EORF,,,,EORC,LSRC,:rem 191
10009 DATAPHA ,EORB,LSR ,,JMPA,EORA,LSRA,
                                    :rem 187
10010 DATABVCJ,EORG,,,,EORH,LSRH,:rem 249
10011 DATACLI ,EORE,,,,EORD,LSRD,:rem 163
10012 DATARTS ,ADCF,,,,ADCC,RORC,:rem 138
10013 DATAPLA ,ADCB,ROR ,,JMPK,ADCA,RORA,
                                    :rem 140
10014 DATABVSJ,ADCG,,,,ADCH,RORH,:rem 211
10015 DATASEI ,ADCE,,,,ADCD,RORD,:rem 118
10016 DATA,STAF,,,STYC,STAC,STXC, :rem 36
10017 DATADEY ,,TXA ,,STYA,STAA,STXA,
                                    :rem 192
10018 DATABCCJ,STAG,,,STYH,STAH,STXI,
                                    :rem 73
10019 DATATYA ,STAE,TXS ,,,STAD,,:rem 143
10020 DATALDYB,LDAF,LDXB,,LDYC,LDAC,LDXC,
                                    :rem 24
10021 DATATAY ,LDAB,TAX ,,LDYA,LDAA,LDXA,
                                    :rem 149
10022 DATABCSJ,LDAG,,,LDYH,LDAH,LDXI,
                                    :rem 248
```

```
10023 DATACLV ,LDAE,TSX ,,LDYD,LDAD,LDXE,
                                    :rem 173
10024 DATACPYB,CMPF,,,CPYC,CMPC,DECC,
                                    :rem 250
10025 DATAINY ,CMPB,DEX ,,CPYA,CMPA,DECA,
                                    :rem 148
10026 DATABNEJ,CMPG,,,,CMPH,DECH,:rem 201
10027 DATACLD ,CMPE,,,,CMPD,DECD,:rem 116
10028 DATACPXB,SBCF,,,CPXC,SBCC,INCC,
                                    :rem 250
10029 DATAINX ,SBCB,NOP ,,CPXA,SBCA,INCA,
                                    :rem 160
10030 DATABEQJ,SBCG,,,,SBCI,INCI,:rem 199
10031 DATASED ,SBCE,,,,SBCD,INCD,:rem 118
10032 DATAEND                       :rem 231
20000 DATA162,0,181,0,157,0,41,189
                                    :rem 167
20001 DATA0,1,157,0,42,189,0,2    :rem 217
20002 DATA157,0,43,232,208,236,96,120
                                     :rem 68
20003 DATA162,0,181,0,168,189,0,41
                                    :rem 172
20004 DATA149,0,152,157,0,41,189,0
                                    :rem 174
20005 DATA1,168,189,0,42,157,0,1   :rem 75
20006 DATA152,157,0,42,189,0,2,168
                                    :rem 180
20007 DATA189,0,43,157,0,2,152,157
                                    :rem 180
20008 DATA0,43,232,208,213,186,138,174
                                    :rem 125
20009 DATA243,40,154,141,243,40,172,242
                                    :rem 165
20010 DATA40,174,241,40,173,244,40,72
                                     :rem 62
20011 DATA173,240,40,40,234,234,234,8
                                     :rem 62
20012 DATA141,240,40,104,141,244,40,142
                                    :rem 147
20013 DATA241,40,140,242,40,186,138,174
                                    :rem 167
20014 DATA243,40,154,141,243,40,162,0
                                     :rem 56
20015 DATA181,0,168,189,0,41,149,0
                                    :rem 180
20016 DATA152,157,0,41,189,0,1,168
                                    :rem 179
20017 DATA189,0,42,157,0,1,152,157
                                    :rem 179
20018 DATA0,42,189,0,2,168,189,0   :rem 84
20019 DATA43,157,0,2,152,157,0,43:rem 124
20020 DATA232,208,213,88,96        :rem 100
```

Program 2: Apple ML Tracer

```
10  GOSUB 6000
35  POKE A,0: POKE X,0: POKE Y,0:
     POKE P,52: POKE S,255
40  PRINT "START ADDRESS (HEX)";:
     INPUT H$
42  IF H$ = "" THEN H$ = "C000"
45 H$ =  RIGHT$ (H$,4): GOSUB 150
    0:B = D: PRINT "ANY KEY TO S
    TEP"
50  GOSUB 7000:D =  FRE (0)
55  PRINT :C =  PEEK (B):D = B: GOSUB
    2000: PRINT H$" ";
60  POKE OP + 1,234: POKE OP + 2,
    234
70 D = C: GOSUB 2000: PRINT  RIGHT$
    (H$,2)" ";
80  IF R$(C) = "" THEN  PRINT "IN
    VALID OPCODE": PRINT : GOTO
    35
90 R$ =  LEFT$ (R$(C),3): PRINT R
    $" ";: POKE OP,C:B = B + 1
100  IF R$ = "BRK" THEN  PRINT : GOTO
    35
110 U$ =  RIGHT$ (R$(C),1): IF U$
     = " " THEN  GOSUB 200: GOTO
    50
120  ON  ASC (U$) - 64 GOSUB 300,
    400,500,600,700,800,900,1000
    ,1100,1200,1300: GOTO 50
199  REM      >IMPLIED MODE<
200  IF R$ = "RTS" THEN  GOSUB 40
    00:B = D: GOSUB 4000:B = D *
    256 + B + 1: GOSUB 5005: RETURN
203  IF R$ <  > "RTI" THEN 208
205  GOSUB 4000: POKE P,D: GOSUB
    4000:B = D: GOSUB 4000:B = D
     * 256 + B: GOSUB 5005: RETURN
208  IF R$ = "SEI" OR R$ = "CLI" THEN
     GOSUB 5005: RETURN
210  GOSUB 5000: RETURN
299  REM      >ABSOLUTE MODE<
300  PRINT "$";: GOSUB 2500
310  IF R$ = "JMP" THEN B =  PEEK
    (OP + 1) +  PEEK (OP + 2) *
    256: GOSUB 5005: RETURN
320  IF R$ <  > "JSR" THEN 340
330 B = B - 1:D =  INT (B / 256):
     GOSUB 3500:D = B -  INT (B /
    256) * 256: GOSUB 3500
335 B =  PEEK (OP + 1) +  PEEK (O
    P + 2) * 256: GOSUB 5005: RETURN
340  GOSUB 5000: RETURN
399  REM      >IMMEDIATE MODE<
400  PRINT "#$";: GOSUB 3000: GOSUB
    5000: RETURN
499  REM      >ZERO PAGE MODE<
500  PRINT "$";: GOSUB 3000: GOSUB
    5000: RETURN
599  REM      >ABSOLUTE,X<
600  PRINT "$";: GOSUB 2500: PRINT
    ",X";: GOSUB 5000: RETURN
699  REM      >ABSOLUTE,Y<
700  PRINT "$";: GOSUB 2500: PRINT
    ",Y";: GOSUB 5000: RETURN
799  REM      >(INDIRECT,X)<
800  PRINT "($";: GOSUB 3000: PRINT
    ",X)";: GOSUB 5000: RETURN
899  REM      >(INDIRECT),Y<
900  PRINT "($";: GOSUB 3000: PRINT
    "),Y";: GOSUB 5000: RETURN
999  REM      >ZERO PAGE,X<
1000  PRINT "$";: GOSUB 3000: PRINT
    ",X";: GOSUB 5000: RETURN
1099  REM     >ZERO PAGE,Y<
1100  PRINT "$";: GOSUB 3000: PRINT
    ",Y";: GOSUB 5000: RETURN
1199  REM     >RELATIVE JUMP<
1200  PRINT "TO ";:D =  PEEK (B):
    B = B + 1:D = D - (D > 127) *
    256:D = B + D:B1 = D
1210  GOSUB 2000: PRINT "$"H$;:BM
     = BM( INT (C / 64)):BC =  INT
    ( PEEK (P) / BM):BC = BC - 2
     *  INT (BC / 2)
1220  IF BC = ( INT (C / 32) - 2 *
     INT (C / 64)) THEN B = B1
```

```
1230  GOSUB 5005: RETURN
1299  REM     >INDIRECT JUMP<
1300  PRINT "(";: GOSUB 2500: PRINT
      ")";:B =  PEEK (OP + 1) +  PEEK
      (OP + 2) * 256
1310 B =  PEEK (B) +  PEEK (B + 1
      ) * 256: GOSUB 5005: RETURN

1499  REM     > HEX TO DEC <
1500 D = 0: FOR I = 1 TO  LEN (H$
      ):J =  ASC ( MID$ (H$,I,1)) -
      48:D = D * H + J - 7 * (J >
      9): NEXT : RETURN
1999  REM     > DEC TO HEX <
2000 H$ = "": FOR I = 1 TO 4:E =
       INT (D / H):J = D - E * H:H
      $ =  CHR$ (J + 48 + 7 * (J >
      9)) + H$:D = E: NEXT
2005  RETURN
2499  REM     > 2BYTE OPERAND <
2500 D =  PEEK (B + 1): POKE OP +
      2,D: GOSUB 2000: PRINT  RIGHT$
      (H$,2);: GOSUB 3000:B = B +
      1: RETURN
2999  REM     > 1BYTE OPERAND <
3000 D =  PEEK (B): POKE OP + 1,D
      : GOSUB 2000: PRINT  RIGHT$
      (H$,2);:B = B + 1: RETURN
3499  REM     > PUSH <
3500 J =  PEEK (S): POKE ML + 512
       + J,D
3505  IF J = 0 THEN  PRINT : PRINT
      "WARNING: STACK OVERFLOW":J =
      256
3510  POKE S,J - 1: RETURN
3999  REM     > POP <
4000 J =  PEEK (S):D =  PEEK (ML +
      513 + J)
4005  IF J = 255 THEN  PRINT : PRINT
      "WARNING: STACK UNDERFLOW":J
       =  - 1
4010  POKE S,J + 1: RETURN
4999  REM     > EXECUTE ONE INSTRU
      CTION <
5000  CALL (ML + 23)
5005  PRINT : FOR K = 0 TO 4:D =
       PEEK (A + K): GOSUB 2000
5010  PRINT  MID$ (" A= X= Y= S=
      P=",3 * K + 1,3);: PRINT  RIGHT$
      (H$,2);: NEXT : PRINT : RETURN

5999  REM     > INITIAL STUFF <
6000 ML = 2 * 4096 + 8 * 256
6001 A = ML + 240:X = A + 1:Y = X
       + 1:S = Y + 1:P = S + 1:H =
      16:OP = ML + 92
6002  DIM R$(255): DIM BM(3): FOR
      I = 0 TO 3: READ B:BM(I) = B
      : NEXT
6003  FOR T = 0 TO 255: READ R$(T
      ): NEXT
6004  READ R$: IF R$ <  > "END" THEN
       PRINT "ERROR IN OPCODES": PRINT
      "CHECK FOR TYPO'S": END
6005 I = 0: FOR T = ML TO ML + 16
      4: READ B: POKE T,B:I = I +
      B: NEXT
6008  IF I <  > 17737 THEN  PRINT
      "ERROR IN ML DATA": PRINT "C
      HECK FOR TYPO'S": END
6010  CALL ML

6015  HOME : PRINT "6502 ML TRACE
      R"
6020  RETURN
6999  REM    > PAUSE <
7000  GET A$: IF A$ = "" THEN 700
      0
7010  IF A$ = "I" THEN D = B:L =
      4: GOSUB 7100:B = D: GOTO 70
      00
7020  IF A$ = "A" THEN D =  PEEK
      (A):L = 2: GOSUB 7100: POKE
      A,D: GOTO 7000
7030  IF A$ = "X" THEN D =  PEEK
      (X):L = 2: GOSUB 7100: POKE
      X,D: GOTO 7000
7040  IF A$ = "Y" THEN D =  PEEK
      (Y):L = 2: GOSUB 7100: POKE
      Y,D: GOTO 7000
7050  IF A$ = "S" THEN D =  PEEK (S)
      :L = 2: GOSUB 7100: POKE S,D: GOTO
      7000
7060  IF A$ = "P" THEN D =  PEEK
      (P):L = 2: GOSUB 7100: POKE
      P,D: GOTO 7000
7065  IF A$ =  CHR$ (3) THEN  STOP

7070  RETURN
7100  PRINT A$"=";: GOSUB 2000:A$
       = H$: INPUT H$: IF H$ = "" THEN
      H$ = A$
7110 H$ =  RIGHT$ (H$,L): GOSUB 1
      500: RETURN
9000  DATA 128,64,1,2
10000  DATA BRK ,ORAF,,,,ORAC,ASL
      C,
10001  DATA PHP ,ORAB,ASL ,,,ORAA
      ,ASLA,
10002  DATA BPLJ,ORAG,,,,ORAH,ASL
      H,
10003  DATA CLC ,ORAE,,,,ORAD,ASL
      D,
10004  DATA JSRA,ANDF,,,BITC,ANDC
      ,ROLC,

10005  DATA PLP ,ANDB,ROL ,,BITA,
      ANDA,ROLA,
10006  DATA BMIJ,ANDG,,,,ANDH,ROL
      H,
10007  DATA SEC ,ANDE,,,,AMDD,ROL
      D,
10008  DATA RTI ,EORF,,,,EORC,LSR
      C,
10009  DATA PHA ,EORB,LSR ,,JMPA,
      EORA,LSRA,
10010  DATA BVCJ,EORG,,,,EORH,LSR
      H,
10011  DATA CLI ,EORE,,,,EORD,LSR
      D,
10012  DATA RTS ,ADCF,,,,ADCC,ROR
      C,
10013  DATA PLA ,ADCB,ROR ,,JMPK,
      ADCA,RORA,
10014  DATA BVSJ,ADCG,,,,ADCH,ROR
      H,
10015  DATA SEI ,ADCE,,,,ADCD,ROR
      D,
10016  DATA ,STAF,,,STYC,STAC,STX
      C,
10017  DATA DEY ,,TXA ,,STYA,STAA
      ,STXA,
```

```
10018  DATA BCCJ,STAG,,,STYH,STAH
     ,STXI,
10019  DATA TYA ,STAE,TXS ,,,STAD
     ,,
10020  DATA LDYB,LDAF,LDXB,,LDYC,
     LDAC,LDXC,
10021  DATA TAY ,LDAB,TAX ,,LDYA,
     LDAA,LDXA,
10022  DATA BCSJ,LDAG,,,LDYH,LDAH
     ,LDXI,
10023  DATA CLV ,LDAE,TSX ,,LDYD,
     LDAD,LDXE,
10024  DATA CPYB,CMPF,,,CPYC,CMPC
     ,DECC,
10025  DATA INY ,CMPB,DEX ,,CPYA,
     CMPA,DECA,
10026  DATA BNEJ,CMPG,,,,CMPH,DEC
     H,
10027  DATA CLD ,CMPE,,,,CMPD,DEC
     D,
10028  DATA CPXB,SBCF,,,CPXC,SBCC
     ,INCC,
10029  DATA INX ,SBCB,NOP ,,CPXA,
     SBCA,INCA,
10030  DATA BEQJ,SBCG,,,,SBCI,INC
     I,
10031  DATA SED ,SBCE,,,,SBCD,INC
     D,
10032  DATA END
20000  DATA 162,0,181,0,157,0,41,
     189
20001  DATA 0,1,157,0,42,189,0,2

20002  DATA 157,0,43,232,208,236,
     96,120
20003  DATA  162,0,181,0,168,189,
     0,41
20004  DATA 149,0,152,157,0,41,18
     9,0
20005  DATA 1,168,189,0,42,157,0,
     1
20006  DATA 152,157,0,42,189,0,2,
     168
20007  DATA 189,0,43,157,0,2,152,
     157
20008  DATA 0,43,232,208,213,186,
     138,174
20009  DATA 243,40,154,141,243,40
     ,172,242
20010  DATA 40,174,241,40,173,244
     ,40,72
20011  DATA 173,240,40,40,234,234
     ,234,8
20012  DATA 141,240,40,104,141,24
     4,40,142
20013  DATA 241,40,140,242,40,186
     ,138,174
20014  DATA  243,40,154,141,243,4
     0,162,0
20015  DATA 181,0,168,189,0,41,14
     9,0
20016  DATA 152,157,0,41,189,0,1,
     168
20017  DATA 189,0,42,157,0,1,152,
     157
20018  DATA 0,42,189,0,2,168,189,
     0
20019  DATA 43,157,0,2,152,157,0,
     43
20020  DATA 232,208,213,88,96
```

Program 3: Atari ML Tracer

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

```
KH 10 GOSUB 6000
DP 35 POKE A,0:POKE X,0:POKE Y,0:POK
     E P,52:POKE S,255
HA 40 PRINT "START ADDRESS (HEX)";:I
     NPUT H$
JC 42 I=LEN(H$)-3:IF I<1 THEN I=1:IF
      NOT LEN(H$) THEN H$="C000"
EE 45 H$=H$(I):GOSUB 1500:B=D:PRINT
     "HIT RETURN TO STEP"
MF 50 GOSUB 7000:D=FRE(0)
MP 55 PRINT :C=PEEK(B):D=B:GOSUB 200
     0:PRINT H$;" ";
HP 60 POKE OP+1,234:POKE OP+2,234
BI 70 D=C:GOSUB 2000:PRINT H$(3);" "
     ;
OO 75 O$=R$(C*4+1,C*4+3):U$=R$(C*4+4
     ,C*4+4)
EO 80 IF O$="{3 SPACES}" THEN PRINT
     "INVALID OPCODE":PRINT :GOTO 3
     5
PB 90 PRINT O$;" ";:POKE OP,C:B=B+1
IK 100 IF O$="BRK" THEN PRINT :GOTO
      35
DE 110 IF U$=" " THEN GOSUB 200:GOTO
       50
JM 120 ON ASC(U$)-64 GOSUB 300,400,5
     00,600,700,800,900,1000,1100,
     1200,1300:GOTO 50
CK 199 REM >IMPLIED MODE<
CH 200 IF O$="RTS" THEN GOSUB 4000:B
     =D:GOSUB 4000:B=D*256+B+1:GOS
     UB 5005:RETURN
AN 203 IF O$<>"RTI" THEN 208
MM 205 GOSUB 4000:POKE P,D:GOSUB 400
     0:B=D:GOSUB 4000:B=B*256+D:GO
     SUB 5005:RETURN
PO 208 IF O$="SEI" OR O$="CLI" THEN
     GOSUB 5005:RETURN
PC 210 GOSUB 5000:RETURN
IG 299 REM >ABSOLUTE MODE<
EE 300 PRINT "$";:GOSUB 2500
BP 310 IF O$="JMP" THEN B=PEEK(OP+1)
     +PEEK(OP+2)*256:GOSUB 5005:RE
     TURN
AK 320 IF O$<>"JSR" THEN 340
PJ 330 B=B-1:D=INT(B/256):GOSUB 3500
     :D=B-INT(B/256)*256:GOSUB 350
     0
IN 335 B=PEEK(OP+1)+PEEK(OP+2)*256:G
     OSUB 5005:RETURN
PG 340 GOSUB 5000:RETURN
LH 399 REM >IMMEDIATE MODE<
PN 400 PRINT "#$";:GOSUB 3000:GOSUB
     5000:RETURN
IG 499 REM >ZERO PAGE MODE<
NL 500 PRINT "$";:GOSUB 3000:GOSUB 5
     000:RETURN
OI 599 REM >ABSOLUTE,X<
KK 600 PRINT "$";:GOSUB 2500:PRINT "
     ,X";:GOSUB 5000:RETURN
OK 699 REM >ABSOLUTE,Y<
KM 700 PRINT "$";:GOSUB 2500:PRINT "
     ,Y";:GOSUB 5000:RETURN
CO 799 REM >(INDIRECT,X)<
PJ 800 PRINT "($";:GOSUB 3000:PRINT
     ",X)";:GOSUB 5000:RETURN
```

```
DA 899 REM >(INDIRECT),Y<
PL 900 PRINT "($";:GOSUB 3000:PRINT
       "),Y";:GOSUB 5000:RETURN
OK 999 REM >ZERO PAGE,X<
NB 1000 PRINT "$";:GOSUB 3000:PRINT
       ",X";:GOSUB 5000:RETURN
BD 1099 REM >ZERO PAGE,Y<
ND 1100 PRINT "$";:GOSUB 3000:PRINT
       ",Y";:GOSUB 5000:RETURN
MK 1199 REM >RELATIVE JUMP<
DG 1200 PRINT "TO ";:D=PEEK(B):B=B+1
       :D=D-(D>127)*256:D=B+D:B1=D
PN 1210 GOSUB 2000:PRINT "$";H$;:BM=
       BM(INT(C/64)):BC=INT(PEEK(P)
       /BM):BC=BC-2*INT(BC/2)
DM 1220 IF BC=(INT(C/32)-2*INT(C/64)
       ) THEN B=B1
CK 1230 GOSUB 5005:RETURN
MB 1299 REM >INDIRECT JUMP<
AJ 1300 PRINT "(";:GOSUB 2500:PRINT
       ")";:B=PEEK(OP+1)+PEEK(OP+2)
       *0
KA 1310 B=PEEK(B)+PEEK(B+1)*256:GOSU
       B 5005:RETURN
IJ 1499 REM > HEX TO DEC <
ON 1500 D=0:FOR I=1 TO LEN(H$):J=ASC
       (H$(I,I))-48:D=D*H+J-7*(J>9)
       :NEXT I:RETURN
IO 1999 REM > DEC TO HEX <
JD 2000 H$="":FOR I=1 TO 4:E=INT(D/H
       ):J=D-E*H:I$=H$:H$=CHR$(J+48
       +7*(J>9)):H$(2)=I$:D=E:NEXT I
KH 2005 RETURN
KF 2499 REM > 2BYTE OPERAND <
IN 2500 D=PEEK(B+1):POKE OP+2,D:GOSU
       B 2000:PRINT H$(3);:GOSUB 30
       00:B=B+1:RETURN
KJ 2999 REM > 1BYTE OPERAND <
KP 3000 D=PEEK(B):POKE OP+1,D:GOSUB
       2000:PRINT H$(3);:B=B+1:RETU
       RN
HH 3499 REM > PUSH <
MC 3500 J=PEEK(S):POKE ML+512+J,D
HC 3505 IF J=0 THEN PRINT :PRINT "WA
       RNING: STACK OVERFLOW":J=256
DJ 3510 POKE S,J-1:RETURN
CL 3999 REM > POP <
BH 4000 J=PEEK(S):D=PEEK(ML+513+J)
NN 4005 IF J=255 THEN PRINT :PRINT "
       WARNING: STACK UNDERFLOW":J=
       -1
DD 4010 POKE S,J+1:RETURN
JE 4999 REM > EXECUTE ONE INSTRUCTIO
       N <
FJ 5000 POKE 54286,0
LM 5001 D=USR(ML+24)
JF 5002 POKE 54286,64
GL 5005 PRINT :FOR K=0 TO 4:D=PEEK(A
       +K):GOSUB 2000
KO 5010 PRINT REG$(3*K+1,3*K+3);:PRI
       NT H$(3);:NEXT K:PRINT :RETU
       RN
NA 5999 REM > INITIAL STUFF <
MP 6000 ML=6*4096
PC 6020 A=ML+240:X=A+1:Y=X+1:S=Y+1:P
       =S+1:H=16:OP=ML+94
CB 6030 DIM R$(1024),H$(12),I$(12),O
       $(3),U$(1),REG$(15),BM(3):FO
       R I=0 TO 3:READ B:BM(I)=B:NE
       XT I
BH 6035 REG$=" A= X= Y= S= P="
EF 6040 FOR T=0 TO 255:READ H$:IF H$
       ="" THEN H$="(4 SPACES)"
BK 6045 R$(T*4+1)=H$:NEXT T
HC 6050 READ H$:IF H$<>"END" THEN PR
       INT "ERROR IN OPCODES":PRINT
        "CHECK FOR TYPO'S":END
NH 6060 I=0:FOR T=ML TO ML+166:READ
       B:POKE T,B:I=I+B:NEXT T
CE 6070 IF I<>19457 THEN PRINT "ERRO
       R IN ML DATA":PRINT "CHECK F
       OR TYPO'S":END
DD 6080 D=USR(ML)
MH 6090 PRINT "6502 ML TRACER"
KH 6100 RETURN
LN 6999 REM > PAUSE <
MD 7000 INPUT H$
CP 7010 IF H$="I" THEN D=B:L=4:GOSUB
        7100:B=D:GOTO 7000
LI 7020 IF H$="A" THEN D=PEEK(A):L=2
       :GOSUB 7100:POKE A,D:GOTO 70
       00
PO 7030 IF H$="X" THEN D=PEEK(X):L=2
       :GOSUB 7100:POKE X,D:GOTO 70
       00
AC 7040 IF H$="Y" THEN D=PEEK(Y):L=2
       :GOSUB 7100:POKE Y,D:GOTO 70
       00
PB 7050 IF H$="S" THEN D=PEEK(S):L=2
       :GOSUB 7100:POKE S,D:GOTO 70
       00
OJ 7060 IF H$="P" THEN D=PEEK(P):L=2
       :GOSUB 7100:POKE P,D:GOTO 70
       00
KO 7070 RETURN
OJ 7100 PRINT H$;"=";:INPUT H$:I=LEN
       (H$)-L+1:IF I<1 THEN I=1:IF
        NOT LEN(H$) THEN RETURN
BD 7120 H$=H$(I):GOSUB 1500:RETURN
MP 9000 DATA 128,64,1,2
IO 10000 DATA BRK ,ORAF,,,,ORAC,ASLC
       ,
HA 10001 DATA PHP ,ORAB,ASL ,,,ORAA,
       ASLA,
OE 10002 DATA BPLJ,ORAG,,,,ORAH,ASLH
       ,
IF 10003 DATA CLC ,ORAE,,,,ORAD,ASLD
       ,
PE 10004 DATA JSRA,ANDF,,,BITC,ANDC,
       ROLC,
JE 10005 DATA PLP ,ANDB,ROL ,,BITA,A
       NDA,ROLA,
NB 10006 DATA BMIJ,ANDG,,,,ANDH,ROLH
       ,
IA 10007 DATA SEC ,ANDE,,,,AMDD,ROLD
       ,
LP 10008 DATA RTI ,EORF,,,,EORC,LSRC
       ,
LL 10009 DATA PHA ,EORB,LSR ,,JMPA,E
       ORA,LSRA,
PJ 10010 DATA BVCJ,EORG,,,,EORH,LSRH
       ,
KD 10011 DATA CLI ,EORE,,,,EORD,LSRD
       ,
IK 10012 DATA RTS ,ADCF,,,,ADCC,RORC
       ,
IM 10013 DATA PLA ,ADCB,ROR ,,JMPK,A
       DCA,RORA,
ND 10014 DATA BVSJ,ADCG,,,,ADCH,RORH
       ,
HG 10015 DATA SEI ,ADCE,,,,ADCD,RORD
```

```
CE 10016 DATA ,STAF,,,STYC,STAC,STXC
         ,
MA 10017 DATA DEY ,,TXA ,,STYA,STAA,
         STXA,
EJ 10018 DATA BCCJ,STAG,,,STYH,STAH,
         STXI,
IP 10019 DATA TYA ,STAE,TXS ,,,STAD,
         ,
BI 10020 DATA LDYB,LDAF,LDXB,,LDYC,L
         DAC,LDXC,
JF 10021 DATA TAY ,LDAB,TAX ,,LDYA,L
         DAA,LDXA,
PI 10022 DATA BCSJ,LDAG,,,LDYH,LDAH,
         LDXI,
KN 10023 DATA CLV ,LDAE,TSX ,,LDYD,L
         DAD,LDXE,
PK 10024 DATA CPYB,CMPF,,,CPYC,CMPC,
         DECC,
JE 10025 DATA INY ,CMPB,DEX ,,CPYA,C
         MPA,DECA,
MJ 10026 DATA BNEJ,CMPG,,,,CMPH,DECH
         ,
HE 10027 DATA CLD ,CMPE,,,,CMPD,DECD
         ,
PK 10028 DATA CPXB,SBCF,,,CPXC,SBCC,
         INCC,
KA 10029 DATA INX ,SBCB,NOP ,,CPXA,S
         BCA,INCA,
MH 10030 DATA BEQJ,SBCG,,,,SBCI,INCI
         ,
HG 10031 DATA SED ,SBCE,,,,SBCD,INCD
         ,
OH 10032 DATA END
KF 20000 DATA 104,162,0,181,0,157,0,
         97
FE 20001 DATA 189,0,1,157,0,98,189,0
DO 20002 DATA 2,157,0,99,232,208,236
         ,96
DP 20003 DATA 120,104,162,0,181,0,16
         8,189
IH 20004 DATA 0,97,149,0,152,157,0,9
         7
MH 20005 DATA 189,0,1,168,189,0,98,1
         57
EP 20006 DATA 0,1,152,157,0,98,189,0
FL 20007 DATA 2,168,189,0,99,157,0,2
IF 20008 DATA 152,157,0,99,232,208,2
         13,186
MB 20009 DATA 138,174,243,96,154,141
         ,243,96
LJ 20010 DATA 172,242,96,174,241,96,
         173,244
FA 20011 DATA 96,72,173,240,96,40,23
         4,234
HE 20012 DATA 234,8,141,240,96,104,1
         41,244
IL 20013 DATA 96,142,241,96,140,242,
         96,186
LN 20014 DATA 138,174,243,96,154,141
         ,243,96
LK 20015 DATA 162,0,181,0,168,189,0,
         97
LM 20016 DATA 149,0,152,157,0,97,189
         ,0
FJ 20017 DATA 1,168,189,0,98,157,0,1
MC 20018 DATA 152,157,0,98,189,0,2,1
         68
MC 20019 DATA 189,0,99,157,0,2,152,1
         57
FO 20020 DATA 0,99,232,208,213,88,96
```

C. Regena

The Singing Computer

If a computer can speak and can play music, can it sing? This month, I'll try to make the TI sing. First, to make the computer talk you need the TI Speech Synthesizer, a small peripheral device that attaches to the right side of the console. To use the speech synthesizer, you also need a command module that is made to provide speech.

To do your *own* programming with speech, you also need a command module. Right now the modules available are Speech Editor, TI Extended BASIC, and Terminal Emulator II. Terminal Emulator II is the easiest to work with because you can type any word in and the computer will pronounce it phonetically. Speech Editor and Extended BASIC use CALL SAY commands and have limited vocabularies.

I've had several letters from people wondering why certain phrases don't work. To make the computer say a phrase, such as Texas Instruments, use the number sign (SHIFT 3) before the phrase. For example, CALL SAY ("#Texas Instruments").

Unlimited Speech

A bit of history here—the original speech synthesizer was designed to use the words in the Speech Editor and Extended BASIC lists. Inserts were going to be made available that had different vocabularies (that's why some of the speech synthesizers have a lift-up lid). Then the Terminal Emulator II command module was invented, which provides unlimited speech, and inserts to the synthesizer were no longer needed.

Extended BASIC has also gone through at least one revision. I assume there are very few of the original version around because most users exchanged the original module for the second version as soon as they could. The first version did not support repeating keys and was notorious for "locking up" the computer. There were also some problems with using IMAGE statements.

The Terminal Emulator II command module has a dual purpose. In fact, it's called Terminal Emulator II because it is used to make your TI act as a terminal for another computer. For telecommunications you can use your TI-99/4A with an RS-232 Interface and a telephone modem, plus the Terminal Emulator II command module.

Pages 33–42 of the Terminal Emulator II instruction manual describe how to use speech. There are two main ways to use speech, "text-to-speech" and allophone speech. I use the text-to-speech method because all you have to do is spell the text phonetically. The allophone speech can be more exact because you can specify certain sounds. The manual contains a list of allophone numbers with their sounds plus a few sample programs of how to use this method.

Singing Requires Experimentation

Working with speech in a program takes a lot of experimentation. First, you need to try different spellings to get the computer to properly say what you want it to say. Then you can try different inflection symbols, ^, _, and >. These are used to change inflections and stress points, but they can also change the tone of the voice. You can also add different pause symbols for different sounds and contours. These symbols are the comma, period, semicolon, colon, exclamation point, question mark, and space. Finally, you can alter the pitch and slope—this is what I do to make the computer sing.

To create speech, you need the following statement:

```
OPEN #1:"SPEECH",OUTPUT
```

You may use any number after the number sign, just as in opening other types of files. Later, when you want the computer to speak, just use a command such as

PRINT #1:"MY NAME IS SINNDY."

The pitch is how high or low the voice sounds and can be a number from 0 through 63. Zero is a whisper, 1 is the highest pitched voice, and 63 is the lowest pitched voice. The slope is the rate at which the pitch changes in a spoken phrase. The slope may be a number from 0 through 255. For the best results, the manual recommends a slope 3.2 times the pitch. There are certain combinations of pitch and slope that will not be accepted. The default values of pitch and slope are 43 and 128. To change the pitch and slope, use the format *//xx yyy* where *xx* is the pitch period and *yyy* is the slope level. There must be a space between the numbers. An example in a program statement would be:

PRINT #1:"//30 96"

Changing The Pitch

The following is a sample program that illustrates how the pitch and slope change the sound of the voice. I am trying different pitches from 0 to 63 (and STEPping by 2 so it won't take forever). The slope S is calculated by taking the recommended factor of 3.2 times the pitch. Remember, you may try different slopes if you prefer. B$ combines the double slashes with the pitch, a space, and the slope, so line 170 can set the pitch and slope. Line 180 then speaks the phrase.

```
100 REM PITCH AND SLOPE
110 CALL CLEAR
120 OPEN #5:"SPEECH",OUTPUT
130 FOR P=0 TO 63 STEP 2
140 S=INT(P*3.2+.5)
150 B$="//"&STR$(P)&" "&STR$(S)
160 PRINT B$
170 PRINT #5:B$
180 PRINT #5:"TRY THIS TEST."
190 NEXT P
200 END
```

Since other statements can be executed while a sound is playing, you can play a tone, then say a word. By changing the pitch and slope numbers for the speech, you can make the voice go higher or lower, and program a singing computer.

Remember—I mentioned that working with speech involves a lot of experimentation. Singing takes even more time because there are many parameters that vary with each new tone. After you change the pitch and slope, you can try the inflection symbols and the punctuation marks to vary the voice even more. The TI with Terminal Emulator II can really create synthesized speech that sounds pretty good.

Teaching The ABCs

"Alphabet Song" illustrates simple singing on the computer. However, I did not spend a lot of time fiddling with the program and trying different things to make the speech sound better. You may want to try spelling out the letter as a word, and you may want to add the inflection symbols and punctuation marks. I used different pitches for the singing, but kept the slope numbers just 3.2 times the pitch. You could vary these numbers to get a more human sound and a better singing voice.

My little boy has played a lot with the Early Learning Fun command module. One section teaches the letters of the alphabet, and the child finds the letters on the keyboard. My son is quite proficient at this and knows the names of the letters, but I realized he'd learned them in a random order. Most children learn the alphabet from the ABC song, but I had never sung it to him. I decided I'd let the computer sing it to him.

Lowercase letters are used in the program because my son already knows the capital letters and really needs a little more practice with the lowercase letters. Schoolteachers often recommend learning the lowercase letters right along with the capital letters, and all beginning reading is in lowercase letters.

Lines 120–200 define the lowercase letters. If you have saved the lowercase letters program from my August 1983 column, you can load that program, delete the PRINTing lines, then continue typing this program. If you have any problems running this program, the most likely cause is in typing the data in lines 160–200. Your actual error message will cite line 130 or line 140, but those lines are dependent upon the DATA statements. Do not type a comma at the end of a line.

Extra Option

To hear the singing you will need the TI Speech Synthesizer and the Terminal Emulator II command module. When you turn on the computer with the module plugged in, press any key to start, then press 1 for TI BASIC and program as usual. To run the program without speech, you can select option 2 when the program starts. In this case, you don't need the module or the speech synthesizer.

If you choose no speech, the variable SP will equal 2. All the IF SP=2 THEN ... statements skip over commands that require the Terminal Emulator II module. The CALL SOUND statements play the tune. I used only one note; you may add accompaniment if you'd like. After the tone is played, the letter is sung. The CALL HCHAR or CALL VCHAR statements then place the letter on the screen.

Lines 1880–1910 wait after the song is over until the user presses ENTER, then the song is repeated.

If you prefer to save typing time and effort, you can obtain a copy of this program by sending a $3 copying fee, a blank cassette or diskette, and a stamped, self-addressed mailer to:

C. Regena
P.O. Box 1502
Cedar City, UT 84720

Please specify the name of the program and that you need the TI version.

Alphabet Song

```
100 REM ALPHABET SONG
110 CALL CLEAR
120 FOR C=97 TO 122
130 READ C$
140 CALL CHAR(C,C$)
150 NEXT C
160 DATA 3D43818181814330,BCC281818
    181C2BC,3C4280808080423C,000001
    0101010101,3C4281FF8080423C
170 DATA 0609080808080830E,0101010114
    1221C,0000080808080808,00000008,
    0808080808887,8890A0C0A0908884
180 DATA 0808080808080808,7884020200
    2020202,BCC2818181818181,3C4281
    818181423C,8080808080808,0101010101
    0101
190 DATA BCC2818080808080,3C42403C02
    02423C,0000008080808F08,8181818
    18181433D,4141222214140808,0404
    88885050202
200 DATA 8244281028448282,1010202040
    4,7F0204081020407F
210 T=600
220 PRINT TAB(8);"ALPHABET SONG"
230 PRINT : : : :"CHOOSE:"
240 PRINT : :"1  WITH SPEECH"
250 PRINT :"TERMINAL EMULATOR 2 REQ
    UIRED"
260 PRINT : :"2  NO SPEECH": :
270 CALL KEY(0,K,S)
280 IF (K<49)+(K>50)THEN 270
290 SP=K-48
300 CALL CLEAR
310 IF SP=2 THEN 340
320 OPEN #1:"SPEECH",OUTPUT
330 PRINT #1:"//43 128"
340 CALL SOUND(T,262,2)
350 IF SP=2 THEN 370
360 PRINT #1:"A"
370 CALL HCHAR(3,3,97)
380 CALL SOUND(T,262,4)
390 IF SP=2 THEN 410
400 PRINT #1:"B"
410 CALL HCHAR(2,7,104)
420 CALL HCHAR(3,7,98)
430 CALL SOUND(T,392,2)
440 IF SP=2 THEN 470
450 PRINT #1:"//30 96"
460 PRINT #1:"C"
470 CALL HCHAR(3,11,99)
480 CALL SOUND(T,392,4)
490 IF SP=2 THEN 510
500 PRINT #1:"D"
510 CALL HCHAR(2,15,100)
520 CALL HCHAR(3,15,97)
530 CALL SOUND(T,440,2)
540 IF SP=2 THEN 570
550 PRINT #1:"//27 86"
560 PRINT #1:"E"
570 CALL HCHAR(3,19,101)
580 CALL SOUND(T,440,4)
590 IF SP=2 THEN 610
600 PRINT #1:"F"
610 CALL HCHAR(2,23,102)
620 CALL HCHAR(3,23,108)
630 CALL SOUND(T*2,392,2)
640 IF SP=2 THEN 670
650 PRINT #1:"//30 96"
660 PRINT #1:"G"
670 CALL HCHAR(3,27,97)
680 CALL HCHAR(4,27,103)
690 CALL SOUND(T,349,2)
700 IF SP=2 THEN 730
710 PRINT #1:"//34 109"
720 PRINT #1:"H"
730 CALL HCHAR(7,6,104)
740 CALL HCHAR(8,6,110)
750 CALL SOUND(T,349,4)
760 IF SP=2 THEN 780
770 PRINT #1:"I"
780 CALL HCHAR(7,10,105)
790 CALL HCHAR(8,10,108)
800 CALL SOUND(T,330,2)
810 IF SP=2 THEN 840
820 PRINT #1:"//36 115"
830 PRINT #1:"J"
840 CALL HCHAR(7,14,105)
850 CALL HCHAR(8,14,108)
860 CALL HCHAR(9,14,106)
870 CALL SOUND(T,330,4)
880 IF SP=2 THEN 910
890 PRINT #1:"K"
900 PRINT #1:"//39 125"
910 CALL HCHAR(7,18,104)
920 CALL HCHAR(8,18,107)
930 CALL SOUND(T/2,294,1)
940 IF SP=2 THEN 960
950 PRINT #1:"L"
960 CALL VCHAR(12,8,108,2)
970 CALL SOUND(T/2,294,3)
980 IF SP=2 THEN 1000
990 PRINT #1:"M"
1000 CALL HCHAR(13,12,110)
1010 CALL HCHAR(13,13,109)
1020 CALL SOUND(T/2,294,2)
1030 IF SP=2 THEN 1050
1040 PRINT #1:"N"
1050 CALL HCHAR(13,17,110)
1060 CALL SOUND(T/2,294,4)
1070 IF SP=2 THEN 1090
1080 PRINT #1:"O"
1090 CALL HCHAR(13,21,111)
1100 CALL SOUND(T*2,262,2)
1110 IF SP=2 THEN 1140
1120 PRINT #1:"//43 128"
1130 PRINT #1:"P"
1140 CALL HCHAR(13,25,98)
1150 CALL HCHAR(14,25,112)
1160 CALL SOUND(T,392,2)
1170 IF SP=2 THEN 1200
1180 PRINT #1:"//30 96"
1190 PRINT #1:"Q"
1200 CALL HCHAR(18,4,97)
1210 CALL HCHAR(19,4,113)
1220 CALL SOUND(T,392,4)
1230 IF SP=2 THEN 1250
1240 PRINT #1:"R"
1250 CALL HCHAR(18,8,114)
```

```
1260 CALL SOUND(T*2,349,2)
1270 IF SP=2 THEN 1300
1280 PRINT #1:"//34 109"
1290 PRINT #1:"S"
1300 CALL HCHAR(18,12,115)
1310 CALL SOUND(T,330,2)
1320 IF SP=2 THEN 1350
1330 PRINT #1:"//36 115"
1340 PRINT #1:"T"
1350 CALL HCHAR(17,16,116)
1360 CALL HCHAR(18,16,108)
1370 CALL SOUND(T,330,4)
1380 IF SP=2 THEN 1400
1390 PRINT #1:"U"
1400 CALL HCHAR(18,20,117)
1410 CALL SOUND(T*2,294,2)
1420 IF SP=2 THEN 1460
1430 PRINT #1:"//39 125"
1440 PRINT #1:"V"
1450 PRINT #1:"//30 96"
1460 CALL HCHAR(18,24,118)
1470 CALL SOUND(T,392,2)
1480 IF SP=2 THEN 1500
1490 PRINT #1:"DUB"
1500 CALL HCHAR(23,10,118)
1510 CALL HCHAR(23,11,119)
1520 CALL SOUND(T,392,4)
1530 IF SP=2 THEN 1550
1540 PRINT #1:"BL"
1550 CALL SOUND(T*2,349,2)
1560 IF SP=2 THEN 1590
1570 PRINT #1:"//34 109"
1580 PRINT #1:"U"
1590 CALL SOUND(T,330,2)
1600 IF SP=2 THEN 1630
1610 PRINT #1:"//36 115"
1620 PRINT #1:"X"
1630 CALL HCHAR(23,15,120)
1640 CALL SOUND(T,330,4)
1650 IF SP=2 THEN 1670
1660 PRINT #1:"Y"
1670 CALL HCHAR(23,19,118)
1680 CALL HCHAR(24,19,121)
1690 CALL SOUND(T*2,294,2)
1700 IF SP=2 THEN 1730
1710 PRINT #1:"//39 125"
1720 PRINT #1:"Z"
1730 CALL HCHAR(23,23,122)
1740 CALL SOUND(T,262,2)
1750 CALL SOUND(T,262,4)
1760 CALL SOUND(T,392,2)
1770 CALL SOUND(T,392,4)
1780 CALL SOUND(T,440,2)
1790 CALL SOUND(T,440,4)
1800 CALL SOUND(T*2,392,2)
1810 CALL SOUND(T,349,2)
1820 CALL SOUND(T,349,4)
1830 CALL SOUND(T,330,2)
1840 CALL SOUND(T,330,4)
1850 CALL SOUND(T,294,2)
1860 CALL SOUND(T,294,4)
1870 CALL SOUND(T*4,262,2)
1880 CALL KEY(0,K,S)
1890 IF K<>13 THEN 1880
1900 CALL CLEAR
1910 GOTO 330
1920 END
```

64 Searcher

John Krause, Assistant Technical Editor
and Michael Jacobi

"64 Searcher" is a time-saving utility that searches through your BASIC program and locates any character or string of characters that you choose. (This is a 64 version of VIC searcher that appeared in COMPUTE!, February 1983.)

When you're working on a long BASIC program, it pays to plan ahead. But it seems that no matter how hard you try, you can't keep track of everything in your program. Can I use H to store the high score, or is that variable already being used for something else? Where is this subroutine called from? You probably end up searching for a certain number or word hidden among scores of program lines.

"64 Searcher" allows you to spend less time searching and more time programming. You simply give it the string of characters to search for and it tells you the numbers of all lines in which the string appears. It can search 100 lines faster than it takes you to search one. It's fast because it's written in machine language. But you don't have to know machine language to use it.

Just LOAD it and RUN it, then LOAD your BASIC program. 64 Searcher doesn't use any BASIC memory, so you can work on your program normally. You can use 64 Searcher at any time by typing 0 followed by the string you want to find enclosed within either slashes or quotes, and hitting RETURN. This stores the string in your program as line 0. If your program already has a line 0, you will have to change that line number because the string must be the first line in the program.

Then type SYS49152 and hit RETURN. Instantly, you should see numbers appear on the screen. These are the line numbers that contain the string you specified. If no match is found, no numbers will be printed. If the string occurs more than once in a line, the line number is printed only once.

Because BASIC commands are stored differently from other characters in a program, there are two ways of specifying the search string. If the string is enclosed within slashes, BASIC commands are recognized as such. If the string is within quotes, it will be treated as a literal string of characters.

For example, to find the BASIC statement AND, line 0 should be:

```
0 /AND/
```

After entering SYS49152, 64 Searcher will find the AND in this line:

```
10 IF X AND Y THEN 50
```

but not in this line:

```
20 PRINT "X AND Y"
```

To find the AND in line 20 above, use quotes instead of slashes.

Remember to delete line 0 before saving or running your program.

64 Searcher

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

```
10 FORI=49152TO49255:READJ:K=K+J:POKEI,J:
   NEXT                              :rem 66
20 IFK<>16302THENPRINT"ERROR IN DATA STAT
   EMENTS":STOP                      :rem 117
30 PRINT"{CLR}SYS49152 TO SEARCH"    :rem 36
100 DATA169,1,133,251,169,8,133,252,160,0
    ,177,251,56,229,251,56           :rem 80
110 DATA233,5,141,104,192,233,2,141,105,1
    92,160,0,177,251,170,200         :rem 142
120 DATA177,251,240,67,133,252,134,251,16
    0,0,177,251,56,229,251,170       :rem 17
130 DATA202,134,2,198,2,165,2,205,104,192
    ,48,222,133,253,173,105          :rem 110
140 DATA192,133,254,164,253,177,251,164,2
    54,217,5,8,208,229,198,253       :rem 45
150 DATA198,254,208,239,160,2,177,251,170
    ,200,177,251,32,205,189,169      :rem 88
160 DATA32,32,210,255,76,26,192,96
                                     :rem 190
```

MACHINE LANGUAGE

Jim Butterfield, Associate Editor

Decimal Mode
Part 2

Decimal mode is quite useful in arithmetic programming such as game scoring and simple accounting. It has other uses, too—for example, in converting binary numbers to decimal for output. It also has certain bugs, pitfalls, and conventions.

Bugs And Pitfalls

Don't depend on the Zero and Negative (Z and N) flags immediately following a decimal addition (ADC) and subtraction (SBC). If you really need them, perform a data transfer (for example, TAX) to insure the flags are set correctly. The Carry flag is correct and has its usual meaning after the addition or subtraction.

Remember that decimal mode uses only the ADC and SBC instructions. The increment and decrement instructions (INX, INY, INC, DEX, DEY, DEC) behave in binary; and comparisons (CMP, CPX, CPY) are based as usual on binary values.

Programmers using machines with interrupt sequences must be careful of decimal mode. The interrupt can clear decimal mode with CLD (Clear Decimal); when the interrupt code finishes with RTI, the status register will be restored and decimal mode will be reinstated if it was in effect before. On Commodore machines, the interrupt sequences do not include a CLD instruction; in this case, the interrupt should be locked out using a SEI (Set Interrupt Disable) before going into decimal mode.

The VIC-20 and Commodore 64 have a useful feature: Registers may be preset before a SYS call. Addresses $030C, $030D, $030E, and $030F (decimal 780 to 783) contain values that will be transferred to registers A, X, Y, and the status register at the time of a SYS. When the machine language program returns to BASIC, these same addresses will contain the contents of the respective register. In other words, we could POKE 780,65 followed by a SYS; and the machine language program would start running with a value of $41 (decimal 65) in the A register.

What does this mean to decimal mode? Here's the possible danger: If the wrong value is contained in address 783, it will be transferred to the status register at the time of a SYS. An uncontrolled value might set decimal mode, or even worse, set the interrupt disable flag. To make things worse, these flags will not be restored when we return to BASIC. They will be neatly stored in 783, but BASIC will resume with the flags in an unworkable state. There goes BASIC.

It's probably wise to leave address 783 alone. If it worries you, POKE 783,0 before giving a SYS command.

Conventions

We can handle fractions in decimal arithmetic. It's best to do this by using an "assumed decimal point." In other words, we will work dollar values as an integer number of pennies, and kilometers as integer values of meters. It's easier to stick in the decimal point at output time.

Negative numbers are a little tricky. We can use a scheme similar to that in binary numbers: That is, the "high bit" of a number represents the sign. This, however, splits positive and negative unevenly: A two-byte number will range from a low of −2000 (value 8000) up to +7999. If you use this method, don't forget that the N flag isn't dependable after an addition or subtraction and that you'll need to take an extra step to test the flag.

A better technique is called "tens complement" and it's been used in many household devices such as counters on tape recorders. We understand that a reading of 9994 really means −6. If we want to use this technique, we might

choose to try to split positive and negative more evenly, so that a two-byte number would range from −5000 to +4999. In this case, we must remember not to use the N bit, but instead compare the high byte to 50 hex. If it is higher, the number is negative.

If "tens complement" is used, remember to invert a negative number at the time of printing. I find that the easiest way to do this is to subtract it from 0000 so that 9993 becomes 0007.

Multiplication

To multiply two decimal numbers we are almost forced to resort to repeated addition. As we go from one decimal digit to the next, we must "shift" either the multiplier or the product: This is a binary shift-four-places. It's awkward and we can quickly see why binary is preferred.

There's an elegant way to multiply a decimal number by a binary value, or by a fixed amount. We can use what I call a "decimal shift."

A binary shift multiplies a number by two. We can do the same thing with a decimal number by adding it to itself. Thus, to multiply by two we add the number to itself (in decimal mode). To multiply by four we multiply by two, twice. To multiply by five, we multiply by four and add the original number.

A Multiplication Example

We'll have the computer (PET, VIC, or 64) output a table of multiples of the number 5. (Two would be too easy.)

```
                ; set value to one
033C A2 01              LDX   #$01
033E 8E B0 03           STX   LOW
0341 CA                 DEX
0342 8E B1 03           STX   MED
0345 8E B2 03           STX   HIGH
0348 8E B6 03           STX   COUNT
                ; copy the number
034B A0 02      LOOP    LDY   #$02
034D B9 B0 03     CP    LDA   LOW,Y
0350 99 B3 03           STA   COPY,Y
0353 88                 DEY
0354 10 F7              BPL   CP
                ; multiply by four
0356 A2 02              LDX   #$02
0358 18           FP    CLC
0359 A0 FD              LDY   #$FD
035B 78                 SEI
035C F8                 SED
035D B9 B3 02     TP    LDA   HIGH-255,Y
0360 79 B3 02           ADC   HIGH-255,Y
0363 99 B3 02           STA   HIGH-255,Y
0366 C8                 INY
0367 D0 F4              BNE   TP
0369 CA                 DEX
036A D0 EC              BNE   FP
                ; add original value
036C A0 FD              LDY   #$FD
036E 18                 CLC
036F B9 B3 02     AP    LDA   HIGH-255,Y
0372 79 B6 02           ADC   COPY-253,Y
0375 99 B3 02           STA   HIGH-255,Y
0378 C8                 INY
0379 D0 F4              BNE   AP
037B D8                 CLD
037C 58                 CLI
                ; print the number
037D A0 02              LDY   #$02
037F B9 B0 03     LP    LDA   LOW,Y
0382 4A                 LSR   A
0383 4A                 LSR   A
0384 4A                 LSR   A
0385 4A                 LSR   A
0386 09 30              ORA   #$30
0388 20 D2 FF           JSR   $FFD2
038B B9 B0 03           LDA   LOW,Y
039E 29 0F              AND   #$0F
0390 09 30              ORA   #$30
0392 20 D2 FF           JSR   $FFD2
0395 88                 DEY
0396 10 E7              BPL   LP
                ; print RETURN and loop
0398 A9 0D              LDA   #$0D
039A 20 D2 FF           JSR   $FFD2
039D EE B6 03           INC   COUNT
03A0 AE B6 03           LDX   COUNT
03A3 E0 07              CPX   #$08
03A5 D0 A4              BNE   LOOP
03A7 60                 RTS
```

Note the peculiar addressing in lines 035D to 0363 and again in 036F to 0375. We need to have a positive-incrementing index (in this case Y), since we must start our addition at the low-order value, LOW, and work upwards. We cannot use the obvious method of starting at zero and testing to see when we have done all three values, because we want the carry flag to be preserved; CPY (Compare Y) would destroy the previous value of the carry and our addition wouldn't work right.

If you'd rather enter the program from BASIC, here's the same program in DATA statements. It will work on all Commodore machines.

Multiples Of 5

```
100 DATA 162,1,142,176,3,202,142,177,3
110 DATA 142,178,3,142,182,3,160,2
120 DATA 185,176,3,153,179,3,136,16,247
130 DATA 162,2,24,160,253,120,248,185,179
    ,2
140 DATA 121,179,2,153,179,2,200,208,244,
    202
150 DATA 208,236,160,253,24,185,179,2,121
    ,182,2
160 DATA 153,179,2,200,208,244,216,88,160
    ,2
170 DATA 185,176,3,74,74,74,74,9,48,32,21
    0,255
180 DATA 185,176,3,41,15,9,48,32,210,255,
    136,16
190 DATA 231,169,13,32,210,255,238,182,3,
    174,182,3
200 DATA 224,8,208,164,96
300 FOR J=828 TO 935
310 READ X:T=T+X
320 POKE J,X
330 NEXT J
340 IF T<>13479 THEN STOP
350 SYS 828
```

64 EXPLORER

Larry Isaacs

This month we will continue our look at printing characters to a bitmapped display. Last month we looked at a method which transferred a character dot pattern to the bitmapped display. This month we will look at a second method, which *draws* the characters.

Printing Bit By Bit

With the appropriate set of line segments, virtually any character shape can be drawn. The characters do not necessarily have to look like the standard ASCII character set. In addition, you are not restricted to a fixed character cell. Each character can be as complex and as large as you like. For this flexibility, you do lose a few advantages offered by the use of character dot patterns. (It becomes a little more difficult to print in reverse video and will likely take a little longer to print the character when characters are drawn rather than transferred.)

With the drawing method, we will need to make use of a line-drawing routine. For convenience, I will be using the machine language line-drawing routines presented in the May issue of COMPUTE!. However, for use in the example BASIC program which follows, almost any line-drawing routine will suffice. (The one found in COMPUTE!'s earlier "SuperBASIC 64" program could be used if you desire. Some minor modifications to the BASIC program will be required, though.)

To draw a given character in the bitmapped display, we will need some data to define how the character should be drawn. Unlike the transfer method, where the format for such data is already fixed, here we have total freedom to define our own format. The format must specify what line segments should be drawn to form the characters. This means that the data must define the starting and ending coordinates of each line segment. Another thing to note is that the data will need to define these coordinates relative to the previous coordinates. By specifying the next point based on the previous point, the character can be drawn anywhere in the bitmapped display.

To simplify the following discussion, I will use the term "vector" to refer to the line segments which make up a character. Also, I will use the term "vector string" for the data which defines how to draw a character.

One way we could define the format of the data in the vector string is to specify each vector with two pairs of relative coordinates. A single byte could be used for each relative coordinate, which could represent a value from 127 to −128. Thus, four bytes would be required for each vector in the vector string.

Moving Points

As I mentioned in the May column, I prefer to have the *draw* function continue from each previous endpoint. This eliminates the need to specify a new beginning point every time. The catch is that there must be some way of moving the last endpoint without drawing. Assuming we define a way of moving the endpoint for our vector string, then it will be possible to specify a vector using one pair of relative coordinates, rather than two. For this to be an advantage, a fair percentage of the vectors would need to draw from the end of the previous vector. When creating characters from vectors, I believe this will generally be true.

If the characters are not going to be that large, there is another phenomenon: Most of the vectors will be fairly short. Assuming they are typically short enough, we could save more bytes by using one byte to specify a vector. The byte could be split into an upper and lower four bits, with each half able to represent a relative coordinate of 7 to −8. This may not seem like very much, but if the vector isn't too long to be represented by two of these bytes, we haven't lost anything.

Vector Bytes

This isn't the only way to use a single byte to specify a vector. The byte could be split into two parts so that one part specifies a direction and the other a distance. The direction in this case would most likely be a multiple of 45 degrees. This actually works quite well for drawing characters. However, I will go with the format of putting relative coordinates into the byte. I will refer to such bytes as "vector bytes" in the discussion which follows.

Given that the vector has a limited range, we will need to define some way of invoking

exceptions to handle the times when the range is exceeded. Also, we still need to define a way of moving instead of drawing, which we will also treat as an exception. One way to do this is to use one of the coordinate values to signal the exception. The other relative coordinate could be used to indicate which exception. Since this uses both halves of the vector byte, the exceptions will require additional bytes.

Now we are ready to get down to specifics. Let's try putting the relative coordinate for X in the upper half of the vector byte. Naturally that means putting the relative coordinate for Y in the lower half. As for a value to signal exceptions, it is most logical to choose a value at an extreme. Since our range is from 7 to −8, −8 would be the best choice. It also would be best to have this value in the upper half of the vector byte. This would cause the exception bytes to fall in the range of 128 to 143. Bytes outside this range will be regular "drawing" vector bytes.

There are four exceptions we will need to deal with initially. These exceptions are for signaling a move, an extended draw, an extended move, and the end of the vector byte string. With the upper half signaling an exception, this leaves the lower half to flag the exceptions.

Also, the numbering of the exceptions will be a little easier if we treat the four bits as unsigned rather than signed. This lets us have values from 0 to 15, instead of 7 to −8. For the exceptions, let's try values of 0, 1, and 2 to select move, extended draw, and extended move, respectively. To mark the end of a vector byte string, let's try 15, to choose an extreme again. The following table summarizes these choices:

Data Formats For Vector Byte String

Byte No.	Bit 7 6 5 4 3 2 1 0			
Vector Byte				
1	DX	DY		DX = 7 to −7 DY = 7 to −8
Move Exception				
1	−8	0		= 128
2	DX	DY		DX,DY = 7 to −8
Extended Draw Exception				
1	−8	1		= 129
2	DX			= 127 to −128
3	DY			
Extended Move Exception				
1	−8	2		= 130
2	DX			= 127 to −128
3	DY			= 127 to −128

Drawing Strings Of Characters

Now we are ready to implement the above in a BASIC program. The result is shown in Program 1. The task of drawing the character has been split into a number of routines. One routine fetches the next vector byte, and another unpacks the relative distances. There is also a separate routine to handle vector byte draws, vector byte moves, extended draws, and extended moves. Finally, there is a routine which draws a character, and which in turn uses that routine to draw a string of characters.

The vector byte data included in Program 1 defines vector byte strings for characters 65 through 90, or A through Z. The vector byte strings will draw the ASCII character corresponding to the character code. The space character is also defined. The vector byte strings are stored in the string array VB$ and are accessed by using the character code as the subscript into the array. Prior to running the program, it will be necessary to load the line-drawing routines presented in the May column.

Character Rotation

I have included a routine which lets you specify a character rotation in increments of 90 degrees. For drawn characters, the rotation involves simply negating or swapping the relative coordinates specified in a vector byte. Rotating the transferred character pattern is not too difficult provided the cell is square, as it is in our case. Rotating the character to other angles typically won't produce desirable-looking characters, and may be too complex to implement.

The following is a table showing the routines that are available, and what their start address is. define the base location of the second jump table. Here is a list of the routines:
Since this will be the second jump table (to complement the line-drawing jump table), I use J2 to

Loc.	Description
J2+0	SET PUT CHAR. DATA LOCATION
J2+3	PUT CHARS. IN BITMAP (TRANSFER METHOD)
J2+6	SET DRAW CHAR. DATA LOCATION
J2+9	DRAW CHARS. IN BITMAP
J2+12	SET ROTATION
J2+15	NOT USED YET
J2+18	NOT USED YET
J2+21	NOT USED YET

The jump vector location of these routines is shown as the variable J2 plus an offset. To obtain the actual address, J2 should be set to the base of the jump table which is 50176 or C400 hex. The following list gives the syntax for using each of the defined routines in the jump table:

```
SYS  J2,LOC
SYS  J2+3,CHAR or STRING
```

```
SYS   J2+6,LOC
SYS   J2+9,CHAR or STRING
SYS   J2+12,ROT
      ROT: 0=NO ROT.,  1=90 DEGREES
      2=180 DEG., 3=270 DEG.
```

Both the put character (J2+3) and draw character (J2+9) will accept either a single character or a string of characters as an argument. If the argument supplied is a numeric value, it will be interpreted as the ASCII value of a single character. If the argument is a string, the entire string will be printed.

The location required by the put character routine should be the base address of the character dot patterns to use. The location required by the draw character routine is the base address of a 256-byte table containing pointers to 128 vector byte strings. The pointers to the vector byte strings are each two bytes, stored as low byte followed by high byte. Use of a table is necessary because the length of each string may vary, making it impossible to calculate the locations of the vector byte strings directly.

Safe Entry

Program 2 will POKE the machine code for the character routines into the proper locations. Like the program which POKEs the line-drawing routines, that last number in each data line is the sum of the previous eight bytes on the line. Provided you don't make two errors which cancel each other, the program will report any lines that have mistakes in them. If there are no detected errors, a SUCCESSFUL LOAD is reported.

Program 3 provides a simple illustration of the use of the character routines. For vector byte data, add the DATA statements shown in Program 1, which will define ASCII characters A through Z, and space. The vector byte data will be placed at the top of BASIC's free RAM, after 1024 bytes of space is reserved from BASIC. You will be able to see the increase in speed over the BASIC routines.

Program 1: Draw Characters In A Bitmap

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

```
10 REM DRAW CHARACTERS IN BIT-MAP:rem 212
20 JT=49152:REM DRAWING JUMP TABLE:rem 16
30 DIM VB$(256):REM DIM. STRING ARRAY
                                    :rem 99
40 X=0:Y=0:REM X,Y LOCATION        :rem 125
50 DX=0:DY=0:REM DELTA-X,DELTA-Y   :rem 219
60 VB=0:REM VECTOR BYTE            :rem 144
70 VB$="":REM VECTOR BYTE STRING   :rem 160
80 VP=0:VL=0:REM PTR INTO VB$, VB$ LEN
                                    :rem 149
90 GOTO 1000                        :rem 99
100 REM GET NEXT VECTOR BYTE       :rem 155
110 VP=VP+1:REM BUMP POINTER       :rem 234
120 IF VP>VL THEN VB=0:RETURN      :rem 246
130 VB=ASC(MID$(VB$,VP,1)):REM GET BYTE
                                    :rem 253
140 RETURN                         :rem 117
200 REM UNPACK VECTOR BYTE          :rem 63
210 DY=VBAND15:IF DYAND8 THEN DY=DYOR-8
                                    :rem 91
220 DY=VBAND15:IF DYAND8 THEN DY=DYOR-8
                                    :rem 92
230 DX=INT(VB/16):IF DXAND8 THEN DX=DXOR-
    8                              :rem 242
240 RETURN                         :rem 118
300 REM EXECUTE VECTOR BYTE DRAW   :rem 191
310 X=X+DX:Y=Y+DY                   :rem 57
320 SYS JT+18,X,Y:REM DRAW THE BYTE
                                    :rem 48
330 RETURN                         :rem 118
400 REM EXECUTE VECTOR BYTE MOVE   :rem 201
410 GOSUB 100:GOSUB 200:REM GET NEXT
                                    :rem 47
420 X=X+DX:Y=Y+DY                   :rem 59
430 SYS JT+12,X,Y:REM DO THE MOVE:rem 148
440 RETURN                         :rem 120
500 REM GET EXTENDED DX AND DY     :rem 182
510 GOSUB 100:DX=VB:REM EXTENDED DX
                                    :rem 93
520 IF DX AND 128 THEN DX=DX OR -128
                                    :rem 61
530 GOSUB 100:DY=VB:REM EXTENDED DY
                                    :rem 97
540 IF DY AND 128 THEN DY=DY OR -128
                                    :rem 66
550 RETURN                         :rem 122
600 REM EXECUTE EXTENDED DRAW       :rem 12
610 GOSUB 500:REM GET DX,DY         :rem 15
620 X=X+DX:Y=Y+DY                   :rem 61
630 SYS JT+18,X,Y:REM DO THE DRAW:rem 147
650 RETURN                         :rem 123
700 REM EXECUTE EXTENDED MOVE       :rem 22
710 GOSUB 500:REM GET DX,DY         :rem 16
720 X=X+DX:Y=Y+DY                   :rem 62
730 SYS JT+12,X,Y:REM DO THE DRAW:rem 142
750 RETURN                         :rem 124
800 REM DRAW STRING OF VECTOR BYTES
                                    :rem 112
810 VP=0:VL=LEN(VB$):IF VL=0 THEN RETURN
                                    :rem 152
820 IF VP>=VL THEN RETURN          :rem 251
830 GOSUB 100:REM GET NEXT VB      :rem 129
840 GOSUB 200:REM UNPACK           :rem 142
850 IF DX<>-8 THEN GOSUB 300:GOTO 820
                                    :rem 246
860 ON DY+1 GOSUB 400,600,700      :rem 205
870 GOTO 820                       :rem 114
900 REM PRINT P$                   :rem 126
910 FOR PP=1 TO LEN(P$)            :rem 214
920 VB$=VB$(ASC(MID$(P$,PP,1)))    :rem 181
930 GOSUB 800:REM DRAW THE CHAR.    :rem 45
940 NEXT:RETURN                    :rem 246
1000 REM MAIN ROUTINE              :rem 240
1010 GOSUB 10000                    :rem 51
1020 SYS JT:SYS JT+6,0:SYS JT+9,6,14
                                    :rem 34
1030 X=10:Y=100:SYS JT+12,X,Y      :rem 137
1040 FOR CH=64 TO 90               :rem 234
1050 VB$=VB$(CH):GOSUB 800         :rem 169
1060 NEXT                           :rem 6
1070 X=10:Y=80:SYS JT+12,X,Y       :rem 100
1080 P$="THIS IS AN EXAMPLE OF"    :rem 194
1090 GOSUB 900                     :rem 227
1100 X=10:Y=60:SYS JT+12,X,Y        :rem 92
1110 P$="PRINTING WITH VECTOR BYTES"
                                    :rem 185
1120 GOSUB 900                     :rem 221
9000 GET Z$:IF Z$="" THEN 9000     :rem 231
```

```
9010 SYS JT+3                                :rem 197
9020 END                                     :rem 162
10000 REM LOAD VB$()                          :rem 2
10010 C=0:READ CH:IF CH<0 THEN RETURN
                                             :rem 82
10020 READ VB:IF C>0 THEN C=C-1:GOTO 1008
      0                                      :rem 221
10030 IF ABS(VB)>7 THEN 10060                :rem 221
10040 READY:VB=(VB*16+(DYAND15)):rem 137
10050 GOTO 10080                             :rem 40
10060 IF VB=143 THEN 10010                   :rem 20
10070 IF VB<>128 THEN C=2                    :rem 21
10080 VB$(CH)=VB$(CH)+CHR$(VB AND 255)
                                             :rem 234
10090 GOTO 10020                             :rem 38
11100 REM @                                  :rem 23
11110 DATA 64,128,3,3,0,1,-1,0,0,-1,3,0
                                             :rem 61
11120 DATA 0,2,-1,1,-3,0,-1,-1,0,-4
                                            :rem 105
11160 DATA 128,-5,2,5,0,128,3,-2,143
                                             :rem 204
11130 DATA 1,-1,3,0,1,1,128,3,-1,143
                                             :rem 180
11140 REM A                                  :rem 28
11150 DATA 65,0,4,2,2,1,0,2,-2,0,-4
                                             :rem 128
11170 REM B                                  :rem 32
11180 DATA 66,0,6,4,0,1,-1,0,-1,-1,-1
                                             :rem 220
11190 DATA -4,0,128,0,-3,4,0,1,1,0,1
                                             :rem 177
11200 DATA -1,1,128,4,-3,143                 :rem 64
11210 REM C                                  :rem 28
11220 DATA 67,128,5,5,-1,1,-3,0,-1,-1
                                             :rem 233
11230 DATA 0,-4,1,-1,3,0,1,1                 :rem 42
11240 DATA 128,3,-1,143                      :rem 90
11250 REM D                                  :rem 33
11260 DATA 68,0,6,3,0,2,-2,0,-2,-2,-2
                                             :rem 225
11270 DATA -3,0,130,8,0,143                  :rem 21
11280 REM E                                  :rem 37
11290 DATA 69,0,6,5,0,128,-1,-3,-4,0
                                             :rem 199
11300 DATA 128,0,-3,5,0,128,3,0,143
                                             :rem 149
11310 REM F                                  :rem 32
11320 DATA 70,0,6,5,0,128,-5,-3,4,0
                                             :rem 144
11330 DATA 128,4,-3,143                      :rem 93
11340 REM G                                  :rem 36
11350 DATA 71,128,5,5,-1,1,-3,0,-1,-1
                                             :rem 232
11360 DATA 0,-4,1,-1,3,0,1,1,0,2,-2,0
                                             :rem 207
11370 DATA 128,5,-3,143                      :rem 98
11380 REM H                                  :rem 41
11390 DATA 72,0,6,128,0,-3,5,0,128,0,3
                                             :rem 45
11400 DATA 0,-6,128,3,0,143                  :rem 21
11410 REM I                                  :rem 36
11420 DATA 73,128,0,6,2,0,128,-1,0,0,-6
                                             :rem 83
11430 DATA 128,-1,0,2,0,128,3,0,143
                                             :rem 148
11440 REM J                                  :rem 40
11450 DATA 74,128,0,1,1,-1,2,0,1,1,0,5
                                             :rem 24
11460 DATA 128,-1,0,2,0,128,3,-6,143
                                             :rem 202
11470 REM K                                  :rem 44
11480 DATA 75,0,6,128,0,-4,4,4,128,-3,-3
                                             :rem 145
11490 DATA 1,0,3,-3,128,3,0,143              :rem 215
11500 REM L                                  :rem 39
11510 DATA 76,0,6,128,0,-6,5,0,128,3,0,14
      3                                      :rem 242
11520 REM M                                  :rem 42
11530 DATA 77,0,6,3,-3,3,3,0,-6,128,3,0,1
      43                                     :rem 26
11540 REM N                                  :rem 45
11550 DATA 78,0,6,5,-5,128,0,5,0,-6
                                             :rem 161
11560 DATA 128,3,0,143                       :rem 49
11570 REM O                                  :rem 49
11580 DATA 79,128,1,0,-1,1,0,4,1,1,3,0,1,
      -1                                     :rem 8
11590 DATA 0,-4,-1,-1,-3,0,128,7,0,143
                                             :rem 29
11600 REM P                                  :rem 44
11610 DATA 80,0,6,4,0,1,-1,0,-1,-1,-1
                                             :rem 214
11620 DATA -4,0,130,8,-3,143                 :rem 69
11630 REM Q                                  :rem 48
11640 DATA 81,128,1,0,-1,1,0,4,1,1,3,0,1,
      -1                                     :rem 254
11650 DATA 0,-4,-1,-1,-3,0,128,2,1:rem 82
11660 DATA 2,-2,128,3,1,143                  :rem 28
11670 REM R                                  :rem 53
11680 DATA 82,0,6,4,0,1,-1,0,-1,-1,-1,-4,
      0                                      :rem 200
11690 DATA 128,2,0,3,-3,128,3,0,143
                                             :rem 161
11700 REM S                                  :rem 48
11710 DATA 83,128,0,1,1,-1,3,0,1,1,0,1,-1
      ,1                                     :rem 251
11720 DATA -3,0,-1,1,0,1,1,1,3,0,1,-1
                                             :rem 205
11730 DATA 128,3,-5,143                      :rem 98
11740 REM T                                  :rem 53
11750 DATA 84,128,2,0,0,6,128,-2,0,4,0
                                             :rem 45
11760 DATA 128,3,-6,143                     :rem 102
11770 REM U                                  :rem 57
11780 DATA 85,128,0,6,0,-5,1,-1,3,0,1,1
                                             :rem 83
11790 DATA 0,5,128,3,-6,143                  :rem 38
11800 REM V                                  :rem 52
11810 DATA 86,128,0,6,0,-4,2,-2,2,2,0,4
                                             :rem 82
11820 DATA 128,3,-6,143                      :rem 99
11830 REM W                                  :rem 56
11840 DATA 87,128,0,6,0,-6,3,3,3,-3,0,6
                                             :rem 94
11850 DATA 128,3,-6,143                     :rem 102
11860 REM X                                  :rem 60
11870 DATA 88,0,1,4,4,0,1,128,-4,0,0,-1
                                             :rem 83
11880 DATA 4,-4,0,-1,128,3,0,143              :rem 9
11890 REM Y                                  :rem 64
11900 DATA 89,128,0,6,2,-2,2,2,128,-2,-2
                                            :rem 141
11910 DATA 0,-4,128,5,0,143                  :rem 27
11920 REM Z                                  :rem 59
11930 DATA 90,128,0,6,4,0,0,-1,-4,-4,0,-1
                                            :rem 168
11940 DATA 4,0,128,4,0,143                   :rem 240
11950 REM SPACE                              :rem 80
11960 DATA 32,130,8,0,143                   :rem 196
11970 DATA -1                               :rem 122
```

Programs 2 and 3 will appear in this column next month. ©

Are Computers A Home Appliance?

Fred D'Ignazio, Associate Editor

Necessary, Easy, And Inexpensive

In recent columns I have written about a growing consumer awareness that things are not right with the microcomputer industry. Some misleading advertisements have made people buy computers as a home appliance. Unfortunately, the computers have not met some people's expectations, and then ended up gathering dust in the closet.

To be a legitimate home appliance, a product should have three characteristics:

It should be inexpensive.

It should meet a real need.

It should be easy to use.

Let's look closely at each characteristic, and see how computers measure up.

A home appliance should be inexpensive. A low-end computer often appears to be inexpensive, but it turns out to be costly after a person adds the necessary "extras," including a disk drive, a printer, and some basic software.

A home appliance should meet a real need. For example, people use telephones to communicate; TVs for entertainment and news; ovens to cook food; and refrigerators to keep food fresh. But what do people need computers for?

A home appliance should be easy to use. For example, you can pick up a phone, dial seven numbers, and reach another person within seconds. You can push a button on a TV, and the world enters your living room. You can pull down a lever on the toaster oven and get a hot biscuit.

When you turn on the computer, it says, "READY." But it is not really ready. First you must load in additional software, turn on additional appliances (disk drives, a printer, a modem, etc.), answer questions, and type in additional information. All these cumbersome, time-consuming steps make the computer ready, but they do not make it easy to use.

WASH! Magazine

How do people learn how to use computers?

They might join a user group, ask a kid, or read a computer magazine.

A magazine like COMPUTE! can be a lifesaver for the consumer who has just bought an inexpensive computer. The magazine offers easy-to-read tutorials, practical tips, and lots of excellent, affordable software.

Kids can also be helpful. So can user groups. But all this is beside the point. The real question is: Should a home appliance be this difficult to use?

To put this question in perspective, ask yourself how many people would own a washing machine if, to operate it, they had to buy a monthly magazine called *WASH!* , and they had to get help from a washing-machine whiz kid and attend weekly meetings of the Whirlpool User Group?

And how fair is it to our children to assume that they will know how to use a machine that has us puzzled and bewildered?

It is easy for kids to get *intimate* with computers, because they share few of our fears, anxieties, and prejudices about these machines. But it is not nearly as easy for them to get computer *literate*—to be competent computer users and programmers. Nevertheless, we adults now

have the misconception that all children take to computers as naturally as ducks to water. But what if our children *don't* take to computers? Does that make them less intelligent or less able than their friends? And where does that leave us?

A Growing Backlash

When millions of people buy a computer, take it home, then discover that it is not going to be inexpensive, that it meets no immediate need, and that it is not always easy to use, how do they feel? Whom do they blame?

Until recently, most people blamed themselves, their families, and their kids. But this is beginning to change. Too many people have been disappointed by computers, and they are talking to their neighbors. The secret is finally out. The fault is not with the consumer. It is with computers themselves—and the companies that make them.

New Consumer Savvy

The computer price wars of 1982–1983 had a disastrous effect on the computer industry and drove many companies out of the market, including Texas Instruments; Mattel, and Timex. In addition, many naive customers were lured by incredibly low prices into buying low-end computers. Unfortunately, the customers had no idea what to do with the computers once they got them home.

However, in spite of these setbacks, the ultimate effect of the price wars may be positive. Between 1982 and 1984, large numbers of people bought "throwaway" computers, became disgruntled consumers, and described their experiences to their neighbors. The result is that, today, people are a lot more knowledgeable about computers than they were just a year ago.

In fact, people's bad experience with computers and their "sour grapes" reaction have created a mild consumer backlash against computers. The average consumer, in mid-1984, is much more skeptical about computers than he was in 1982 or 1983. He realizes that a good price is not the only thing to look for when choosing a computer for the home. He understands that computers, to be useful, need good software, memory, printers, and disk drives. He realizes that even with all this equipment a computer is *not* a home appliance. On its own it won't guarantee him or his family anything.

The average consumer is returning to the healthier show-me attitude that prevailed before the era of *high-tech chic* that reigned from 1982 and 1984. "Show me real needs that computers meet," the consumer is saying. "Show me a computer with no hidden costs that is useful and simple to operate." ©

64 Error Suppression

Tom Nuss

There are times when you don't want error messages (and the resulting interruption) in a program. Here's how to avoid some kinds of system freezes.

While constructing a general graphing program that would handle varied equations, I realized that it would crash when it tried to divide by zero or take the square root of a negative number. Since a graphing program depends on drawing a fairly smooth curve, these two possibilities would definitely occur from time to time in a general program loop that plots. I also found I had to learn machine language to get things to happen before my hair turned gray.

After delving into the BASIC interpreter with my trusty Supermon-64, I discovered that if the accumulator contained a zero after a division, it would branch to the error routine at $A437, which would jump indirectly (via a vector address in $300 and $301) to $E38B and proceed to print messages and stop the BASIC program.

I soon confirmed that all error messages (at least the ones I tried) went through $A437, then jumped indirectly by the bytes loaded in $300 and $301 to $E38B. All I had to do was change the contents of $300 and $301 to an address pointing to where I would have my own routine that would skip over the error to allow the program to continue.

Back To BASIC

Simple enough, but how to get back in the BASIC program at the right spot once I left? No, not back into the interpreter again; anything but that!

I noticed that just down the page of the memory map at $A906 was the routine "Scan for next statement." Now that I look back on it, I should have started there. Needless to say, that entry point was the last of the major pieces to the puzzle of skipping the BASIC error handler.

To get down to the mechanics, I have made up a demonstration program to illustrate the method to bypass arithmetic errors. If you type in the program and RUN, you will get the results shown in the figure. AC SR XR YR are the Accumulator, Status, X, and Y registers respectively. Directly beneath them are several series of four numbers and either SYNTAX OK or a variation of F(#)= a number.

Ignoring the first line of eights and SYNTAX OK for the moment, you will notice that line 70 in the program defines the function SQR(4–C↑2)/C and that the second line of numbers in the figure is 255 49 14 1. When one has C=–3 in the above equation, mathematicians will shake their heads, and the computer should crash. Why didn't it? In fact it even gave an answer for SQR(–5)/–3 and blithely continued to calculate the rest of the F(C)s from –3 to 3. F(–2), F(–1), F(1), and F(2) give the correct answer, while F(–3), F(0), and F(3) don't.

Remember, our objective is to skip dilemmas like division by zero, so we must first find out if that is what the computer is trying to do. The way to do this is to look at line 10, which POKEs addresses 768 and 769 ($300 and $301) with 52 and 3 (or $34 and $03). These are the bytes indirectly used to tell the error routine where to go after it finds an error; normally these bytes contain the address $E38B, but line 10 changes this to address 820 decimal ($334). This is where our machine language routine is POKEd by line 20. Line 230 changes things back to normal after the program is finished. For those of you who wish to see the disassembled machine language routine, here it is:

```
STA $FB
PHP
PLA
STA $FC
STX $FD
STY $FE
PHA
PLP
LDA $FB
JMP $A906
```

The above routine is only used when there is an error. Locations 251–254 ($FB–$FE) are loaded with eights at the start (line 10) and loaded again each time through the loop that calculates F(C). However, if an error occurs in line 150, the error routine will load locations 251–254 with the contents of the registers at the time of the error and then continue with the next BASIC line. Line 160 prints out the contents of the registers and F(C). Thus the contents of 251–254 change only when an error occurs.

So, now the program doesn't crash; it just gives erroneous results, and that should also be avoided. Type in:

```
185 IF PEEK(252)>=48 THEN PRINT:GOTO 200
```

RUN the program again and there should be blanks where F(–3), F(0), and F(3) are involved. In other words, by PEEKing 252 and by comparing it to 48 we have skipped the errors; nothing has been printed, saved, recorded, or crashed. Only the proper numbers are still able to be used.

So much for mathematics. What if we define the function wrongly? LIST the program and change line 70 to: DEFFNF(C)=SQR(4 – C↑2/C and RUN. If all is not well you should see a line of four numbers, not eights, a SYNTAX ERROR (70) and line 70. Our error routine kicked in and in line 100 checked location 252 to see if it was less than 112 and told you about the error in syntax. This is really no advantage over the regular system, but if you are using the dynamic keyboard method to enter your DEFFNF(C) (see "Bootmaker for VIC, PET, and 64," COMPUTE!, May 1983), this routine would come in mighty handy.

Errors That Get Through

It should be pointed out that there is a potential problem with this routine. Change line 70 to DEFFNF(C)=SRR(4–C↑2)/C. Errors galore, but they weren't caught. Why not? I wish I knew. Please, not the BASIC interpreter again. All I can say is that in an instance like this you will, on most occasions, be able to tell there is an error and that the error is being caused by the DEFFN statement. Also, before including this specific Syntax Error routine in a program of your own, you should try putting a multiply sign (*) before the SQR in line 70 and then RUN. As you can see, the computer locks up. The only way to correct this situation is to turn the power off and reload the program. Weigh the advantages of including the Syntax Error routine described here against the very obvious disadvantage of system lockup.

To sum up:

1. POKE 768 and 769 ($300, $301) with the address of your machine language routine that will handle the BASIC errors. In the example presented here, 52 and 3 are POKEd, for location 820 ($334).

2. The error handling routine loads byte 252 ($FC) and provides the jump address to "Scan for next statement" at $A906 so you can reenter your program.

3. Check byte 252 (Status Register during an error) to see if it is greater than or equal to 48 for a mathematical error or 112 for a syntax error.

4. Take the appropriate action either to save an answer or to skip it.

5. POKE 768 and 769 with 139 and 227 respectively to restore the normal error vector address ($E38B). This is important since the computer won't be able to function in the immediate mode.

Error Suppression

```
10 POKE768,52:POKE769,3:FORC=0TO3:POKE251
   +C,8:NEXTC                          :rem 108
20 FORC=0TO16:READD:POKE820+C,D:NEXTC
                                       :rem 58
30 FORC=0TO17:PRINTCHR$(96);:NEXTC:PRINTC
   HR$(105)                             rem 51
40 PRINT" AC";TAB(5);"SR";TAB(10);"XR";TA
   B(15);"YR ";CHR$(125)               :rem 122
50 FORC1=0TO38:PRINTCHR$(96);:NEXTC1:PRIN
   T                                   :rem 178
60 PRINTCHR$(145);TAB(18);CHR$(177)
                                       :rem 215
70 DEFFNF(C)=SQR(4-C↑2)/C              :rem 206
80 SX=FNF(1)                           :rem 172
90 PRINTPEEK(251);TAB(4);PEEK(252);TAB(9)
   ;PEEK(253);TAB(14);PEEK(254);       :rem 27
100 IFPEEK(252)<112THENPRINT"{3 SPACES}SY
    NTAX OK":GOTO120                   :rem 134
110 PRINT"{3 SPACES}SYNTAX ERROR (70)":GO
    TO230                              :rem 148
120 FORC1=0TO38:PRINTCHR$(96);:NEXTC1:PRI
    NT                                 :rem 224
130 FORC=-3TO3                          :rem 49
140 C$=STR$(C)                         :rem 234
150 X=FNF(C)                           :rem 153
160 PRINTSTR$(PEEK(251));TAB(4);STR$(PEEK
    (252));TAB(9);                     :rem 198
170 PRINTSTR$(PEEK(253));TAB(14);STR$(PEE
    K(254))                             :rem 37
180 PRINTCHR$(145);TAB(20);"F("C$")=";
                                       :rem 16
190 PRINTX                             :rem 127
200 FORC1=0TO38:PRINTCHR$(96);:NEXTC1:PRI
    NT                                 :rem 223
210 FORC1=0TO3:POKE251+C1,8:NEXTC1:rem 34
220 NEXTC                              :rem 22
230 POKE768,139:POKE769,227             :rem 8
240 IFPEEK(252)>=112THENLIST70         :rem 21
300 DATA 133,251,8,104,133,252,134,253,13
    2,254,72,40,165,252,76,6,169:rem 65 ©
```

Hi-Res VIC Drawing

Jeff Wise

There comes a time when programmers want more subtle graphics than can be achieved with characters and low resolution. Do you ever feel like creating swirling, intricate webs of delicate, slender lines? Here's how to achieve high resolution on the VIC.

The designers of the VIC-20 thoughtfully included in the VIC chip a special programmable character generator. Though mainly intended for creating custom alphabets and symbols, it can also be used to generate an entire high-resolution screen.

Each character that the VIC puts on the screen, whether user-defined or standard, is stored in eight bytes of memory. Each byte defines one of the eight rows that comprises a VIC character. Furthermore, each of the rows is split up into eight sections corresponding to the eight bits in that row's byte. If a bit is *on* (there is a 1 in its location), then its chunk of the row is lit up. If a bit is *off* (it contains a 0), then its chunk of the row is blank.

Character Matrices

"Microdraw" sets up a matrix of 12 × 15 custom characters, all of which are initially made blank (by POKEing 0 into the defining bytes). Since each character is defined by eight bytes of eight bits each, we have a total of 11,520 bits, or dots, on the screen which we can turn on or off at will. To light up a dot, simply POKE a 1 into its corresponding bit.

Such high resolution comes at a price. In order to use custom characters, we first must set the character memory apart from the BASIC program area. Since we are using so many characters, a lot of memory is consumed—1.5K, nearly half the memory available in an unexpanded VIC.

Now that we've covered the theory, it's time to enjoy your VIC's hi-res capability. Type in Microdraw, save it, then run it. Plug in your joystick, if you have one. Select the foreground and background colors for the drawing area by pressing the number key with the appropriate color on it. The program will then set up the drawing area and display the cursor. You control the cursor by moving the joystick in the direction you want the cursor to go.

Initially, the cursor is in the erase mode, which means that the cursor does not create a line as it travels and will erase any line it comes in contact with. In this mode the TV speaker emits a low beeping tone.

To change to the drawing mode, press the fire button on the joystick. The TV speaker will then beep in a higher-pitched tone, and the cursor will leave a line as it travels. To change back to the erase mode, simply press the fire button once again.

The SAVE Function

The function keys offer three additional options: The f1 key erases the drawing screen and leaves the cursor in position; f3 starts the program from the beginning and resets all variables; and f5 causes the program to jump to a screen-saving routine. The saving routine is self-explanatory, as is the retrieval routine. To replay the data you have stored, choose selection 2 ("load an old one").

If you do not have a joystick, a simple modification will allow you to use the keyboard instead. Delete lines 330 and 340, and change line 320 to read:

```
320 A=PEEK(197):J0=-(A=44):J1=-(A=36):J2=
    -(A=20):J3=-(A=12):IF A=32THENB=ABS(B
    -1)                                rem 155
```

Now the cursor's up, down, left, and right motions are controlled by the I, M, J, and K keys, respectively. To change modes, press the space bar instead of the fire button. It is not necessary to hit the control keys repeatedly; the cursor will move as long as the key is held down. If no key is pressed, the cursor will stop. In all other respects, the program works as before.

Microdraw

```
10 POKE36869,240:POKE52,24:POKE56,24:POKE
   36879,27:CLR                          :rem 84
40 PRINT"{CLR}{3 DOWN}1) DRAW A NEW PICTU
   RE.{DOWN}2) LOAD AN OLD ONE.":POKE198,
   0                                     :rem 105
60 GETA$:IFA$<>"1"ANDA$<>"2"THEN60
                                         :rem 133
70 IFA$="2"THEN450                       :rem 214
80 PRINT"{CLR}{3 DOWN}BORDER COLOR?":GOSU
   B100:G=VAL(A$)-1                      :rem 123
90 PRINT"{3 DOWN}BACKGROUND COLOR?":GOSUB
   100:H=VAL(A$):GOTO120                 :rem 180
100 GETA$:IFA$<"1"ORA$>"8"THEN100 :rem 53
110 RETURN                               :rem 114
120 POKE36879,G+16*H-8:PRINT"{CLR}"
                                         :rem 77
130 FORX=0TO21:FORY=0TO22:POKE7680+X+22*Y
    ,160:POKE38400+X+22*Y,G:NEXT:NEXT
                                         :rem 44
140 IFW=1THEN160                         :rem 175
150 FORI=6144TO7679:POKEI,0:NEXT:FORI=742
    4TO7431:POKEI,255:NEXT               :rem 182
160 C=0:POKE36869,254:FORX=5TO16:FORY=3TO
    18:IFX+Y=34THENNEXT:GOTO180          :rem 53
170 C=C+1:POKE7680+X+22*Y,C:NEXT:NEXT
                                         :rem 235
180 Y=18:FORX=3TO18:POKE7680+X+22*Y,160:N
    EXT:C=90                             :rem 191
190 IFJ1THENF=F+1:IFF>7AND(C+1)/16<>INT((
    C+1)/16)THENF=0:C=C+1:GOTO210        :rem 82
200 IFF>7THENF=7                         :rem 197
210 IFJ3THENF=F-1:IFF<0AND(C-1)/16<>INT((
    C-1)/16)THENF=7:C=C-1:GOTO230        :rem 85
220 IFF<0THENF=0                         :rem 183
230 IFJ0THENE=E+1:IFE>7ANDC<177THENE=0:C=
    C+16:GOTO250                         :rem 222
240 IFE>7THENE=7                         :rem 199
250 IFJ2THENE=E-1:IFE<0ANDC>16THENE=7:C=C
    -16:GOTO270                          :rem 176
260 IFE<0THENE=0                         :rem 185
270 POKE6144+(8*C)+F,PEEK(6144+(8*C)+F)AN
    DNOT(2↑(7-E))                        :rem 203
280 POKE6144+(8*C)+F,PEEK(6144+(8*C)+F)OR
    2↑(7-E)                              :rem 88
290 POKE36878,15:POKE36874+2*B,130+INT(C/
    2.14):POKE36878,0                    :rem 221
300 POKE36874+2*ABS(B-1),0               :rem 120
310 IFB=0THENPOKE6144+(8*C)+F,PEEK(6144+(
    8*C)+F)-2↑(7-E)                      :rem 75
320 POKE37154,127:Z=128ANDPEEK(37152):J0=
    -(Z=0)                               :rem 123
330 POKE37154,255:Z=PEEK(37151)  :rem 217
340 J1=-((ZAND8)=0):J2=-((ZAND16)=0):J3=-
    ((ZAND4)=0):J=-((ZAND32)=0):IFJTHENB=
    ABS(B-1)                             :rem 91
350 GETA$:IFA$<>CHR$(133)ANDA$<>CHR$(134)
    ANDA$<>CHR$(135)THEN190              :rem 112
360 IFA$=CHR$(134)THEN10                 :rem 68
370 IFA$=CHR$(135)THEN390                :rem 129
380 FORI=6144TO7423:POKEI,0:NEXT:FORI=743
    2TO7679:POKEI,0:NEXT:GOTO190  :rem 92
390 POKE36869,240:POKE36879,27    :rem 172
400 PRINT"{CLR}{DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK?":POKE198,0    :rem 237
402 GETA$:IFA$<>"T"ANDA$<>"D"THEN402
                                         :rem 26
405 IFA$="T"THENPRINT"REWIND TAPE":rem 85
415 PRINT"HIT A KEY WHEN READY"   :rem 33
420 B$="":GETB$:IFB$<>" "THEN420 :rem 175
425 IFA$="T"THENOPEN1,1,1         :rem 176
426 IFA$="D"THENINPUT"FILENAME";N$:N$=N$+
    ",S,W":D=8:OPEN1,8,5,N$          :rem 5
430 PRINT#1,G:PRINT#1,H:FORA=6144TO7679:P
    RINT#1,PEEK(A):NEXT:CLOSE1   :rem 192
440 PRINT"{CLR}YOUR PICTURE IS SAVED.":GO
    TO10                             :rem 120
450 PRINT"{DOWN}{RVS}T{OFF}APE OR {RVS}D
    {OFF}ISK?":D=1:N$=""             :rem 17
455 GETA$:IFA$<>"T"ANDA$<>"D"THEN455
                                         :rem 42
460 IFA$="T"THENPRINT"{3 DOWN}INSERT CASS
    ETTE AND{3 SPACES}REWIND IT"   :rem 0
470 PRINT"{DOWN}WHEN YOU ARE READY HIT(SP
    ACE).{DOWN}":W=1:WAIT198,1    :rem 71
475 IFA$="T"THENOPEN1,1,0         :rem 180
480 IFA$="D"THENINPUT"FILENAME";N$:D=8:N$
    =N$+",S,R":OPEN1,8,5,N$          :rem 0
490 INPUT#1,G:INPUT#1,H:FORA=6144TO7679:I
    NPUT#1,C:POKEA,C:NEXT:CLOSE1 :rem 116
500 GOTO120                          :rem 97
10000 OPEN15,8,15:INPUT#15,A,B$,C,D:CLOSE
      15                             :rem 198
```

©

ML Applesort

Richard Salley

A machine language program is necessary when sorting large amounts of data. "ML Applesort" is a machine language utility that will quickly sort an array of any length.

In COMPUTE! (September 1982), David Lummis presented an excellent machine language sort routine for PET/CBM computers. "ML Applesort" is a modification of Lummis's program for the Apple.

The original program has been compressed to fit it into page 3 of memory. This is a safe area for ML programs so you do not have to worry about overwriting the routine with a long program or numerous and lengthy string arrays. The zero page locations used for temporary storage, counting, and address indexing were chosen because most of these locations are used primarily in connection with hi-res graphics, and it is unlikely such programs will be used concurrently with a sort utility.

ML Applesort makes use of the Apple's special & command. When the BASIC interpreter encounters the & in a program, it goes to location $3F5 (1013 decimal) and then performs an unconditional jump to the address contained in $3F6 and $3F7. In this program, 0 is placed in $3F6 (1014 decimal) and 3 is placed in $3F7 (1015 decimal). This will cause a jump to location $300 (768 decimal), which is where the machine language routine begins.

The first instruction at $300 is a JSR (Jump to SubRoutine) $F7D9. This is a monitor subroutine that fetches the address of a string and stores it in locations $9B and $9C. By placing the name of the string array we want sorted immediately after the &, this routine will tell us where in memory that array is stored. The correct format for calling the ML sort routine from BASIC is as follows:

```
100 &X$
```

where X$ is the name of the array to be sorted. When the routine returns to BASIC, the named array will be sorted alphabetically in ascending order. How the program does the sorting can be understood by studying a disassembly. To enter the program, use the BASIC Loader (Program 1).

After placing the program into memory by running Program 1, save it to disk by typing:

```
BSAVE APPLESORT, A$300, L$FF
```

You can then BLOAD the sort routine and use it with any of your own BASIC programs.

Program 2 shows how easily the program can be used and how quickly it can sort an array with strings of varying lengths. I'm sure COMPUTE! readers with Apple machines will find numerous applications for this useful utility.

Program 1: ML Applesort

```
100  REM  ...ML APPLESORT
110  REM  ...POKE & JUMP ADDRESS
```

```
120  POKE 1013,76: POKE 1014,0: POKE 10
     15,3
130  REM  ...POKE ML
140  FOR ADDR = 768 TO 941: READ CODE:C
     KSUM = CKSUM + CODE: POKE ADDR,COD
     E: NEXT
150  IF CKSUM <  > 26104 THEN  PRINT "E
     RROR IN DATA STATEMENTS": STOP
768  DATA  32, 217, 247, 165, 155, 133,
      1, 165
776  DATA  156, 133, 2, 160, 5, 177, 1,
      133
784  DATA  208, 200, 177, 1, 133, 209,
     169, 1
792  DATA  133, 210, 169, 0, 133, 211,
     24, 165
800  DATA  1, 105, 7, 133, 235, 165, 2,
      105
808  DATA  0, 133, 236, 165, 235, 133,
     225, 165
816  DATA  236, 133, 226, 24, 165, 225,
      105, 3
824  DATA  133, 235, 165, 226, 105, 0,
     133, 236
832  DATA  160, 0, 177, 225, 208, 34, 2
     4, 165
840  DATA  210, 105, 1, 133, 210, 165,
     211, 105
848  DATA  0, 133, 211, 197, 208, 144,
     212, 165
856  DATA  210, 197, 209, 144, 206, 165
     , 212, 208
864  DATA  1, 96, 169, 0, 133, 212, 240
     , 174
872  DATA  133, 213, 177, 235, 240, 239
     , 133, 214
880  DATA  200, 177, 225, 133, 233, 177
     , 235,133
888  DATA  237, 200, 177, 225, 133, 234
     , 177, 235
896  DATA  133, 238, 160, 0, 177, 233,
     209, 237
904  DATA  144, 188, 240, 2, 176, 9, 2
     00, 196
912  DATA  213, 240, 179, 196, 214, 208
     , 237, 160
920  DATA  2, 177, 225, 72, 177, 235,
     145, 225
928  DATA  104, 145, 235, 136, 16, 243,
      169, 1
936  DATA  133, 212, 76, 70, 3, 0
```

Program 2: ML Applesort Demo

```
10  REM ...SORT DEMO
20  IF  PEEK (768) = 32 THEN 40
30  PRINT  CHR$ (4);"BLOAD APPLESORT"
40  HOME : VTAB 5: PRINT "POINTER SORT
     DEMO"
50  VTAB 7: INPUT "ENTER # OF STRINGS T
     O SORT ";N
60  DIM R$(N)
70 B$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
80  FLASH : PRINT : PRINT "BUILDING STR
     INGS"
90  NORMAL
100  FOR F = 1 TO N:X =  INT ( RND (1) *
     7) + 2: FOR G = 1 TO X
110 R$(F) = R$(F) +  MID$ (B$, INT ( RND
     (1) * 26) + 1,1)
120  NEXT : NEXT
130  FOR F = 1 TO N: PRINT R$(F),: NEXT

140  PRINT : INPUT "PRESS <RETURN> TO S
     ORT ";XX$
150  PRINT : PRINT "SORT BEGUN"
160  & R$
170  PRINT : PRINT "SORT FINISHED!!"
180  PRINT : INPUT "PRESS <RETURN> TO P
     RINT SORTED LIST ";XX$
190  FOR F = 1 TO N: PRINT R$(F),: NEXT

200  PRINT : PRINT "END OF DEMO": END  ©
```

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 questions frequently asked by beginners.*

Q Is it safe to plug a whole home computer system into a single wall socket? I'm talking about a computer, TV, cassette recorder, disk drive, printer, and a modem. Will it blow a fuse? Or will I have to run extension cords from nearby wall sockets for some of the peripherals?

A As long as no other power-hungry appliance is on the same socket, plugging a whole home computer system into one outlet is perfectly safe. Home computers and their peripherals actually don't use very much electricity at all. In fact, the typical home computer consumes less power than the light bulb you'll burn to see it by.

For instance, one of our editors has a computer system at home which consists of an Atari 800 with 48K of memory, a disk drive, a cassette recorder, an Atari 850 Interface Module, a color monitor, an 80-column dot-matrix printer, and a modem. Everything but the monitor and the modem is plugged into a six-socket power strip, which in turn is plugged into a single wall outlet. The power strip has a 15-amp circuit breaker which has never popped. That means the system uses less than 1650–1800 watts, or a little more electricity than a blow dryer.

By far the most power-hungry component of a home computer system is the TV set or monitor. A small black-and-white TV or monochrome monitor won't use much electricity, but a large color TV can use more power than the rest of the system put together. If you're worried about overloads, plug the TV into a different outlet.

One thing you should avoid is hooking up the computer system to a circuit shared by heavy-duty appliances like air conditioners, dishwashers, clothes washers, dryers, refrigerators, and water heaters. Have you ever noticed your room lights dim for a second when a heavy appliance kicks on? The sudden demand for power momentarily drains the circuit and lowers the voltage. Those kinds of fluctuations aren't healthy for computers, whose chips are very sensitive to power sags and surges. (That's why some people invest in surge protectors or voltage stabilizers.)

If you aren't sure whether a certain wall outlet is wired to the same circuit as another outlet serving a heavy appliance, test it by plugging in a lamp. Then switch on some of the major appliances in your home while someone watches the lamp for any telltale dimming. If an outlet is affected, you may have to run an extension cord from a more distant socket to reach your computer system. This is particularly true in houses and apartment buildings with older wiring.

Q I'm moving to another state and I'd like to transport my computer by plane. Do you think it would be safe in the baggage compartment?

A Recently some of COMPUTE!'s editors went on a trip to the Comdex trade show in Las Vegas and witnessed some unpleasant violence to a Compaq transportable computer. Although the Compaq is one of the better transportables, by the time the poor computer tumbled off the airport conveyor belt onto the revolving baggage-claim carousel, it looked almost destroyed. The top of the case was torn off, exposing the built-in monitor screen and delicate disk drives. Heavy hard-shell suitcases kept sliding off the conveyor belt and bashing into the computer, knocking more parts loose. Wires and cables were hanging out. It wasn't pretty.

Based on what we saw that day, and on other airline experiences, our advice is not to ship a computer as baggage unless it's *very* well packed and padded, preferably in its original box with the form-fitting Styrofoam inserts. Have you ever seen the TV commercial in which a suitcase is batted around by an ape? If your computer is packed well enough to withstand that kind of battering, you're probably safe. Otherwise, you might consider another method of shipping.

Incidentally, if you're traveling by air with a computer as carry-on baggage, insist on having it hand-checked when passing through security checkpoints. We know of a newspaper reporter who unknowingly allowed his TRS-80 Model 100 lap computer to suffer exposure from an airport x-ray machine. "It just went crazy," he said.©

NEWS&PRODUCTS

Percussion Emulator For Apple

Drum-Key, recently introduced by Peripheral Visions, Inc., is an electronic music interface board for use with the Apple II series of computers. It will allow you to interface stereos and electric instrument amplifiers to your computer.

Drum-Key lets you compose, play, and record percussion sounds and riffs, as well as play along with the 100 included rhythm patterns and 26 songs.

A complement of 28 sounds is included. Among these are snare, tom-toms, cymbals, cowbell, tambourine, and six sounds made by conventional drum synthesizers.

Suggested retail price is $139.

Peripheral Visions, Inc.
Great Valley Parkway
Malvern, PA 19355
(215) 627-3535

RS-232 Modem Adapter For Atari

Advanced Interface Devices, Inc., has announced the R-Verter, a serial bus modem adapter for Atari 400, 600XL, 800, and 800XL home computers.

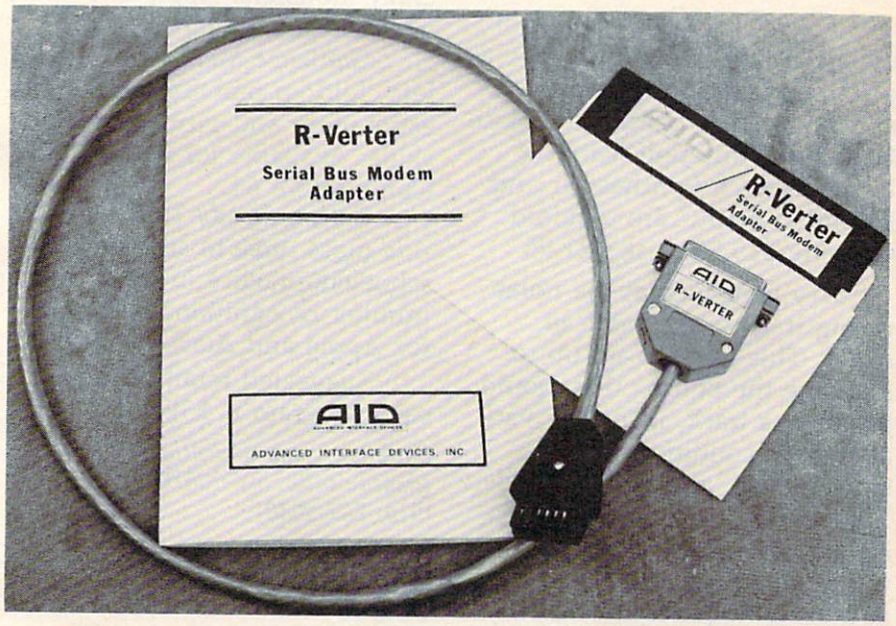
The R-Verter allows most modems and other RS-232C devices to be used directly with Atari computers without using the Atari 850 Interface Module or other interface. It requires no modifications of the computer or other peripherals. All circuitry is contained in an RS-232C type connector to minimize size.

The R-Verter comes with a software package which includes a smart terminal emulator and an RS-232C device handler.

Most common RS-232C handshaking configurations are available using internal jumpers.

Price for the R-Verter and print echo software is $49.95.

Advanced Interface Devices, Inc.
P.O. Box 2188
Melbourne, FL 32902
(305) 676-1275



The R-Verter allows most modems and other RS-232C devices to be used directly with Atari computers without using the Atari 850 Interface Module.

Game Development Program For The Commodore 64

Aspiring arcade-game designers can develop graphics for their games more quickly and easily by using the *Graphics Master*.

Written for the Commodore 64, this programming aid adds 52 new commands to BASIC and has numerous features that support game development.

Software Unlimited will soon release a compiler to make the completed game run faster.

Disk only; $29.95. (Please include $3.00 for postage and handling.)

Software Unlimited
P.O. Box 429
Klamath Falls, OR 97601

Educational And Entertainment Software For The TI-99/4A

American Software has announced four new software packages for the Texas Instruments 99/4A.

In *Fireball*, an arcade game for ages ten and older, you must climb a volcano without being hit by fireballs or falling into holes. The game requires either the Editor/Assembler cartridge or the Mini-Memory cartridge. Disk only; $16.95.

Letter Fun helps preschoolers learn the letters of the alphabet using colorful graphics and music. The child can choose from three different learning levels. Speech Synthesizer and Extended BASIC are required. Cassette $19.95; disk $21.95.

Try your luck at the horse racing track with *American Derby*. This game is set up to simulate the betting that would go on at a track, including variable track conditions, an insider's sheet, and realistic odds. You can bet on up to 36 different horses. Designed for ages ten to adult; up to six may play at a time. Requires Extended BASIC. Cassette $14.95; disk $16.95.

Speed Read was written for adults who want to improve their reading speed. This package of programs includes information on the reading process as well as pacing aids and reading passages to test your speed. It requires Extended BASIC. Cassette $29.95; disk $31.95 (disk version requires memory expansion).

American Software Design & Distribution Co.
P.O. Box 46
Cottage Grove, MN 55016
(612) 459-0557

Home Educational Software For Apple, Atari, And Commodore

Sunburst Communications, which has supplied educational materials to schools for 12 years, has released three new products from their microcomputer division.

The Incredible Laboratory (ages seven to adult) uses the problem-solving strategy of trial and error and note-taking to discover what combinations of mysterious chemicals make up crazy monsters. Apple and Atari versions are available.

Challenge Math (ages 6–11) lets children practice basic math, estimation, and problem-solving skills. Available for Apple and Commodore 64.

Getting Ready To Read And Add (ages three to six) gives preschoolers practice in letter and number recognition. Available for Apple and Atari.

Suggested retail price for each program is $39.95.

Sunburst Communications, Inc.
Pleasantville, NY 10570
(914) 769-5030

New Telecommunications Package For Apple

The Networker modem, recently introduced by ZOOM Telephonics, is a complete telecommunications package for the Apple II, II+, and IIe computers.

For $129, you get a single-slot, direct-connect, 300-baud modem, terminal software, and a free subscription to The Source.

An enhanced version of the terminal software, Netmaster, can be purchased separately for

Apple owners can get a complete telecommunications package, including modem and terminal software, by purchasing the Netmaster system.

$79. If purchased with the Networker, the price of the entire package is $179.

ZOOM Telephonics plans to offer a complete line of modems, including modems for the IBM-PC.

ZOOM Telephonics
207 South St.
Boston, MA 02111
(617) 423-1072

Telecommunications Aid

Source Telecomputing Corporation (STC) has announced *Apple Sourcelink*, the second in its series of communications software designed to supplement use of The Source by personal computer owners.

The software is compatible with the new Apple modem, as well as with the Hayes and Transend modem products, and is designed for the Apple II, IIe, and II+ with a minimum 48K of memory.

It combines features such as automatic dial-up and sign-on procedure for Telenet, Uninet, and Sourcenet data communications networks; "one-button" access to major services on The Source; simultaneous capture of data from The Source in the Apple memory or disks, including

a capture editor; and data transfer from Apple disks to The Source, or vice versa, while on-line.

An additional feature allows Apple and IBM users to access automatically any number of predetermined services and data bases, once on-line.

The Source
1616 Anderson Road
McLean, VA 22102
(703) 734-7500

Inexpensive Light Pen For Commodore Computers

Creative Electronics has announced the introduction of a new light pen for the Commodore 64 and VIC-20.

The light pen, which offers close to one-pixel accuracy for high-resolution graphics, comes with two sample programs.

Both versions retail for $14.95.

Creative Electronics
P.O. Box 4253
1714 Sandalwood
Thousand Oaks, CA 91360
(805) 492-1506

Alphabet, Math Games For Children

Two educational software games designed to help children understand the alphabet, multiplication, and division have been introduced by Avalon Hill Game Company's Intelligence Quest Software division.

DIVEX, appropriate for ages 8–12, has three levels of multiplication and division to master, and requires a child to use mathematical skills to protect his or her "land" from incorrect answers.

It is available on diskette ($21) for Ataris with at least 32K memory. Cassette editions ($16) for the Commodore 64 and Atari will be available later.

In *ABC Caterpillar*, the player, controlling a bright green caterpillar, searches for letters of the alphabet as they pass by on the screen. The goal is to find and gobble up the letters in alphabetical order.

For children 3–8 years old, *ABC Caterpillar* is available for the VIC-20 at a cost of $16. A Commodore 64 edition is planned also.

Intelligence Quest Software
4517 Harford Road
Baltimore, MD 21214
(301) 254-9200

New Data Base Management Software For IBM Home Computers

Condor Jr. is a data base management system specially customized for beginning microcomputer users.

The program is available for the IBM-PC and PCjr, and retails for $195.

Beyond its extensive math and printing capabilities, *Condor Jr.* can be upgraded to *Condor 3* (a more sophisticated data base manager). Other features include multilevel sorts and a variety of report generators.

Condor
2051 S. State St.
Ann Arbor, MI 48104
(313) 769-2418

New Speech Synthesizers

Three new Voice Box speech synthesizers have been introduced by The Alien Group, two of which are designed for Apple II and Apple-compatible computers. The third is for any machine which has a standard (RS-232C) serial port.

Using a new speech chip, the programs produce speech directly from English text, adding inflection either automatically or according to numbers inserted by the user. All units have an unlimited vocabulary, and can speak with a male or female voice, fast or slow, or loud or soft, depending on what commands are added to the text. It is not necessary to mark syllable boundaries or to use phoneme spelling when adding intonation.

The Voice Box 3m model, designed for the Apple, retails for $129. Voice Box 3i, also for the Apple, costs $219. Prices include a Voice Box board, disk software, and external speaker.

The Voice Box 3s, which can connect to any computer via the standard RS-232C serial interface, includes an integral speaker and retails for $269.

The Alien Group
27 West 23rd St.
New York, NY 10010
(212) 741-1770

New Product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication.

COMPUTE! *welcomes notices of upcoming events and requests that the sponsors send a short description, their name and phone number, and an address to which interested readers may write for further information. Please send notices at least three months before the date of the event, to: Calendar, P.O. Box 5406, Greensboro, NC 27403.* ©

MLX Machine Language Entry Program For Commodore 64

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. *You need to know nothing about machine language to use MLX—it was designed for everyone.*

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file. You can then use the LOAD command to read the program into the computer:

LOAD "filename",1,1 (for tape)
LOAD "filename",8,1 (for disk)

To start the program, you enter a SYS command that transfers control from BASIC to machine language. The starting SYS number appears in the article.

Using MLX

Type in and save MLX for your 64 (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

You'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the SPACE bar, or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

```
    U  I  O              7  8  9
H   J  K  L   become  0   4  5  6
    M  ,  .              1  2  3
```

MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

SHIFT-S: Save
SHIFT-L: Load
SHIFT-N: New Address
SHIFT-D: Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

What if you forgot where you stopped typing? Use the Display command to scan memory from the beginning to the end of the program. When you reach the end of your typing, the lines will contain a random pattern of numbers. When you see the end of your typing, press any key to stop the listing. Use the New Address command to continue typing from the proper location.

MLX: Machine Language Entry

```
10 REM LINES CHANGED FROM MLX VERSION 2.0
   0 ARE 750,765,770 AND 860       :rem 50
100 PRINT"{CLR}{6}";CHR$(142);CHR$(8);:
    POKE53281,1:POKE53280,1        :rem 67
101 POKE 788,52:REM DISABLE RUN/STOP
                                   :rem 119
110 PRINT"{RVS}{39 SPACES}";       :rem 176
120 PRINT"{RVS}{14 SPACES}{RIGHT}{OFF}
    {*}£{RVS}{RIGHT} {RIGHT}{2 SPACES}
```

```
    [*][OFF][*]£[RVS]£[RVS]
    [14 SPACES]";                          :rem 250
130 PRINT"[RVS][14 SPACES][RIGHT] [G]
    [RIGHT] [2 RIGHT] [OFF]£[RVS]£[*]
    [OFF][*][RVS][14 SPACES]";             :rem 35
140 PRINT"[RVS][41 SPACES]"                :rem 120
200 PRINT"[2 DOWN][PUR][BLK] MACHINE LANG
    UAGE EDITOR VERSION 2.01[5 DOWN]"
                                           :rem 237
210 PRINT"[5][2 UP]STARTING ADDRESS?
    [8 SPACES][9 LEFT]";                   :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+119*F)
                                           :rem 166
220 IFS<256ORS>40960ANDS<49152ORS>53247
    THENGOSUB3000:GOTO210                  :rem 235
225 PRINT:PRINT:PRINT                      :rem 180
230 PRINT"[5][2 UP]ENDING ADDRESS?
    [8 SPACES][9 LEFT]";:INPUTE:F=1-F:C$=
    CHR$(31+119*F)                         :rem 20
240 IFE<256ORE>40960ANDE<49152ORE>53247
    THENGOSUB3000:GOTO230                  :rem 183
250 IFE<STHENPRINTC$;"[RVS]ENDING < START
    [2 SPACES]":GOSUB1000:GOTO 230
                                           :rem 176
260 PRINT:PRINT:PRINT                      :rem 179
300 PRINT"[CLR]";CHR$(14):AD=S:POKEV+21,0
                                           :rem 225
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),
    2),5);":";                             :rem 33
315 FORJ=ATO6                              :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                           :rem 228
390 IFN=-211THEN 710                       :rem 62
400 IFN=-204THEN 790                       :rem 64
410 IFN=-206THENPRINT:INPUT"[DOWN]ENTER N
    EW ADDRESS";ZZ                         :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    [RVS]OUT OF RANGE":GOSUB1000:GOTO410
                                           :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                           :rem 238
420 IF N<>-196 THEN 480                    :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                              :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"[LEFT], NOT MORE THAN";E:GOTO43
    0                                      :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";             :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
    "+MID$(STR$(N),2),3);",";              :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                           :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                              :rem 50
480 IFN<0 THEN PRINT:GOTO310               :rem 168
490 A(J)=N:NEXTJ                           :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT            :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
    );                                     :rem 94
511 IFN=-1THENA=6:GOTO315                  :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
                                           :rem 122
520 PRINT:PRINT"LINE ENTERED WRONG : RE-E
    NTER":PRINT:GOSUB1000:GOTO310:rem 176
530 GOSUB2000                              :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,0:POKE54273,0                      :rem 227
550 AD=AD+6:IF AD<E THEN 310               :rem 212
560 GOTO 710                               :rem 108
570 N=0:Z=0                                :rem 88
580 PRINT"[£]";                            :rem 81
581 GETA$:IFA$=""THEN581                   :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*(A$=".")-4*
    (A$="J")-5*(A$="K")-6*(A$="L"):rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
    ):IFA$="H"THENA$="0"                   :rem 134
584 IFAV>0THENA$=CHR$(48+AV)               :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                          :rem 229
590 IFA>128THENN=-A:RETURN                 :rem 137
600 IFA<>20 THEN 630                       :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    [OFF][LEFT] [LEFT]";:GOTO690           :rem 62
620 GOTO570                                :rem 109
630 IFA<48ORA>57THEN580                    :rem 105
640 PRINTA$;:N=N*10+A-48                   :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                           :rem 229
660 Z=Z+1:IFZ<3THEN580                     :rem 71
670 IFZ=0THENGOSUB1000:GOTO570             :rem 114
680 PRINT",";:RETURN                       :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                           :rem 149
691 FORI=1TO3:T=PEEK(S%-I)                 :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                           :rem 205
700 PRINTLEFT$("[3 LEFT]",I-1);:RETURN
                                           :rem 7
710 PRINT"[CLR][RVS]*** SAVE ***[3 DOWN]"
                                           :rem 236
715 PRINT"[2 DOWN](PRESS [RVS]RETURN[OFF]
    ALONE TO CANCEL SAVE)[DOWN]":rem 106
720 F$="":INPUT"[DOWN] FILENAME";F$:IFF$=
    ""THENPRINT:PRINT:GOTO310              :rem 71
730 PRINT:PRINT"[2 DOWN][RVS]T[OFF]APE OR
    [RVS]D[OFF]ISK: (T/D)"                 :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
                                           :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$:
    OPEN15,8,15,"S"+F$:CLOSE15             :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                       :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                           :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                                     :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
    *256:POKE780,253                       :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
    2)*256:SYS65496                        :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
                                           :rem 111
775 PRINT"[DOWN]DONE.[DOWN]":GOTO310
                                           :rem 113
780 PRINT"[DOWN]ERROR ON SAVE.[2 SPACES]T
    RY AGAIN.":IFDV=1THEN720               :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720                   :rem 103
790 PRINT"[CLR][RVS]*** LOAD ***[2 DOWN]"
                                           :rem 212
795 PRINT"[2 DOWN](PRESS [RVS]RETURN[OFF]
    ALONE TO CANCEL LOAD)"                 :rem 82
800 F$="":INPUT"[2 DOWN] FILENAME";F$:IFF
    $=""THENPRINT:GOTO310                  :rem 144
810 PRINT:PRINT"[2 DOWN][RVS]T[OFF]APE OR
    [RVS]D[OFF]ISK: (T/D)"                 :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
                                           :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                           :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                       :rem 2
```

```
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                              :rem 70
850 POKE780,0:SYS65493              :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
                                    :rem 111
865 PRINT"{DOWN}DONE.":GOTO310      :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
                                    :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800            :rem 102
1000 REM BUZZER                     :rem 135
1001 POKE54296,15:POKE54277,45:POKE54278,
     165                            :rem 207
1002 POKE54276,33:POKE 54273,6:POKE54272,
     5                              :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
     273,0:POKE54272,0:RETURN       :rem 202
2000 REM BELL SOUND                 :rem 78
2001 POKE54296,15:POKE54277,0:POKE54278,2
     47                             :rem 152
2002 POKE 54276,17:POKE54273,40:POKE54272
     ,0                             :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
                                    :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1000                       :rem 89
```
©

CAPUTE!

Modifications Or Corrections To Previous Articles

Atari XL Compatibility Update

Upon testing with our new 800XL, we have been pleased to discover that the vast majority of our previously published Atari programs will run without modification. Of the few programs that will not run as is, almost all, including the popular "Scriptor" word processor (April 1983), operate properly when used with the Atari *Translator.* This program, available on a disk (DX5063) from Atari, enables most programs written for the older Ataris to be run on the new XL models. So far we have discovered only two programs, "Demons Of Osiris" (January 1984) and "Ski" (February 1983), that the *Translator* cannot cure. These programs can be run on the new computers only if you have a BASIC cartridge from the older Atari series to plug in.

For information on obtaining the *Translator* disk, call Atari's Customer Relations Department at 800-538-8543 (inside California, 800-672-1404).

MLX For Commodore 64

There is an error in the article accompanying the "MLX" machine language editor program in the March and May issues. The article states, "If you enter less than three digits, you can press either the comma, SPACE bar, or RETURN key to advance to the next number." However, when the numeric keypad feature was added to MLX, the comma key was redefined as the numeral 2. As a result, the comma key can no longer be used to advance to the next number; however, the SPACE bar and RETURN key still work as stated.

A number of readers have expressed concern at the number of revisions to MLX since it was first published. These changes generally represent enhancements, *not* corrections. Any version of MLX may be used to type in any program for the 64 presented in MLX format, regardless of whether the program is from COMPUTE!, COMPUTE!'s GAZETTE, or a COMPUTE! book. The only version of 64 MLX known to contain a bug is the one from the March issue, and the correction was given in the May "CAPUTE!" column.

Automatic Proofreader For The 64

The final paragraph of the article which accompanies the "Automatic Proofreader" program each month indicates that on the 64 the Proofreader can be protected during tape LOADs and SAVEs by typing POKE 178,165. Richard Murphy points out that the proper value to protect the Proofreader is POKE 178,251. This POKE is not necessary for disk operations.

64 Hi-Res Screen Printing

Many readers have asked for a way to print a copy of the elaborate designs they create with the "3-D Plotting" program from the May issue (p. 58). Reader Henry Mervis observes that, for Commodore 64 owners, the solution is in the same issue, in the "Hi-Res Graphics Editor" program (p. 82). To create a hard copy of the results of the 3-D Plotting programs (or of almost any other hi-res screen display), load the machine language program you created for the Hi-Res Editor (Program 2, p. 80), using the LOAD command format described on page 82. Remember to enter the line to move the BASIC memory area:

```
POKE 642,128:POKE 44,128:POKE 32768,0:NEW
```

Next, load either "Rectan" (Program 1, p. 60) or "Spheri" (Program 2, p. 62). For Rectan, change line 600 to read 600 SYS 49152; for Spheri, change line 610 to 610 SYS 49152. Then RUN the program in the normal manner. When your design is complete, a rectangle will appear on the screen. Turn on your printer and press the P key and your design should begin to print.

The screen dump routine will work only on a Commodore 1525 printer or with an interface that emulates the 1525. The routine will not work with Commodore 1526 printers. ©

Lyco Computer Marketing & Consultants

TO ORDER CALL US

TOLL FREE 800-233-8760

In PA 1-717-327-1824

DISK DRIVES

RANA

1000 ... $ CALL

PERCOM

ATARI AT88-PD $329.00
AT-88 DOUBLER $129.00
ADD-ON DRIVES ..... $ CALL

HARD DISK DRIVES AVAILABLE FOR APPLE IBM-PC 5 MEG 10 MEG 15 MEG 20 MEG

TRAK DISK DRIVES

AT-D1 ..... $379.00
AT-D2 ..... $399.00
AT-D2 TURBO PAK $22.95
PRINTER CABLE $22.95

CONCORD DISK DRIVES

APPLE 163K DRIVE $199.00
APPLE 326K DRIVE $229.00
APPLE CONTROLLER CARD $69.00

COMMODORE VIC 174K ... $289.00
COMMODORE VIC 348K ... $359.00

ATARI 176K MASTER ... $289.00
ATARI 348K MASTER ... $369.00
ATARI ADD-ON DRIVES ... $CALL

DISKETTES

ELEPHANT

SINGLE SIDE SD (10) $17.75
SD (100) $16.75/10
SINGLE SIDE DD (10) $21.75
DD (100) $20.75/10
DOUBLE SIDE DD (10) ... $26.75
DD (100) $24.75/10

MAXELL

MD1 (10) $27.75
MD2 (10) $37.75

CERTRON CASSETTES

CC-10 (12) $15.99
CC-20 (12) $17.99

COMPUTER CARE

BIB

DISK DRIVE CLEANER ... $12.75
COMPUTER CARE KIT ... $19.75

NORTRONICS

DISK DRIVE CLEANER WITH SOFTWARE FOR IBM-PC, ATARI, VIC, APPLE, & TI $29.75
DISK CLEANER REFILLS ... $14.75
CASS DRIVE CLEANER $9.95
MEDIA BULK ERASER ... $46.75

commodore

VIC 64 ..... $ CALL
VIC 1541 ..... $239.00

SSI

KNIGHTS OF DESERT ... $26.75
EAGLES $26.75
TIGERS IN SNOW $26.75
COMBAT LEADER $26.75
BATTLE FOR N $26.75

EPYX 64

JUMPMAN C/D $27.75
JUMPMAN JR R $27.75
PITSTOP R $27.75
TEMPLE ASPHAI $27.75
GATE T ASPHAI $27.75
CRUSH, C&C C/D $21.75

INFOCOM 64

ENCHANTER $34.75
PLANETFALL $34.75

DATASOFT

POOYAN C/D $21.75
O'RILEYS MINE $21.75

PARKER 20

FROGGER $33.75

PROFESSIONAL SOFTWARE

WORD PRO 3 ..... $59.75
SPELL RIGHT $39.75
BOTH TOGETHER $79.75

EASTERN HOUSE

RABBIT 20 ROM $34.75
RABBIT 64 ROM $34.75

INNOVATIVE CONCEPTS

DISK STORAGE (10) $4.95
DISK STORAGE (15) $9.95
DISK STORAGE (25) $19.95
DISK STORAGE (50) $26.75
ROM STORAGE (10) $24.75

MODEMS

ANCHOR MARK I $45.75
ANCHOR MARK II $79.75
HAYES SMART $239.00
HAYES MICRO II $309.00
MICROBIT 1000C $129.00
CAT $144.00
D-CAT $155.00
J-CAT $115.00
APPLE CAT II $279.00
212 APPLE CAT $589.00

MONITORS

SAKATA COLOR $249.00
TAXAN GREEN $119.00
TAXAN AMBER $129.00
TAXAN RGB COMPOSITE $289.00
ZENITH AMBER $105.00
ZENITH GREEN $95.00
GORILLA GREEN $88.00
GORILLA AMBER $95.00
NEC JB1260 ..... $99.00
NEC JB1205 ..... $145.00
NEC JB1215 COLOR ... $269.00
AMDEK GREEN $145.00
AMDEK AMBER $149.00
AMDEK COLOR 1 ..... $289.00

apple®

BRODERBUND

BANK STREET D $44.75
CHOPLIFTER D $24.95
DAVID'D MAGIC $24.95
DROL D $24.95
LODE RUN D $24.95
SEAFOX D $22.95
SPARE CHANGE D $24.95
AE D $24.75

KOALA

KOALA TABLET $84.75
PROGRAMMERS GUIDE ... $12.75
GEOMETRIC DESIGNS ... $22.75
LOGO DESIGN $27.75
ILLUSTRATOR $99.75
SONG WRITER D $27.75
MASTER TYPE $27.75

CONTINENTAL

HOME ACCOUNT D $44.75
TAX ADVANTAGE $35.75
BOOK OF APPLE SOFTWARE $16.95
GENERAL LEDGER $179.95
ACCOUNTS PAY $179.95
ACCOUNTS REC $179.95
PAYROLL $179.95
PROPERTY MGMT $329.95

INFOCOM

DEADLINE $34.75
ZORK 1, 2, or 3 ..... $34.75
ENCHANTER $34.75
PLANETFALL $34.75

SPINNAKER

KINDERCOMP D $21.75
STORY MACH D $23.75
FACE MAKER D $23.75
SNOOPER TR D $29.75
SNOOPER T2 D $29.75
DELTA DRAW D $32.75
FRACTION F D $23.75
ALPHABET ZOO D $21.75
MOST AMAZING D $26.75
RHYMES & RID D $21.75

APPLE DUMPLING GX $99.75
APPLE DUMPLING 16K BUFFER $179.75
TEXT PRINTER INTERFACE $79.75

SSI

KNIGHTS OF DESERT ... $26.75
EAGLES $26.75
TIGERS IN SNOW $26.75
GERMANY 1985 ..... $36.75
BATTLE FOR NORMANDY $26.75
SHATTERED ALLIANCE ... $39.75

SIERRA ON-LINE

ULTIMA II $39.75
DARK CRYSTAL $27.75

ATARISOFT

PACMAN $25.75
DONKEY KONG $25.75
DIG DUG $25.75
CENTIPEDE $25.75
ROBOTRON $25.75

PROGRAM DESIGN

ANALOGIES $18.50
PREP FOR SAT'S $79.75
PRESCHOOL IQ BUILDER $18.75
READING COMP $18.75
VOCABULARY BUILDER ... $18.75

PEACHTREE SOFTWARE

LIST MANAGER $199.00
BUSINESS GRAPHICS $225.00
GRAPHICS LANGUAGE ... $299.00

REQUIRES COBAL RUNTIME

ACCTS PAYABLE I $495.00
ACCTS PAYABLE II $1695.00
PAYROLL I $495.00
PAYROLL II $1695.00
GENERAL LEDGER I $495.00
GENERAL LEDGER II ... $1695.00

CP/M VERSION

PEACHPACK 4 ..... $295.00
ACCTS PAYABLE III $549.00
ACCTS PAYABLE IV $899.00
ACCTS REC III $549.00
ACCTS REC IV $899.00
GENERAL LEDGER III ... $549.00
GENERAL LEDGER IV ... $899.00
INVENTORY MGMT I $549.00
INVENTORY MGMT II ... $899.00
PAYROLL III $549.00
PAYROLL IV $899.00
SALES INVOICING I $549.00
SALES INVOICING $899.00

ATARI

INFOCOM

DEADLINE $34.75
ZORK 1, 2, or 3 ..... $26.75
ENCHANTER $34.75

EDUFUN

GULP ARROW GRAPHICS $24.75
FACE FLASH $24.75

CONTINENTAL

HOME ACCOUNT D $44.75
TAX ADVANTAGE $35.75
BOOK OF ATARI SOFTWARE $16.95

SCARBOROUGH

SONG WRITER D $27.75
MASTER TYPE $27.75

EASTERN HOUSE

MONKEY WRENCH II $52.75

DON'T ASK SOFTWARE

SAM $39.75
ABUSE $22.75
WORD RACE $24.75

PROGRAM DESIGN

ANALOGIES $14.50
PREP FOR SAT'S $79.75
PRESCHOOL IQ BUILDER $16.75
READING COMP $16.75
VOCABULARY BUILDER ... $16.75
LET'S SPELL $14.75

MasterCard VISA

TO ORDER

CALL TOLL FREE

800-233-8760

In PA 1-717-327-1824

or send order to

Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740

Lyco Computer Marketing & Consultants

TO ORDER CALL US

TOLL FREE 800-233-8760

In PA 1 717-327-1824

SAVE ON THESE IN-STOCK PRINTERS

MANNESMANN TALLY

SPIRIT 80 ..... $299.00
MTL-160L $559.00
MTL-180L $775.00

NEC

NEC8023 ..... $369.00
NEC8025 ..... $699.00

PRINTER INTERFACING

AVAILABLE FOR COMMODORE VIC, APPLE, ATARI, IBM-PC, TRS-80, TI, AND OTHERS

STAR MICRONTICS

GEMINI 10X $259.00
GEMINI 15X $379.00
DELTA 10 ..... $449.00
DELTA 15 ..... $525.00
RADIX 10 ..... $575.00
RADIX 15 ..... $675.00
POWERTYPE ..... $CALL
SWEET P (Model 100) ... $549.00
STX80 ..... $149.00

SMITH CORONA TP2 ... $449.00

CITOH

GORILLA GX-100 ..... $169.00
GORILLA SERIAL $199.00
PROWRITER 8510 ..... $335.00
PROWRITER II $575.00
8600 ..... $899.00
STARWRITER $999.00
PRINTMASTER $1299.00
SHEET FEEDER $425.00
620 ..... $929.00
630 ..... $1699.00
8510 SP $499.00
8510 SCP $559.00
A10 LETTER QUAL $499.00

EPSON

RX-80 ..... $CALL
RX-80 FT ..... $FOR
FX-80 ..... $CURRENT
FX-100 ..... $PRICES

OKIDATA

80 ..... $SAVE
82A ..... $ON THESE
83A ..... $IN-STOCK
84 ..... $PRINTERS
92 ..... $CALL &
93 ..... $SAVES

PRINTING PAPER

3000 SHEETS FANFOLD $42.75
1000 SHEETS FANFOLD $19.75
1000 SHEETS LETTER ... $21.95
200 SHEETS LETTER $8.99
150 RAG STATIONERY ... $10.99
MAILING LABELS (1 in) ... $9.75
14x11 1000 FANFOLD ... $24.75

WICO COMMODORE & ATARI

JOYSTICK $16.75
RED BALL $18.75
BOSS JOYSTICK $12.75
TRACK BALL $32.75
12 FT EXT CORD $6.95
TI ADAPTER $9.95
APPLE ANALOG $34.75
IBM-PC ANALOG $34.75

commodore

CONTINENTAL 64
HOME ACCOUNT D $44.75
TAX ADVANTAGE $35.75
F.C.M. $35.75
64 USERS OF ENCYC $12.75
20 ENCYCLOPEDIA $12.75

KOALA 64
KOALA TABLET $69.75
PROGRAMMERS GUIDE ... $12.75
GEOMETRIC DESIGNS ... $22.75
LOGO DESIGN $27.75
SPIDER EATER $22.75

CARDCO

G PRINTER INTERFACE $64.75
B PRINTER INTERFACE $39.75
KEY PAD 64 ..... $29.75
LIGHT PEN $29.75
3 SLOT EXPANSION $24.75
5 SLOT EXPANSION $54.75
6 SLOT EXPANSION $79.75
PRINTER UTILITY $19.75
CASSETTE INTERFACE ... $29.75
WRITE NOW 20 ..... $29.75
WRITE NOW 64 ..... $39.75
MAIL NOW $29.75
DM-1 PRINTER $109.75
LQ-1 PRINTER $499.75

DUST COVERS

800 ..... $3.99
400 ..... $3.99
1200 ..... $3.99
410 ..... $3.99
810 ..... $3.99
600 XL $5.99
800XL $5.99
1050 ..... $5.99
1010 ..... $5.99
PROWRITER $5.99
GEMINI 10X $5.99
PERCOM $5.99
EPSON $5.99
RANA $5.99
VIC 64/20 ..... $5.99
VIC 1541 ..... $5.99

TIMEWORKS
INVENTORY $39.75
ACCOUNTS REC ... $39.75
ACCOUNTS PAY ... $39.75
GENERAL LEDGER $39.75
PAYROLL $39.75
INVENTORY MAN .. $39.75
CASH FLOW $39.75
SALES ANALYSIS .. $39.75
DATA MANAGER ... $19.75
MONEY MGR C/D ... $19.95
CHECKBOOK C/D .. $19.95
FORGET-ME-NOT .. $19.95
STAR BATTLE $19.95
ROBOT REVOLT $19.95
SPELLBOUND C/D . $19.95

SCARBOROUGH 64
SONG WRITER D ... $27.75
PHI BETA FILER $32.75
MASTER TYPE $27.75

RESTON 64
MINER 2049 ROM . $29.75

HES 64
TURTLE TUTOR $22.75
TURTLE TRAINER .. $22.75
TURTLE GRAPHICS $37.75
SOUND BOX $9.95
64 FORTH $39.95
HESMON $22.75
GRIDRUNNER $19.75
ATTACK OF M C $22.75
HESWRITER $28.75
OMNI WRITER $45.75
TYPE N WRITER $24.75
PAINT BRUSH $22.75
BENJI $25.75
HOME MANAGER .. $28.75
TIME MONEY MGR . $44.75
OMNI CALC $33.75
SWORD POINT $19.95
HES MODEM $49.95
M. MULTIPLAN $65.75

SYNAPSE 64
BLUE MAX C/D $24.75
ZEPPELIN C/D $24.75
PHAROAH'S C/D $24.75
SHAMUS C/D $24.75

PARKER 64
FROGGER $32.75
QBERT $32.75
TUTANKHAM $32.75

SPINNAKER 64
KINDERCOMP D/R $21.75
STORY MACH ROM $24.75
FACE MAKER D/R $24.75
SNOOPER TR D $26.75
SNOOPER T2 D $26.75
DELTA DRAW ROM $26.75
FRACTION F ROM $24.75
KIDS ON KEYS $24.75
ALPHABET ZOO $24.75
MOST AMAZING R $26.75

BRODERBUND 64
BANK STREET D $44.75
CHOPLIFTER R $29.95
DAVID'D MAGIC $24.95
DROL D/R $24.95
LODE RUN D/R $24.95
SEAFOX R $29.95
SPARE CHANGE D $24.95

ADVENTURE INTERNATIONAL
ADVENTURE 1-12 ea. $17.75
SAGA $27.75
PREPPIE $24.75
DISKEY $34.75
SEA DRAGON $24.75

KOALA TABLET $69.75
PROGRAMMERS G ... $12.75
GEOMETRIC DESIGN. $22.75
LOGO DESIGN $27.75
ILLUSTRATOR $99.75

INHOME
400 KEYBOARD .. $IN-STOCK

BIT 3
80 COLUMN SCREEN BOARD $259.75

PARKER
FROGGER $32.75
QBERT $32.75

SPINNAKER
KINDERCOMP D/R .. $20.75
STORY MACH D/R ... $26.75

PERCOM

AT88S1 .. $249.00
RDF40SI $269.00

SSI
KNIGHTS OF DESERT $26.75
EAGLES $26.75
TIGERS IN SNOW $26.75
COMBAT LEADER $26.75

DATASOFT
SPELL WIZZARD $34.75
TEXT WIZZARD $34.75
WORD WIZZARD $46.75
ZAXXON C/D $26.75

GEMINI 10 ..... $259.00

BRODERBUND
BANK STREET D ... $44.75
CHOPLIFTER D $32.95
DAVID'D MAGIC $24.95
DROL D $24.95
LODE RUN D $24.95
SEAFOX D $22.95
SPARE CHANGE D . $24.95
AE D $24.75

EPYX
JUMPMAN C/D $27.75
JUMPMAN JR R $27.75
PITSTOP R $27.75
TEMPLE ASPHAI ... $27.75
GATE T ASPHAI $27.75
CRUSH, C&C C/D ... $21.75

LJK
LETTER PERFECT .. $69.75
DATA PERFECT $69.75
SPELL PERFECT ... $69.75

ATARI

Computers for people.

PACMAN $29.75
DONKEY KONG $29.75
DIG DUG $29.75
CENTIPEDE ... $29.75
STAR RAIDERS $32.75
ROBOTRON $32.75
PAINT $32.75
JOUST $35.75
LOGO $69.75
ATARI WRITER $79.75
BOOK KEEPER $104.75
VISICALC $159.75
HOME FILING MGR $35.75
PILOT HOME $54.75
KEYPAD $95.75

800XL ..... $ CALL
1050 DISK DRIVE $ CALL
1050 RECORDER $74.00
1027 PRINTER $ CALL
1025 PRINTER IN-STOCK

MasterCard VISA

TO ORDER

CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825

or send order to
Lyco Computer
P O Box 5088
Jersey Shore, PA 17740

POLICY

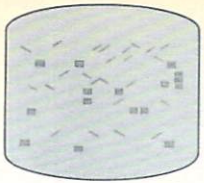
In-stock items shipped within 24 hours of order. Personal checks require four weeks clearance before shipping. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. PA residents add sales tax. All products subject to availability and price change. Advertised prices show 4% discount offered for cash, add 4% for Master Card or Visa. DEALER INQUIRIES INVITED.

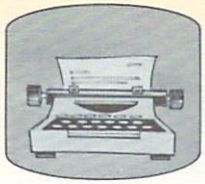
Advertisers Index

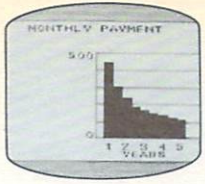
Reader Service Number/Advertiser	Page
Abacus Software	65
Abacus Software	101
102 AB Computers	87
103 Activision	12,13
104 Aspen Ribbons, Inc.	141
105 Atari Inc.	4
106 Batteries Included	31
107 Batteries Included	43
108 Broadway Computer Corporation	101
109 Cardco, Inc.	IBC
Carpenter's Creative Designs	141
Cass-A-Tapes	101
Commodore Computers	BC
110 CompuServe	7
111 CompuServe	35
ComputAbility	103
Computer Mail Order	60,61
Computer Warehouse	100
Cosmic Computers	107
Creative Software	17
Creative Software	19
Datasoft, Inc.	27
Dennison Computer Supplies, Inc.	IFC
Discount Data Supply	141
112 Disk World!, Inc.	65
Disk World!, Inc.	141
Disk World!, Inc.	141
Disk World!, Inc.	141
113 Eastern House	114
E Mart, Inc.	99
Epyx	21
Frontrunner Computer Industries	127
114 The Furniture Byte	97
Hytec Systems	29
IBM	2,3
Indus Systems	59
Infocom, Inc.	44,45
J & R Music World	97
John T. Dow	141
115 Kalglo	30
116 Lyco Computer Marketing & Consultants	142,143
Micro-Sys Distributors	63
Micro World Electronix, Inc.	99
Midwest Micro Inc.	118

Reader Service Number/Advertiser	Page
117 Nibble Notch	127
Pacific Exchanges	23
Pacific Exchanges	82
Pacific Exchanges	127
Pacific Exchanges	131
118 Parker Brothers	36
Parker Brothers	36
Parker Brothers	37
Powersoft, Inc.	23
119 Professional Software Inc.	1
120 Protecto Enterprizes	67
Protecto Enterprizes	68,69
Protecto Enterprizes	70,71
SM Software Inc.	85
SM Software Inc.	85
Spinnaker	25
Strategic Simulations Inc.	15
121 subLOGIC Corporation	9
122 Such A Deal	98
Susie Software	141
123 SWINTEC Corporation	48
3G Company, Inc.	82
Walling Co.	141
Xerox Education Publications	11
York 10	99

COMPUTE! Subscription	33
COMPUTE!'s PC & PCjr Subscription	48
COMPUTE!'s Subscriber Services	101







Save UP TO 40% on COMPUTE!

Every issue of COMPUTE! contains up to 30 new programs and games. And a year's subscription brings them to you for less than 15 cents each! Plus you'll enjoy the most useful home computer advice, ideas and information anywhere! Subscribe now at up to 40% off the newsstand price. At less than 15 cents per program, this COMPUTE! offer is too good to pass up!

☐ 1 year $24—Save 32%! ☐ 2 years $45—Save 36%!
☐ 3 years $65—Save 40%!

Name _____

Address _____

City _____ **State** _____ **Zip** _____

☐ Payment enclosed ☐ Bill me

Charge my ☐ Visa ☐ MasterCard ☐ American Express

_____ / _____

Account No. **Exp. date**

COMPUTE! brings you programs and games for the following machines: Atari, PET/CBM, VIC-20, TI 99/4A, Apple, Commodore 64, Radio Shack Color Computer, IBM PC and IBM PCjr.

348BO1

Save UP TO 40% on COMPUTE!

Every issue of COMPUTE! contains up to 30 new programs and games. And a year's subscription brings them to you for less than 15 cents each! Plus you'll enjoy the most useful home computer advice, ideas and information anywhere! Subscribe now at up to 40% off the newsstand price. At less than 15 cents per program, this COMPUTE! offer is too good to pass up!

☐ 1 year $24—Save 32%! ☐ 2 years $45—Save 36%!
☐ 3 years $65—Save 40%!

Name _____

Address _____

City _____ **State** _____ **Zip** _____

☐ Payment enclosed ☐ Bill me

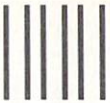
Charge my ☐ Visa ☐ MasterCard ☐ American Express

_____ / _____

Account No. **Exp. date**

COMPUTE! brings you programs and games for the following machines: Atari, PET/CBM, VIC-20, TI 99/4A, Apple, Commodore 64, Radio Shack Color Computer, IBM PC and IBM PCjr.

348BO1



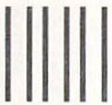
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

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



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

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

The Editor's Feedback:

Computer: ☐ PET ☐ Apple ☐ Atari ☐ VIC-20 ☐ Commodore 64
☐ TI-99/4A ☐ Timex/Sinclair ☐ Radio Shack Color Computer
☐ Other _____ ☐ Don't yet have one ...

Are you a COMPUTE! Subscriber? ☐ Yes ☐ No I would like to see:

More	Just Right	Fewer		More	Just Right	Fewer	
☐	☐	☐	Specific applications programs.	☐	☐	☐	Games.
☐	☐	☐	BASIC programs.	☐	☐	☐	Reviews of game software.
☐	☐	☐	Machine language programs.	☐	☐	☐	Reviews of business software.
☐	☐	☐	Tutorials.	☐	☐	☐	Reviews of educational software.
☐	☐	☐	Educational articles.				
☐	☐	☐	Detailed explanations of programs.	☐	☐	☐	Reviews of hardware.

What do you like best about COMPUTE!?

What do you like least?

Place
Postage
Here

COMPUTE! Magazine
P.O. Box 5406
Greensboro, NC 27435-0406

COMPUTE!'s FREE Reader Information Service

Use these cards to request FREE information about the products advertised in this issue. Clearly print or type your full name and address. Only one card should be used per person. Circle the numbers that correspond to the key number appearing in the advertisers index.

Send in the card and the advertisers will receive your inquiry. Although every effort is made to insure that only advertisers wishing to provide product information have reader service numbers, COMPUTE! cannot be responsible if advertisers do not provide literature to readers.

Please use these cards *only* for subscribing or for requesting product information. Editorial and customer service inquiries should be addressed to: COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Check the expiration date on the card to insure proper handling.

Use these cards and this address only for COMPUTE!'s Reader Information Service. Do not send with payment in any form.

COMPUTE!

101	102	103	104	105	106	107	108	109	110	11
112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166
167	168	169	170	171	172	173	174	175	176	177
178	179	180	181	182	183	184	185	186	187	188
189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275	276
277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298
299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331
332	333	334	335	336	337	338	339	340	341	342
343	344	345	346	347	348	349	350			

Circle 101 for a one year new U.S. subscription to COMPUTE!: you will be billed for $24.

Please print or type your full name and address. Limit one card per person.

Name

Address

City

State/Province Zip

Country

Please include zip code. Expiration 10/31/84. CO884

Place
Postage
Here

COMPUTE! Reader Service
P.O. Box 2141
Radnor, PA 19089

CARDCO "NOW" SOFTWARE

. . . available now for your Commodore-64™ and more!

A fine line of software developed by CARDCO for your Commodore-64 computer with all the features you should expect in much more costly software. CARDCO's "NOW" Series provides many unique and exclusive features and are packaged for easy reference, simple storage, instant recognition.

"WRITE NOW" . . . WORD PROCESSOR SOFTWARE . . . An excellent time saver, CARDCO offers the "Write Now" C/Ø2 word processor program with built-in 80 column display. You see exactly what will print. All special codes can be transmitted to printers maintaining justification. Easy full-screen editing; works with any printer.

"MAIL NOW" . . . MAILING LIST SOFTWARE . . . CARDCO's D/Ø1 "Mail Now" quickly (in memory) sorts by zip, category, name and state; fully compatible with "Write Now". Other fine features include: user-oriented; menu-driven operation; each disk supports 600 entries. Format can print single, double or triple labels across.

"SPELL NOW" . . . Cardware D/Ø4 . . . a fine program designed as a spell checker for use with "Write Now" on the Commodore-64. A 34,000 word dictionary with two additional user constructed dictionaries. Menu-driven operation for ease of use. And "Spell Now" allows you to see each misspelled word in the context of your document for correction.

"FILE NOW" . . . D/Ø5 . . . is a totally integrated, menu-driven database software package which interfaces with both the "Write Now!" for the 64 and the "Spell Now." 40K of working storage space is available with "File Now". "File Now" appears on the screen as index cards for easier manipulation of your data base; you see 5 index cards at a time. Cards are user defineable, i.e., user determines what goes where on the "index cards" and can sort by any given field. Every card has a general topic field which allows for quick sorting through cards.

"GRAPH NOW" INCLUDING . . . "PAINT NOW" . . . D/Ø6 . . . This disk-based graphic/logo generator is totally menu-driven. Allows for the development of pies, charts, bar graphs and other vivid graphic illustrations. Also has the ability to design, and print logos and high resolution pictures. "Commodore-ready"; interfaces with CARDCO'S "Write Now" Word Processor, "Mail Now", "Spell Now" and "File Now".

Write for illustrated literature and prices or see CARDCO Computer Accessories and Software wherever Computers are sold.

cardco, inc.

300 S. Topeka Wichita, Kansas 67202 (316) 267-6525

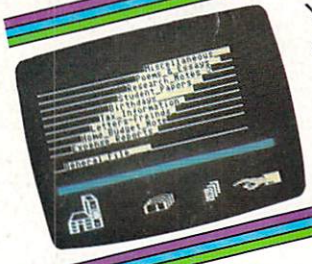
"The world's largest manufacturer of Commodore accessories."

Commodore 64 Magic Desk I

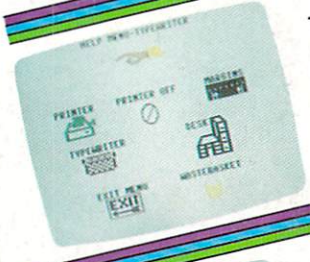
Only From Commodore—The Excitement and Simplicity of Magic Desk!

Only Commodore brings you the magic of MAGIC DESK . . . the next generation of "user friendly" software! Imagine using your computer to type, file and edit personal letters and papers *without learning any special commands!* All MAGIC DESK commands are PICTURES. Just move the animated hand to the picture of the feature you want to use (like the TYPEWRITER) and you're ready to go.

The MAGIC DESK Typewriter works just like a real ELECTRIC TYPEWRITER . . . and it's COMPUTERIZED. All the filing is *electronic*. Excellent sound effects and screen animation make typing fun, whether you're typing letters, reports or memos . . . and the built-in filing feature makes MAGIC DESK useful for keeping names and addresses, home inventory lists, insurance information and more.



Your COMMODORE 64, COMMODORE DISK DRIVE and MAGIC DESK are an unbeatable combination. Filing operations are automatically linked to your Commodore disk drive—but you don't have to know any commands—just "file" the pages you type in the file cabinet and your text is automatically saved on diskette. There are 3 file drawers with 10 file folders in each drawer and 10 pages in each folder.



To PRINT a page you've typed, just "point" at the picture of the printer and your pages are automatically printed on your COMMODORE PRINTER or PRINTER/PLOTTER. If you want to erase what you've typed, the WASTE-BASKET under the desk lets you "throw away" pages. There's even a DIGITAL CLOCK which helps you keep track of time while you're typing.

Not only is MAGIC DESK easy to use . . . it's hard to make a mistake! Just press the COMMODORE key and one of several "help menus" appears to tell you exactly what to do next. Special messages show you how the various picture commands work and help you when you make a mistake. Help messages also show you how to use the printer, filing cabinet, digital clock and wastebasket.

Another reason why Commodore is number one in the world of microcomputers—Come join us.

commodore COMPUTERS

First In Quality Software