Apple's Evolutionary New IIc

COMPUTE!

$2.95
July
1984
Issue 50
Vol. 6, No. 7
£2.25 UK $3.75 Canada
02193
ISSN0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

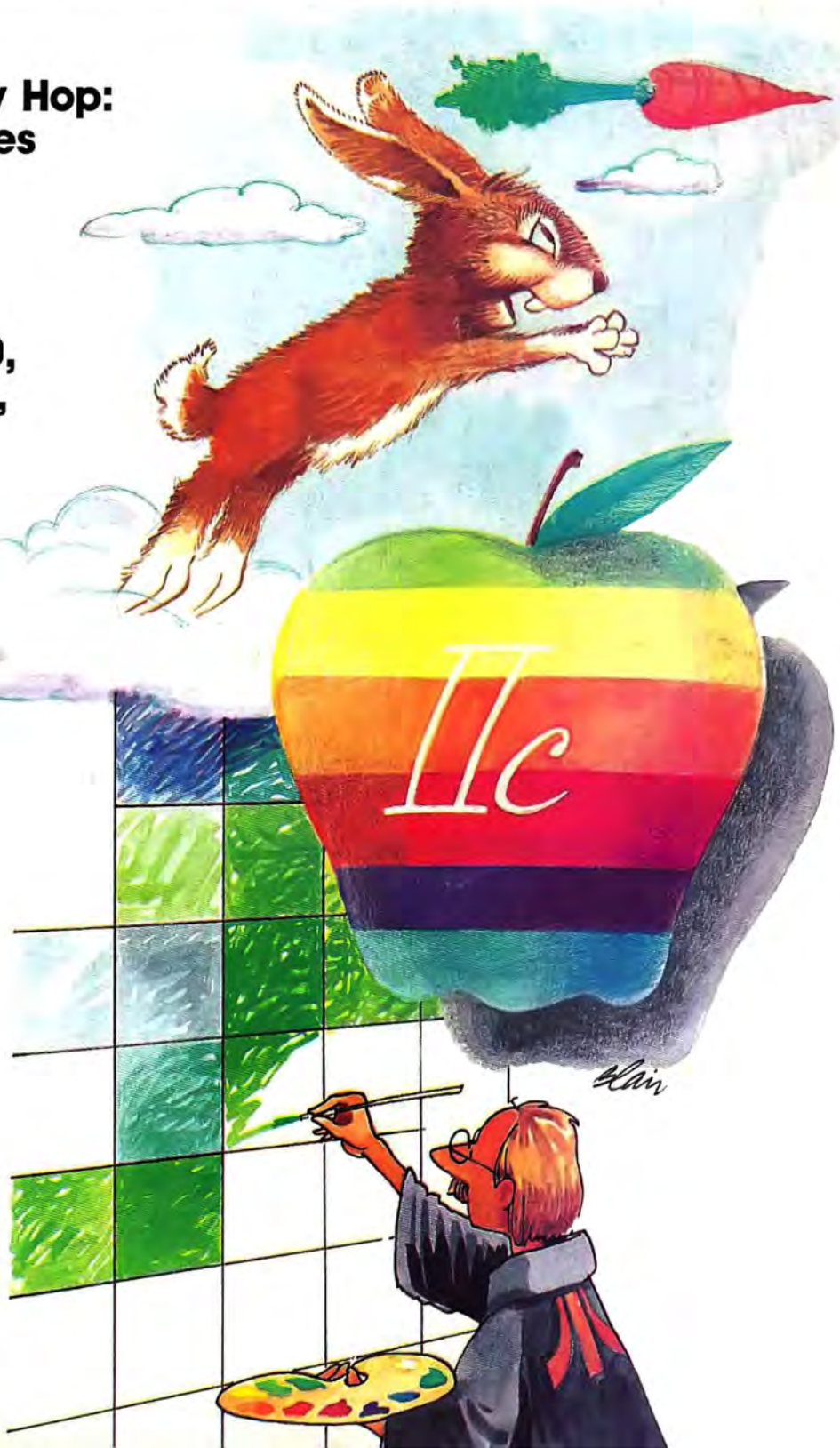
Blueberries And Bunny Hop: Two Fascinating Games For Children

Statistics For Nonstatisticians For The Apple, VIC-20, Commodore 64, Atari, IBM PC And PCjr, TI, And Radio Shack Color Computer

Commodore 64 And 1521 Disk Drive ROM Generations: All The Changes

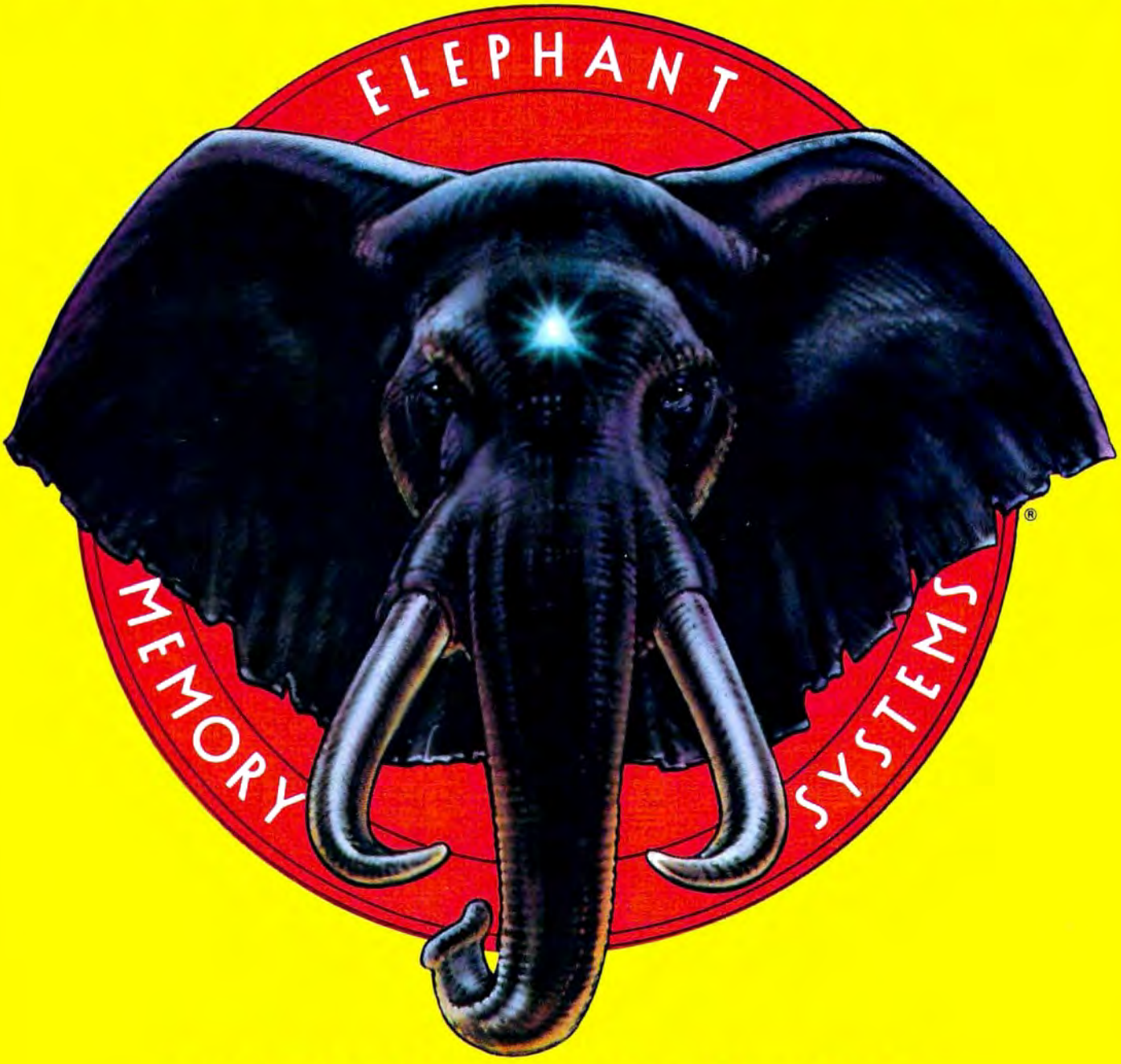
Applesoft Lister: Format Programs For Fast Debugging

Atari Artist: Create Graphics The Easy Way



0 71486 02193 07

REMEMBER.



ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5 1/4" 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.

ELEVEN THINGS YOU CAN DO ON THE VERY FIRST DAY.

Your first day with PC*jr* isn't just exciting.

It's rewarding.

This IBM® personal computer quickly hooks up to the family TV. And there's a lot you can do right from the start.

PC*jr* MAKES IT EASY

PC*jr* was designed to make the whole family feel at home with computers.

The keyboard, for example, doesn't need a connecting cord. This refreshing bit of technology—the IBM "Freeboard"—frees you to get comfortable up to twenty feet away.

And the keys are color-coded, so even a beginner can find the right key for the right job.

Right away.

1. Home Loan
2. Monthly Expenses
3. Word Race
4. Shopping List
5. Recipe File
6. Tile Game
7. Typewriter
8. Alarm Clock
9. Checkbook
10. Address Book
11. Telephone Connector

YOU'RE OFF AND RUNNING

You can start using PC*jr* as soon as you set it up. The Sampler Diskette (included with diskette-drive models) gives you eleven useful mini-programs to choose from.

Including:

A home spreadsheet to help keep your expenses in line.

An electronic address book to help you sort out who's who and who's where.

A home loan calculator that can tell you interesting things about your principal.

An easy-access file for recipes. A checkbook balancer. And a challenging word game.

Of course, the Sampler Diskette is merely a taste of what you can do with PC*jr*. You can buy easy-to-follow programs to help you write letters, plan your finances, educate the kids, file tax data—to help the whole family use its time to better advantage.

Plus, PC*jr* runs many of the IBM Personal Computer programs that run on the IBM PC and PC/XT. So you can finish at home what you start at the office, and vice versa.

And when it's time to relax, PC*jr* is always game. Plug in an entertainment cartridge and let the fun begin.

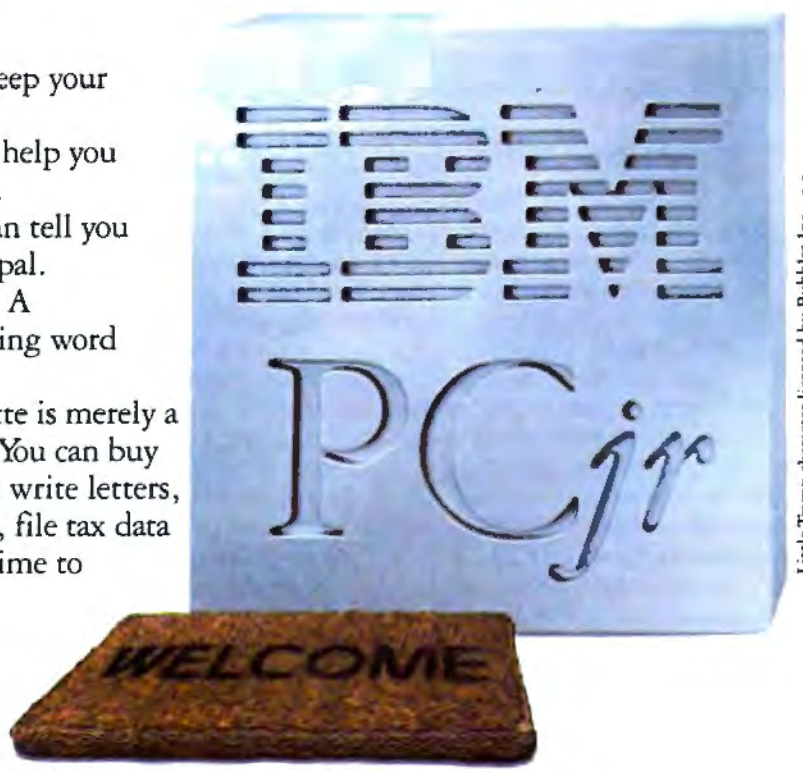
A COMPUTER THAT CAN GROW WITH YOU

PC*jr* is the most affordable of the IBM personal computers.

You can start with the 128KB model, equipped with a diskette drive, for about $1300. Or the 64KB base model, for about $700. (Prices apply at IBM Product Centers, and may vary at other stores.)

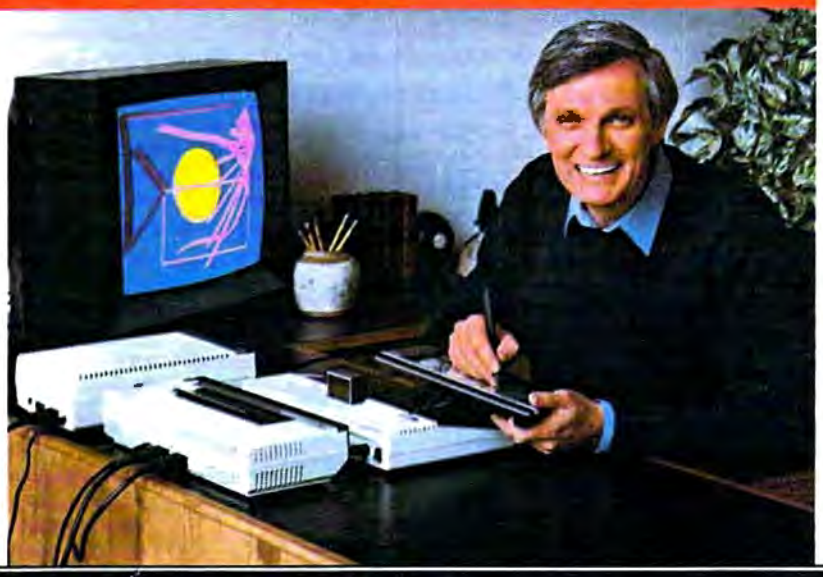
And as your needs become more sophisticated, PC*jr* can easily keep pace. With add-it-yourself options like a printer, diskette drive and internal modem for telecommunications, even the lowest-priced model can grow up fast.

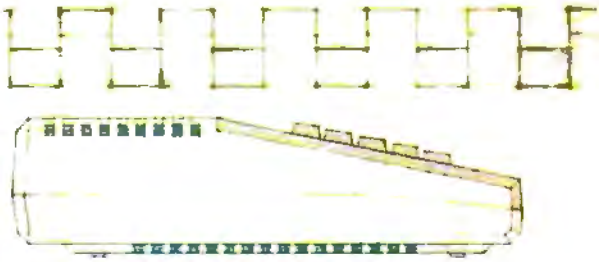
Visit an authorized IBM PC*jr* dealer or IBM Product Center and see all the things PC*jr* can do. For the location nearest you, call 1-800-IBM-PCJR. In Alaska and Hawaii, 1-800-447-0890.



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

DISCOVER COMPUTERS WITH ATARI.





Atari® presents the five greatest advances in the creative arts since someone put 72 crayons in one box.

What would Cezanne say to an electronic orange? Surely Van Gogh would go for some flowers painted in phosphors (those glowing things in your TV screen). And you bet Beethoven would be blown away by a computer synthesized symphony.

Too bad. They were all born too early. But luckily you weren't. Because Atari makes several home computer products to help you create all these things and more.

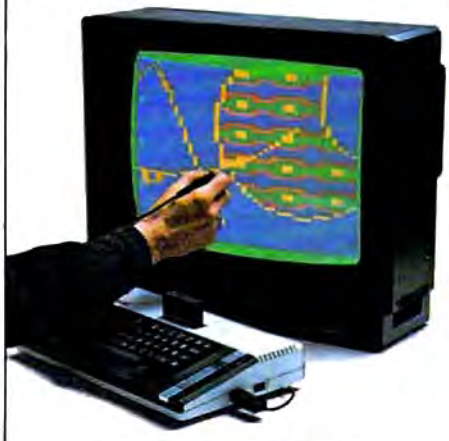
First, there's ATARI Paint,* the program that turns the joystick you already own into a computerized paintbrush that helps you explore the fascinating world of computer art.

Get the magic touch with Atari Touch Tablet.

The ATARI Touch Tablet with Atari-Artist™ software cartridge turns your TV into a magic palette of 128 dazzling colors.

The Touch Tablet works a little like an electronic slate. Hook it into any ATARI Computer and what you draw on the tablet will instantly appear on your TV screen. You can draw on the Touch Tablet with the unique electronic stylus that comes with it—or even with your finger.

And all the on-screen commands that control your Touch Tablet are labelled with simple symbols as well as words. So everyone from preschoolers to grandparents can create without going near the keyboard.



Atari Light Pen lets you write right on the screen.

ATARI Light Pen with Atari-Graphics™ software cartridge is among the best tools available for teaching hands-on computer creativity. To draw circles, rectangles, or simply fool around with freehand sketching, just press the pen to your TV screen and VOILA ...A trail of sparkling color follows it.

You can fill in your sketches with one or more of 128 colors. Or choose from over 2,800 patterns.

Roll over Beethoven, AtariMusic is here.

Sure, Beethoven took music lessons. But even he didn't get the head start you'll get with AtariMusic™ I or AtariMusic™ II. And once you understand the basics you'll be ready to move up to ATARI Music Composer® and create original compositions in four part harmony!

All of these programs were designed to get the best from your ATARI Computer, including the ATARI 800XL™ or the less expensive ATARI 600XL.™ Both machines give you unsurpassed Atari graphics and four sound channels. And whether you're painting with light or composing at the computer keyboard, you can store your creation on the ATARI 1010™ Program Recorder or the more sophisticated 1050™ Disk Drive.

And if all that doesn't convince you that our new programs are a giant step beyond crayons, consider this: the kids will never write on the wall with a computer.

*SuperBoots™ Software developed by Capital Children's Museum, Washington, D.C., licensed by Reston Publishing Company, Inc. © 1982 Reston Publishing Company, Inc. All Rights Reserved.

DISCOVER WHAT YOU AND ATARI CAN DO.

July 1984 · Vol. 6, No. 7

FEATURES

EDUCATION AND RECREATION

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 152 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

This month, Senior Editor Richard Mansfield responds to an attack on personal computing in this guest editorial.

Robert C. Lock
Editor In Chief

A few weeks ago, pundit Andy Rooney launched an astonishing attack on personal computing on CBS's show *60 Minutes.* It was astonishing because he revealed a staggering misunderstanding of computing. But first a bit of background.

There seem to be fewer curmudgeons around these days. It used to be that when an important invention was unveiled, dozens of experts could be counted on to denounce the device as interesting, but impractical. No more. Maybe it was the splitting of the atom or the moon landing. Who knows? For whatever reason, few people are now willing to publicly predict that an invention is fundamentally unimportant, useless, or impossible.

Most people, experts included, still secretly think like that about new technologies. They just won't talk about it with reporters anymore. There is a defect, a weakness in many people, which makes them unable to accept new machines and discoveries. Perhaps we could define this as futureblindness—a skewed view of the potential of new hardware.

Part of the problem is that hardware always precedes software. The car was invented before there were proper roads for it to travel on. TV sets were constructed before there were programs to watch.

So, in the past, when a major new technology was announced, futureblind "experts" would come out of their dwellings and talk with reporters. The experts would acknowledge that the new device was interesting, but that it was also impractical and hardly deserving of all the notice it was getting. Less stodgy experts might have gone so far as to envision a limited use for the device, a very limited use.

It's always amusing to read such pronouncements a decade or two later. For example, when the first telephone was demonstrated, one expert predicted that there would, in fact, be a real use for this new technology. He announced that he could even foresee a day when there would be one in every large city.

Such a prediction likely drew gasps and murmurs from the more severely futureblind in his audience. A phone in every major city was, of course, desirable, but hardly practical. After all, there would have to be a wire strung from city to city across the land. And that was beyond imagining.

It's always this way with hardware. Most people, and many experts, cannot understand that important new hardware is naked at first, but creates a powerful vacuum, eventually pulling in huge amounts of software. Few people realized that the automobile would throw webs of asphalt over entire continents. Or that nets of telephone wires would cover our cities. Before those webs and nets were in place, the car and the phone seemed, if not frivolous, at least pretty limited.

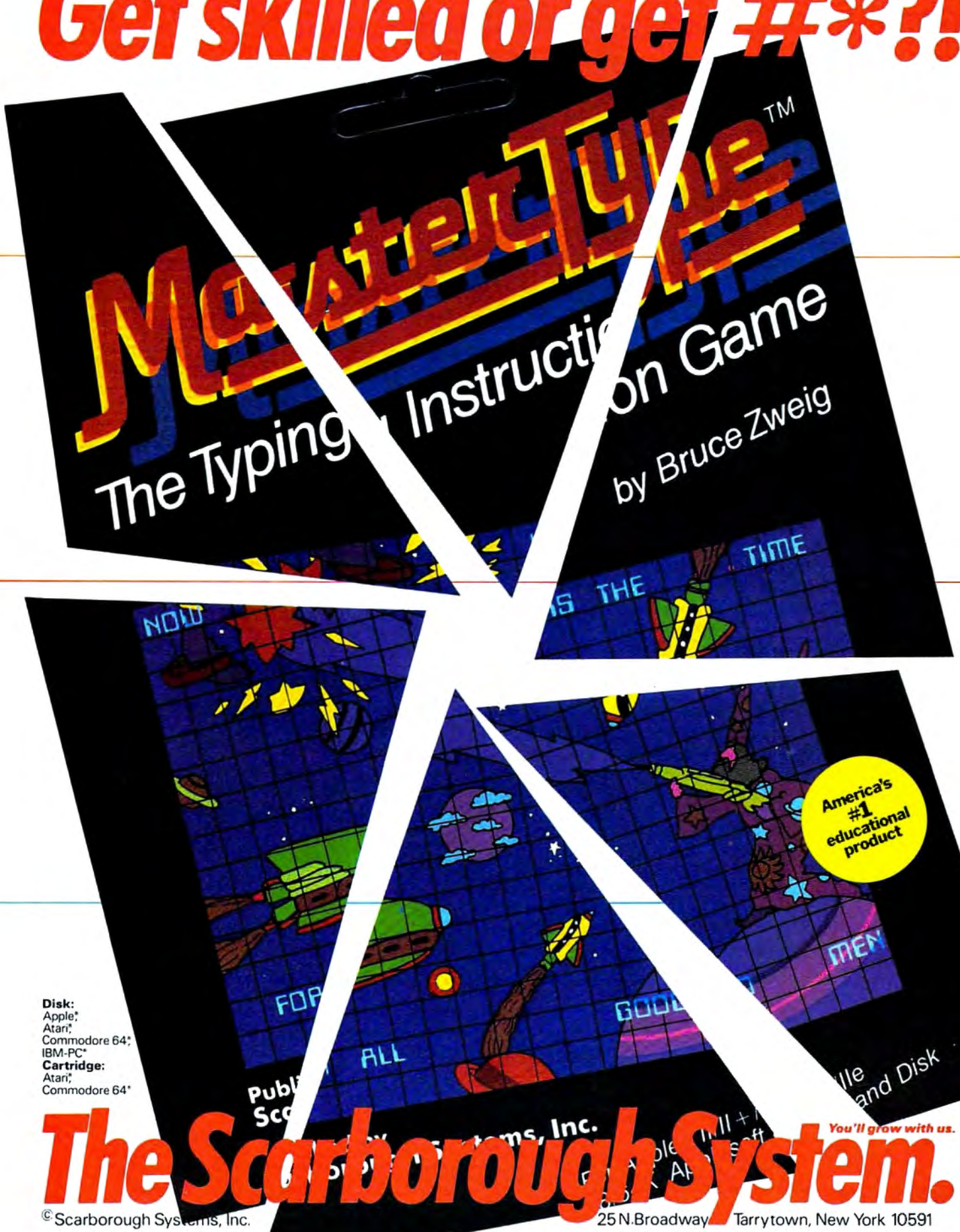
Yet these days very few experts are willing to reflexively denounce new technology. Burned once too often, the average expert will now either refuse to comment or make some mild, rhetorical, anticomment like "I'm excited about the prospects of this, but I must wonder where it will lead us in our modern society." Since remarks like this say nothing whatsoever, they're safe enough.

So it was with mixed emotions that we listened to Andy Rooney attack personal computers on *60 Minutes.* On the one hand, it was touchingly nostalgic to watch him denounce technology in the traditional way by confusing hardware with software. On the other hand, his stance was so grossly confused that his pronouncements cannot take a place alongside the classic, the truly great historical failures of vision.

He decided to test the value of personal computers by seeing if he could make corrections faster via word processor or pencil and paper. He timed himself and found that his pencil was indeed faster. Anyone remotely familiar with computers would immediately say, "So what?" This was not a test of word processing, much less of computers in general. But Rooney went on to draw several increasingly bizarre conclusions about computing.

Curmudgeons do serve a purpose beyond their amusement value. They are a healthy balance against promoters' hype and uncritical technophilia. But an essential prerequisite of any good denunciation is that, at least on the surface, it must appear to make some sense.

Get skilled or get #*?!



Disk:
Apple®
Atari®
Commodore 64®
IBM-PC®
Cartridge:
Atari®
Commodore 64®

The Scarborough System.

You'll grow with us.

25 N. Broadway Tarrytown, New York 10591

Publisher	Gary R. Ingersoll
Editor In Chief	Robert C. Lock
Director of Administration	Alice S. Wolfe
Senior Editor	Richard Mansfield
Managing Editor	Kathleen E. Martinek
Production Director	Tony Roberts
Production Editor	Gail Walker
Editor, COMPUTE!'s PC & PCjr Magazine	Tom R. Halfhill
Editor, COMPUTE!'s GAZETTE	Lance Elko
Technical Editor	Otis R. Cowper
Assistant Technical Editors	John Krause, George Miller
Program Editor	Charles Brannon
Features Editor	Selby Bateman
Assistant Editors	Dan Carmichael, Robert Sims, Todd Heimarck, Blake Lambert, Robert Alonso
Editorial Assistant	Kathy Yakal
Programming Supervisor	Patrick Parrish
Assistant Programming Supervisor	Gregg Peele
Editorial Programmers	Jeff Hamdani, Kevin Martin, Chris Poer, Tim Victor
Programming Assistants	Mark Turtle, David Florance, Kevin Mykytyn
Copy Editors	Juanita Lewis, Joan Rouleau
Proofreaders	Ethel Silver, Dwight Smith, Karen Uhlendorf
Administrative Assistants	Vicki Jennings, Laura MacFadden, Julia Fleming, Susan Young
Associate Editors	Jim Butterfield, Toronto, Canada Harvey Herman, Greensboro, NC Fred D'Ignazio, 2117 Carter Road, S.W., Roanoke, VA 24015 David Thornburg, P.O. Box 1317, Los Altos, CA 94022
Contributing Editor	Bill Wilkinson
COMPUTE!'s Book Division	
Editor	Stephen Levy
Assistant Editors	Gregg Keizer, Stephen Hudson
Assistant Managing Editor	Randall Fosner
Artists	Janice Fary, Debbie Bray
Director, Books Sales & Marketing	Steve Voyatzis
Assistant	Carol Dickerson
Production Manager	Irma Swain
Art & Design Director	Janice Fary
Assistant Editor, Art & Design	Lee Noel
Mechanical Art Supervisor	De Potter
Artists	Leslie Jessup, Cindy Mitchell
Typesetting	Terry Cash, Debi Thomas
Illustrator	Harry Blair
Director of Advertising Sales	Ken Woodard
Advertising Coordinator	Patti Williams
Assistant	Joyce Margo
Advertising Accounts	Bonnie Valentino
Promotion Manager	Mindy K. Kutcher
Subscriber Services Supervisor	Patty Jones
Assistants	Chris Patty, Christine Gordon, Sharon Sebastian, Rosemarie Davis
Dealer Sales Supervisor	Fran Lyons
Assistants	Gail Jones, Sharon Minor, Rhonda Savage
Individual Order Supervisor	Dorothy Bogan
Assistants	Judy Taylor, Lisa Flaharty, Anita Roop, Debi Goforth, Jenna Nash, Elizabeth White, Sybil Agee, Mary Hunt, Gayle Benbow, Betty Atkins, Sandra Jenkins
Shipping & Receiving	Jim Coward, Larry O'Connor, Dai Rees, John B. McConnell, Eric Staley, Sam Parker, Eddie Rice, David Hensley, John Archibald, Mary Sprague (Mail Room Coordinator)
Data Processing Manager	Leon Stokes
Assistant	Chris Cain
Vice President, Finance & Planning	Paul J. Megliola
Director, Finance & Planning	R. Steven Vetter
Accountant	Robert L. Bean
Credit Manager	David F. Carpenter
Purchasing Manager	Greg L. Smith
Financial Analyst	Karen K. Rogalski
Staff	Linda Miller, Doris Hall, Jill Pope, Anna Harris, Anne Ferguson, Pat Fuller, Tracey Hutchins, Susan Booth, Sybil Agee

Robert C. Lock, Chief Executive Officer
Gary R. Ingersoll, President
Paul J. Megliola, Vice President, Finance and Planning
Debi Nash, Executive Assistant
Cassandra Robinson, Assistant

Member ABC Audit Bureau of Circulations

MPA

Coming In Future Issues

How To Choose The Best Educational Software

A Sophisticated Chess Program For Atari And Commodore

Missile Math And Space Tunnel—Exciting Games For Multiple Computers

Commodore Disk Autoboot And Pattern Matching

Atari Speedreading

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

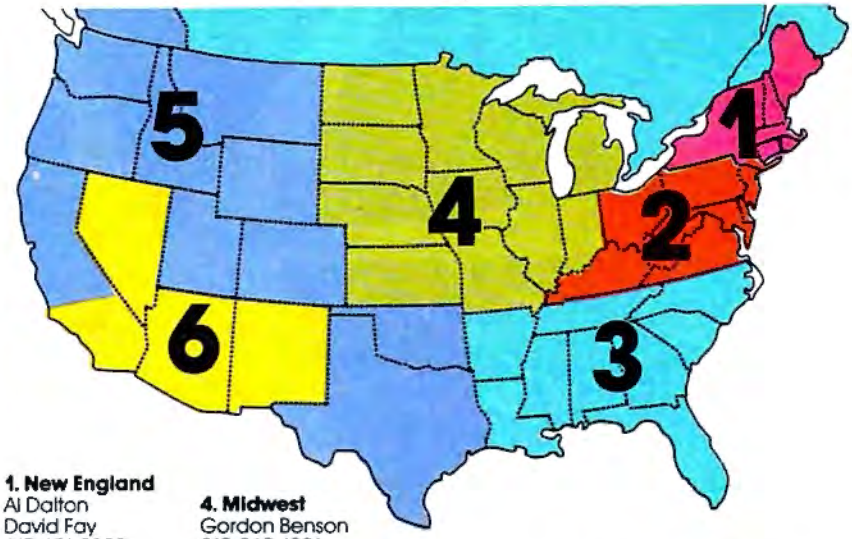
COMPUTE! Circulation Dept.
P.O. Box 914
Farmingdale, NY 11737

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

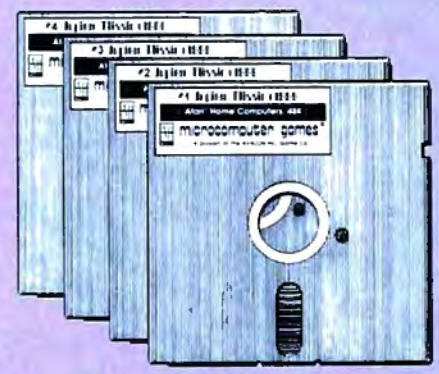
COMPUTE! Home Office 919-275-9809.

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

Jupiter Mission 1999™

Scott Lamb's Interactive Space Fantasy Adventure

Jupiter Mission 1999 is Avalon Hill's Trademark Name for its Microcomputer Space Adventure Role Playing Game.



Ship Status Display:
Damage report

Navigation Display:
Used to plot course

Science Lab Display:
Probe report

Science Lab Display:
Jupiter system diagramatic

® Trademark of
Warner Communications

RRRING!!! RRRING!!! RRRING!!!

Too early on a chilly January morning, I was jarred awake by the noxious blaring of my traitorous doorbell. As I moved to accost the unknown aggressor, with the full force of semi-conscious wrath, I pulled on my robe and lost my dignity to the pain of a stubbed toe. Now fully awake, I opened the door, prepared to educate the mysterious interloper on the meaning of manners. My determination to this end was somewhat shattered when I saw two large men clad in long overcoats and wide-brimmed hats. Instinctively, I tried to slam the door. My retreat to safety was denied by the advance of the strangers. Before I could protest their entry, my vision was drawn to the shining silver badges that hung from their now unfolded wallets. They were government agents.

Hesitantly, trying to remember any crimes that I had ever committed, I invited them into my home. At their request, I produced my driver's license and other forms of identification. After examining these credentials, they asked me to pack a bag for an extended journey. After some protest and argument, I was made to understand that my options in this matter were less than limited. My country needs me, they said—with the clear implication that either I pack and dress or I take an extended journey in my robe.

This is how my adventure began. From my cold apartment, I was taken to a towering vehicle for an emergency mission to Jupiter. My very life on the line and, possibly, the survival of the planet Earth as well, and only God knows what other kind of perils await.

JUPITER MISSION 1999 is a highly detailed role-playing space adventure game that includes challenging arcade segments and mystifying puzzles. **Four separate program disks** are enclosed to test the creative imagination of the advanced computer gamer. **JUPITER MISSION 1999** is ready to run on your Atari® Home Computer with 48K memory and one disk drive. **$50.00**

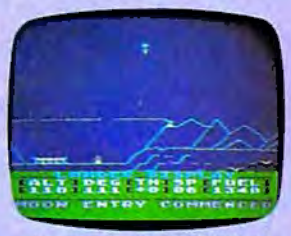
Available at leading computer software dealers, or call toll-free for further information: 1-800-638-9292

microcomputer games®

A Division of **The Avalon Hill Game Company**

4517 HARFORD ROAD, BALTIMORE, MD 21214 1-800-638-9292

Porthole View
of Jupiter and a moon



Lander Approach Display:
note descending spaceship

Exploring an
Alien Complex

Exploring an
Alien Space Station

Joystick required

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

What's An Algorithm?

I have frequently seen the word *algorithm* and would like to know what it means.

I've also heard that you should have a voltage spike protector, because when appliances are turned on, they can produce a surge in the electricity in your house. Is it absolutely necessary to have one?

Andrew S. Hartman

An algorithm is a set of steps or relationships which, taken as a whole, solve a problem. For example, regular pay = regular hours * regular rate of pay *is a standard algorithm that might be found in a payroll program.*

Or: 1. Have user INPUT *his height 2. Look up correct weight for this height. This would be an algorithm for telling people how much they should weigh.*

Using a voltage spike protector couldn't hurt. The electricity in the average home is subject to periodic surges from various sources. A lightning strike, for example, can introduce very high voltage for a few microseconds.

If you see the lights in your house flickering often, you may need one. On the other hand, many people have operated their unprotected microcomputers for years without any problems.

A spike protector is a kind of insurance.

The Commodore Internal Clock

I am trying to write my own game program. I have already programmed my screen display, but I would also like to include an onscreen timer. None of my references mention how I can get a simple TIME REMAINING: XXX SECONDS display on the screen. I have tried loops that subtract 1 from 1000 and print the results on the screen, but so far they either scroll the game display off the screen, or clear the screen. Please help.

Mark Adkins

Try this BASIC line:

```
PRINTTI$
```

This will print a six-digit number to the screen. The format is HHMMSS where HH = hours, MM = minutes, and SS = seconds. Unless you have reset the timer yourself, the six digits you see will reflect how much time has elapsed since you turned on your computer.

To set the timer, use the same HHMMSS format in this manner:

```
TI$="HHMMSS"
```

For example, TI$ = "123335" would set the clock to 12:33 and 35 seconds. Now enter TI$ = "000000" and PRINTTI$ to see the value changed. Setting TI$ with any value between 000000 and 235959 will start the clock running with that value. Enter and RUN the following short BASIC program and you'll be able to watch the clock as it's running:

```
20 PRINT"{HOME}"TI$:GOTO20
```

Using the TI$ function to create a timer can be done with an IF–THEN statement. For example, if you desire a 10-second timer, set TI$ to 0, then check for the ten-second limit with: IF TI$ = "000010" THEN... (action desired). *Remember that TI$ returns a string, and its lowest value is seconds.*

The TI command is much like the TI$ command except it returns values in seconds and fractions of seconds. Enter

```
PRINTTI/60
```

The numeric *value returned here is seconds in the format XX.XXXXXXX. Programming a timer with the TI command is much the same as with TI$. The TI value is set with the TI$ command. For example, to set TI at 60 seconds, you would enter: TI$ = "000100" (one minute). To program the same 10-second timer, you would set TI$ to zero then check the TI variable with* IF TI/60 =›10 THEN (action desired).

The problem of the scrolling screen display can be solved with cursor controls within the PRINT statement. For example, you can use the home (upper left corner of the screen) position as a starting point. Each time you want to print your score, timer, etc., simply use the cursor controls to move to that line, print the display, then move back home.

You bypass the bumps, detours and pitfalls of computing the moment you reach for Maxell.

The road to floppy success is paved with our Gold Standards.

Maxell speeds your success in computing. Helping you avoid traps that can block the way to information you've stored. After all, our disk is an industry leader in error-free performance. Performance backed by a lifetime warranty.

Consider this: Disks travel through a disk drive where heat builds up. And up. So Maxell designed its protective outer jacket to defy 140°F. The disk keeps its shape and keeps your information on track.

How good is Maxell Gold? We're the disk that many floppy drive manufacturers trust to put new equipment through its final paces. And the unqiue way we pack our oxide particles and bind them together means quality for the long run.

Dropouts? Disk errors? Just pass them by. You're on the Gold Standard.

maxell®

IT'S WORTH IT.

Maxell Corporation of America, 60 Oxford Drive, Moonachie, N.J. 07074 201-440-8020

Atari 400 RAM And Tape

1. Where do you find the RAM that's required to run most programs?

2. Is there a program that will convert Commodore programs to run on the Atari 400?

3. What kind of tapes should I use for the Atari 410 cassette recorder? Every one I try gives me an ERROR –143 when I try to CLOAD from them.

Mike Meyer

1. RAM is user memory, built into your machine. Your computer is equipped with a certain amount, which can usually be expanded (the Atari 400 comes with 16K). Software packages designate the amount of RAM they require for the program and storage, with a notice such as "48K required." Match this with the amount of RAM you have in your machine to see if you have enough. Most large arcade games require the maximum memory your machine can support, such as 48K or 64K. Most cassette-based programs require only a minimum configuration, such as 16K. Don't buy a program that requires more memory than you have—it probably won't work.

2. No conversion programs exist—and there will probably never be any. The Commodore and Atari computers have some things in common: the same type microprocessor (6502 or 6510), a similar BASIC, and similar graphics capabilities. But the differences are overwhelmingly different. Since many programs are inextricably bound to the hardware, no program can be written to reconcile all the differences. An emulator in hardware, essentially a computer on a cartridge, is the only avenue worth pursuing. We know of no Commodore emulators for the Atari computers. As you gain experience, you may want to try converting individual BASIC programs yourself.

3. First, use short tapes. Long, 60-minute tapes are thinner, and are more prone to flaking, binding, and stretching. Although you may not notice these problems with audio tapes, computers can be much more exacting. For the Atari, use a good-quality audio tape. Computer digital tapes are recommended for machines like the Commodore 64 or TI-99/4A, but you should use only audio tape on the Atari, due to the storage technique. Still, if you are using your recorder properly (CSAVEing and CLOADing past the blank leader), you will still be able to get some tapes to work. We've seen tape of marginal quality used with success. Try cleaning and demagnetizing your tape heads. If that doesn't work, you will need to get your recorder replaced or adjusted (this can sometimes be done in audio stores).

Commodore Disk Drive Device Number Update

I am the owner of a 64 and two 1541 disk drives, and was therefore quite interested in the item you published about changing (1541) device numbers in "Readers' Feedback" (April 1984).

Unfortunately, the picture you showed has little relationship to the circuit boards in the newest models of the 1541 disk drive (light brown case).

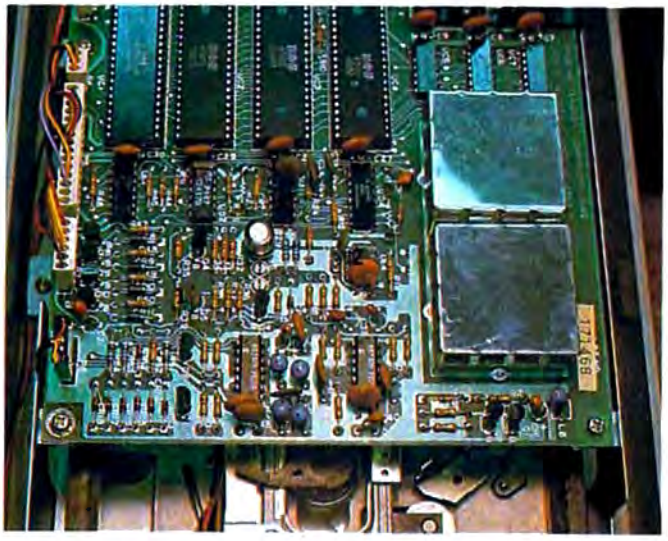
I would appreciate your printing a note showing the jumper locations for the newer model 1541s.

Eric H. Larson

As you pointed out, there are two versions of the 1541 disk drive. The older version, usually with a white case, contains a long circuit board that runs the length of the drive. The newer version, usually in a brown case, contains a shorter board.

The method for changing device numbers on the newer 1541s is the same as we described before, except for the position of the jumpers.

Unplug the drive from the wall and then carefully unscrew and remove the top half of the brown case. Then remove the large silver colored shield inside the disk drive. This will expose the circuit board. As you view the drive with the front toward you, the jumpers are near the center of the circuit board toward the front (see picture below).



The two jumpers look like round spots of silver solder (approximately ⅜-inch diameter each) separated into halves, with the two halves connected by a thin strand of solder. To change the device numbers of the 1541 disk drive, the thin strand connecting the two halves should be scratched away on one or both of the jumpers. As you look from the front of the drive, the jumper nearest the front is jumper number 1, and the rear is jumper 2.

As stated in the 1541 user's manual, cutting jumper one, jumper two, or both, produces different numbers. Following is a chart showing the different combinations.

Jumper Cut	New Device Number
None	8
1	9
2	10
1 and 2	11

WE CREATED IT.
FAMILY COMPUTING RATED IT. ★★★★
INTRODUCING "4 STAR" PUZZLE MANIA.

Puzzle Mania™ is a challenging jigsaw program for puzzle lovers.

Puzzle Mania is also an inspired jigsaw program for puzzle creators.

Because in addition to the seven great puzzles on the disk, you can create your own pictures on the screen, paint them in sixteen different colors and let the program turn them into puzzles to save on a separate disk.

Each Puzzle Mania puzzle (including your own creations) can be played on six levels of difficulty. There's help when you need it. And all turns are tallied. So you can turn the puzzle play into competitive play. If you have to part with a puzzle before it's finished, there's a stop-and-save feature built into the program, too.

Reader's Digest Software created Puzzle Mania for kids and their friends and their parents and their grandparents and everybody else who likes fun and games. Look for it at your software store or call Customer Service at 1-800-431-8800. (NY: 1-800-262-2627; AK, HI: 914-769-7000; Canada: 514-934-0751).

SOFTWARE GOOD ENOUGH
TO GO OUT AND BUY A COMPUTER FOR.

R E A D E R ' S D I G E S T

Warranty information available upon request by writing to: Reader's Digest Services, Inc., Microcomputer Software Division, Pleasantville, N.Y. 10570. Puzzle Mania runs on Apple® II, II Plus, IIe, IIc (48K), Commodore® 64, IBM® PC and PCjr. Disk drive and color monitor required. Family Computing is a publication of Scholastic, Inc.
Apple is a registered trademark of Apple Computer, Inc. Commodore is a registered trademark of Commodore Electronics Limited. IBM is registered trademark of International Business Machines Corp.

Remember, though, that unless you're handy with a soldering iron, changing the device number via hardware modifications is permanent.

Also remember that—although not specifically stated in the user's manual—opening the disk drive and performing these modifications yourself may void the warranty. Consult your local Commodore representative and thoroughly read the user's manual before attempting any of these modifications.

If you feel uncomfortable changing the device number using the hardware method, the same thing can be achieved with a software (program) modification. Here is the procedure:

1. *Turn off all disk drives.*
2. *Turn on the disk drive whose device number you want to change.*
3. *Type and enter the following commands:*

```
CLOSE 15: OPEN 15,8,15
PRINT#15,"M-W",CHR$(119)CHR$(0)CHR$(2)CHR$
  (n+32)CHR$(n+64)
CLOSE 15
```

Change the n *in the PRINT#15 command above to the device number you want to assign to the disk drive. It might be best to limit the device numbers to the range between 9 and 14.*

Now you can turn on the other drive(s), and start processing.

Note the syntax of the PRINT#15,"M-"... *command. Contrary to the instructions in the 1541 user's manual, do not include the colon after the memory-write (M-W) command. If the colon is included, the device number change will not be successful.*

The 1541 demo disk that was included with your drive also contains a program to change device numbers. LOAD the program DISK ADDR CHANGE *then RUN. The user prompts will tell you what to do.*

VIC Animation

I am 11 years old and I own a VIC-20. My friend owns a Commodore 64. We both make graphics on them. He has the advantage of sprites, but I have figured out a way for the VIC-20 to have a form of sprites. The VIC-20's graphics are made from top to bottom, which allows vertical sprites. Here's a demonstration program.

Bryan D. Stanton

```
5 DIM A(8):K=7167
10 PRINT"{CLR}"
15 POKE36869,255
20 FORM=7168TO7311:POKEM,0:NEXTM
30 FORX=1TO18:PRINTTAB(9);CHR$(63+X):NEXT
   X
40 FORN=1TO8:READA(N):POKEK+N,A(N):NEXTN
50 FORR=1TO137
55 FORL=1TO8:POKEK+L,A(L):NEXTL:POKEK+1,0
   :K=K+1
60 NEXTR
65 PRINT"{CLR}":POKE36869,240:END
70 DATA 60,24,24,24,255,126,24,24
```

Joystick To Keyboard Control On The TI

Many of your TI-99/4A games require a joystick. Unfortunately, I don't own one. Could you provide a routine that would enable me to convert these programs to keyboard control?

Mike Burgin

Several approaches can be taken to convert a program from joystick to keyboard control on the TI. Probably the simplest approach, in console BASIC, is to GOSUB to a keyboard subroutine whenever the JOYST subprogram is CALLed.

You should locate this keyboard subroutine at the beginning of the program, to speed execution. Let's put such a subroutine at line 10. The entire routine will occupy four lines beginning at line 10, so RESequence your program to begin at line 50.

*Next, find where the subprogram JOYST is CALLed within the program. The general form for this statement is CALL JOYST (*n*,X,Y). Here,* n *refers to the joystick number (either 1 or 2) while X and Y are values returned based on the joystick position.*

X and Y may be represented by any legitimate numerical variable name. Note the variable names used for X and Y in the CALL JOYST statement and then replace this statement with GOSUB 10.

Then, type in the following lines:

```
5 GOTO 50
10 CALL KEY(0,K,SS)
20 X=((K=67)+(K=68)+(K=82))*-4+((K=83)+(K
   =87)+(K=90))*4
30 Y=((K=69)+(K=82)+(K=87))*-4+((K=67)+(K
   =88)+(K=90))*4
40 RETURN
```

Now, substitute the variable names from the CALL JOYST statement into the above subroutine for X and Y. Also, if K and SS are used in the main program, you may need to name them differently here.

Just as with the CALL JOYST statement, X and Y will be returned as –4, 0, or +4 in lines 20 and 30. The standard arrow keys (E, S, D, and X) are tested for in this routine along with W, R, Z, and C for diagonal movement.

Providing a routine for keyboard control in Extended BASIC is even easier. Since we can write our own subprogram (using SUB), we no longer need worry about the variable names for X and Y in the main program. Variables used in a subprogram are local to that subprogram.

Our subprogram, which we'll call JOY, must be placed at the end of the program. Assuming there's room above line 999, type in the following:

```
1000 SUB JOY(Z,X,Y)
1020 X=((K=67)+(K=68)+(K=82))*-4+((K=83)+
     (K=87)+(K=90))*4
1030 Y=((K=69)+(K=82)+(K=87))*-4+((K=67)+
     (K=88)+(K=90))*4
1040 SUBEND
```

Compumate 2100 from SWINTEC

IT'LL INTERFACE WITH YOUR COMPUTER AND YOUR BUDGET

SWINTEC'S Compumate 2100 letter-quality copy printer has built-in parallel *and* serial interfaces that make it compatible with your computer . . . no matter whose computer you use. What's more, its very modest price tag of $649 is no more than you'd pay for an inexpensive dot matrix printer.

It adds instant letter-quality word processing to your computer capabilities. It handles correspondence . . . prints out charts and financial projections . . . in fact, gives you a sharp, clear hard copy of anything in the computer.

Microprocessor-controlled Compumate has a logic-seeking algorithm that skips over white space to print up to 20 characters a second. The 100-character daisy wheel print element is available in any one of the 10 most wanted type faces. More features: 10, 12 and 15 pitch selections and a 256-character buffer, plus LED displays tell you when your Compumate is at the ready. The Compumate 2100 requires no special training to operate. It has its own simple self-testing procedures and is backed by the renowned SWINTEC network of dealer service professionals.

Why pay more for a printer and get much less? Join the switch to SWINTEC. At your dealer's now, or return the coupon.

Dealers, too, are invited to contact us about carrying the Compumate and the full SWINTEC line of quality office equipment including electronic typewriters and calculators.

☐ Send me more information about the Compumate 2100.
☐ Have a dealer contact me for a demonstration.
☐ Contact me about becoming a SWINTEC dealer.

NAME _____

COMPANY/DEALERSHIP _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE _____

swintec™
CORPORATION

23 Poplar Street
East Rutherford, New Jersey
201/935-0115 Outside New Jersey, call toll-free 800/225-0867

Next, in the main program, change CALL JOYST(n,X,Y) to CALL JOY(n,X,Y) so that our keyboard subprogram will be CALLed rather than the system joystick subprogram (n is 1 or 2).

Last, for either console or Extended BASIC, check to see if the fire button is used. You should find a statement of the form CALL KEY(n,K,S) in the program (n is 1 or 2). Shortly thereafter in the program, a check for the value of K will be made. If K is equal to 18, then the fire button has been pressed.

With keyboard control, we can use the space bar rather than the fire button. Change n (which is 1 or 2) to 0 in the appropriate CALL KEY(n,K,S) statement. Also, change 18 to 32 in the subsequent check for the value of K.

Hex-To-Decimal Conversions

As a faithful reader of your magazine, I'd like to say that I'm surprised at how many computer hobbyists still have not found a simple decimal-to-hexadecimal conversion program. And I haven't noticed one in any issue of your magazine, so I've written this short BASIC program to do the conversions. It will work on most computers with little or no modification.

Frank Sgabellone

```
10 A$="0123456789ABCDEF":INPUT"DEC/HEX";A
   :B=1:C=9:D=16↑C:PRINTA;" = $";:A=A+1
                                  :rem 107
20 IFA-D>0THENA=A-D:B=B+1:GOTO20 :rem 156
30 PRINTMID$(A$,B,1);:B=1:C=C-1:D=16↑C:IF
   C>-1THEN20                     :rem 235
40 PRINT"{5 SPACES}":GOTO10         :rem 9
```

Hexadecimal numbers are widely used in machine language since they are more convenient for that kind of programming than the normal decimal numbers.

Compilers For The 64

I would appreciate some clarification on compilers. I have seen advertisements for several compilers (DTL-BASIC, and Metacompiler for Forth) and would like to know if they actually produce ML code that will run on any 64.

In other words, can I write a program in BASIC or Forth, run it through the compiler, and have ML code that will run on another 64 that doesn't have access to the compiler?

Paul Filiant

There are two types of compilers: those that produce native code *(machine language), and those that generate* pseudocode *(P-code). P-code compilers translate the source program into another, smaller, faster language. This pseudocode must still be interpreted, like BASIC, but it's interpreted much more quickly. Also, P-code interpreters can run the same P-code program on many machines, whatever the microprocessor used. But to run a P-code compiled program, you must have a copy of the P-code interpreter.*

Other compilers generate true machine language. This has the advantage of speed, if not portability. The object code produced by the compiler needs a set of general-purpose subroutines. Otherwise, the compilation of PRINT would expand into a large chunk of machine language each time it is used. Instead, it is more memory-efficient to compile PRINT into a subroutine call to the general PRINT routine. The set of subroutines required is called the runtime package, and is included in the compiled program.

A compiler generally produces a complete program that will run on any machine, without the compiler itself. However, we now run into the problem of copyright. You have written and therefore own the rights to the original, uncompiled program, but who owns the compiled program? You might think you retain the copyright, since compiling is something like translating a book into a different language.

However, you don't own the runtime package. Some companies require you to pay a royalty for selling the compiled program. Other companies require a special security key to run the compiled program. (A security key prevents a program from running without it. It is usually a ROM chip or some device that plugs into a joystick or cassette port.) This is like a royalty; you must buy security keys for every copy of the compiled program you distribute. Still other companies give you the freedom to distribute your compiled code, as long as you include a notice specifying that it was compiled with their product. Be sure you understand what copyright rules are enforced by the compiler company. If in doubt, write them.

Reading The Atari 800 PIA Registers

I am 16 years old and own an Atari 800. Currently, I am trying to use the joystick ports for certain I/O applications. So far, the only problem I've encountered is speed. The registers that store input information from the joyports are updated only every sixtieth of a second. This is too slow for me. Is there any way to read the joystick ports at a faster rate?

Christopher Terpin

Instead of using the shadow locations for the joysticks, you can read the joystick ports directly from the data direction registers in the 6520 PIA chip. These are truly general-purpose input/output ports, with one byte used for two joysticks. Each bit can be programmed independently for input and output. Complete information on this can be found in the Atari Hardware Reference Manual. *Some information is also found in* Mapping the Atari, *available from COMPUTE! Books. In the meantime, examine the information found at $D300.*

Get the jump on the weather-man 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!

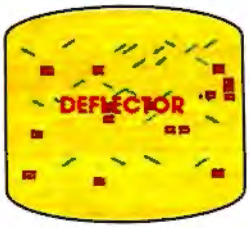
Cut your energy costs by monitoring your phone, electric and gas bills.

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

"Filer" Modification

I have really enjoyed using the "Filer" program which appeared in October 1983 COMPUTE! "Beginner's Page." However, I would like to modify it to store and search any number of items (defined by variable T in line 1) without having to update T each time another item is added. Can you help me with this?

Richard Hamilton

You can allow for a varying number of items in your file by making just a few modifications to this program. First, delete line 1. Then, adjust the DIMension statement in line 3 to some maximum number of entries—say 50. Next, add the following lines:

```
10 I=I+1:READ A$(I),B$(I),C$(I)
12 IF A$(I)<>"END" THEN 10
13 T=I-1
522 DATA END,0,0
```

In line 10, the variable I is the number of the item being READ. Line 12 checks for the end-of-data entry in line 522. If the end-of-data flag is not observed in line 12, the counter I is incremented, and another item (title, date, and author as A$(I), B$(I), and C$(I)) is READ. Otherwise, no more DATA is READ and T, representing the actual number of items, will be set in line 13.

What Is A Scratch Disk?

I have a 1541 disk drive, and have a question about the performance test program included on the *TEST/DEMO* disk that comes with the drive.

When you run the program it instructs you to insert a scratch disk. What is a scratch disk? Is it the *TEST/DEMO* disk, and can I damage the *TEST/DEMO* disk if the program continues with it in the drive?

Harry Metz

Yes, the programs on the disk can be damaged. If the write-protect notch is not covered and you continue running the program with the TEST/DEMO *disk in the drive, all of the programs could be wiped out.*

In computer terminology, scratched *is defined as meaning* empty or available for use. *A scratch disk is one that has no useful programs on it, and can be used to SAVE files. A scratch tape is the same. It's like a "scratch pad."*

When the performance test prompts you for a scratch disk, take out the TEST/DEMO *disk, and place a blank disk into the drive. Do* not *use a disk which has programs on it that you desire to keep.*

Changing The Atari 800 Cursor

I am working on a program for my Atari 800. I would like to know how to change the cursor from the ordinary box shape to a line. I saw this done in an adventure game and I thought it would be something nice to use in my programs.

John Runions

You will need to write your own input routine which accepts keys from the keyboard, puts them into a string, and permits correction. Then, the cursor you use is up to you. Disable the system cursor with POKE 752,1.

You can display a graphics character on the line below the character it is highlighting. You could use a redefined character set, with patterns for letters with the cursor included (you would then alternate between the normal character and the underlined character). You could also use a player or a missile as a cursor (as done in "Scriptor," April 1983). There are so many ways to display graphics on the Atari that these suggestions barely scratch the surface of the possibilities.

Memory Management In IBM, Atari, And Macintosh

Methods of RAM management are among the most important aspects of computer operation, but I can find practically no information on this. How does one computer compare with another in memory management? Why is it that even in the IBM 16-bit computer, BASIC RAM is limited to 60864 bytes with 256K memory installed? How fast would IBM load a program? I am told that IBM drives use 512-byte sectors, 9 per track. Does this mean a fourfold increase in speed (from Atari's 128 bytes per sector)? What about the new Apple Macintosh? More than anything else, the speed with which swaps in memory are made determines the horsepower of a computer.

Orville E. Bean

The factors which determine how effectively a computer can manage memory are the amount of memory the central processing (CPU) chip can address directly, and the number of bits the CPU can transfer to or from memory at one time. To use your phrase, a 16-bit machine has more horsepower, since it can grab 16 bits at a time from memory, instead of 8 bits. Most home computer CPUs, like the 6502 chip used in the Apple, Atari, and Commodore, can directly address 64K bytes and transfer 8 bits at a time. The 8088 CPU in the IBM PC and PCjr can directly address 1024K bytes (or one megabyte). However, it organizes this memory into 16 blocks (called segments) of 64K each. IBM's Microsoft BASIC was written to operate within one segment, which is why only 60K is available for programming after the computer takes away what it needs for its own operations. (There is at least one third-party BASIC for the PC that supports all available memory.) Although the PC and PCjr are usually called 16-bit computers, their CPUs can actually transfer only 8 bits at a time. The CPU in the new Macintosh can directly address 16

9 to 5 it's business... after 5 it's...

Mail Controller™

No matter what time of day it is, MAIL CONTROLLER is always there to keep me organized. Whether for my business life or social life, it's my choice for list information management.

MAIL CONTROLLER is a mini-database that can store over 2,000 individual records per disk, each with up to 7 different fields. And, one of the most useful features of MAIL CONTROLLER is the self-design field capability allowing me to specify the type of data to be stored. So not only can I keep a listing of mailing addresses, but of phone numbers and anything else that needs to be filed in a concise and comprehensive manner.

MAIL CONTROLLER also enables the exact information I need to be called up in seconds by specifying any or all fields. The "wild-card" search technique adds even greater assistance. And, of course, MAIL CONTROLLER is printer compatible with instant printouts of both lists and mailing labels.

This list handler provides many advanced capabilities, yet, allows incredible ease of operation. With its step-by-step documentation and instructional screen prompts, MAIL CONTROLLER is a cinch to learn and use ... any time of day! Commodore 64 (Disk).

ORBYTE SOFTWARE

P.O. Box 948, Waterbury, CT 06720 (203) 621-9361

megabytes (16384K) and can transfer data to or from memory 16 bits at a time.

As to disk access, the most important limitation to speed is in the disk interface. Even if you had a hard disk—and they're extremely fast—you would still be limited by how quickly your computer and disk drive could communicate. Many home computers use serial interfaces, both for economy and to comply with FCC regulations. (A serial interface transfers one bit at a time, using few wires, whereas a parallel interface transfers a whole byte (8 bits) at a time, over many wires. Too many wires amplify the tendency of a cable to act as a transmitting antenna of radio-frequency interference.)

The IBM computer uses a parallel, high-speed, direct-memory access (DMA) interface. The computer hardware can directly interact with the disk controller. Other computers have to treat the disk drive as a remote peripheral, communicating and buffering data. Again, the interface limits the speed, but disk formats vary in efficiency. An IBM drive can read one 512-byte sector without moving the head, whereas you have to locate four 128-byte sectors to read as much data on an Atari disk drive. Since the Macintosh uses a small 3¼-inch disk, with data tightly packed (400K), less movement of the head is needed to find information, so these new drives are usually faster.

Automatic SYS For Commodore ML Programs

I really enjoy your programs that are written in machine language. I am accumulating quite a collection of ML programs. The problem I'm having is trying to remember the SYS (beginning) addresses to start them. Is there a way to include a line like 10 SYS 49152 in the program so that all one would have to do is type and enter RUN to start the program?

Kris Wechter

Yes, it is possible. As a matter of fact, many commercial games do just that.

The BASIC program 10 SYS 2064 *uses 14 bytes. You can start writing your machine language program past the end of this short BASIC program (memory location 2064 = $0810, on the Commodore 64, for example).*

After it's completed, SAVE it (with a machine language monitor) from address $0801 to the end of your ML program. This technique enables you to LOAD it like a BASIC program and enter RUN to start it.

When you use this method, you can LOAD the program with either the LOAD "filename",number *or LOAD* "filename",number,*1 format (*number = *device number, 1 for tape or 8 for disk).*

Another nice trick is to SAVE the programs with the SYS addresses in the filename. For example, if you have a game called Saucers *that starts at address 49152, SAVE it to tape or disk with a filename of* Saucers 49152. *That way you'll never forget.*

If you presently have programs on tape or disk, and you can't remember their starting addresses, RUN one of these BASIC programs. It will tell you what the starting address is.

Starting Address For Disk Programs

```
10 INPUT "{CLR}ENTER PROGRAM NAME";PN$
20 OPEN8,8,8,PN$+",P,R"
30 GET#8,A$,B$
40 PRINT"START ADDRESS OF {RVS}";PN$;"
   {OFF} IS:"ASC(A$+CHR$(0))+256*ASC(B$+C
   HR$(0))
50 CLOSE8:END
```

Starting Address For Tape Programs

```
10 INPUT"{CLR}ENTER PROGRAM NAME";PN$
20 OPEN 1,1,0,PN$
30 PRINT"START ADDRESS OF {RVS}";PN$;"
   {OFF} IS:"PEEK(829)+256*PEEK(830)
40 CLOSE1:END
```

Relocating Commodore Programs

If I type in a VIC-20 program from your magazine on my 64 and save it on tape or disk, will the program run on my friend's VIC-20?

Ann Harrison

Yes, it will. Both the VIC-20 and the 64 have the ability to automatically relocate BASIC programs. There is only one restriction. You must LOAD them using this syntax:

```
LOAD "filename",number
```

where number *is the device number (1 for tape, 8 for disk). It will not work if you load with the syntax:*

```
LOAD "filename",number,1
```

The extra ,1 after the device number tells the computer to load the program back into the exact area of memory from where it was originally SAVEd. This may cause your programs to run abnormally because the beginning of BASIC memory is different for the VIC and 64.

Atari 800 Keyboard Failure

I own an Atari 800 computer with 48K. Five of my console keys (these are 6, 7, T, Y, and N) no longer call a character to the screen when they are pressed. I have tried turning off the computer and then turning it back on, but this does not solve my problem. The 90-day warranty by Atari has expired, so I have to solve the problem myself.

Is this problem simple enough to be solved at home? If not, then where should I take my machine and approximately how much would it cost to get it fixed?

Luis A. Betances

Books that teach you to program
While turning work into play.
Books of music, math, mysteries,
Mazes...and more.
Books that let your originality shine,
And explore by computer
Your creative mind.

Creative Pastimes.™
Books as unique as you are.

Look for 9 new Creative Pastimes Books in your favorite bookstore, computer store or supermarket. Available now for only $6.95. Compatible with the most popular home computers. Suitable for all ages. For more information, call us toll free: (800) 336-0338. Or write: Reston Computer Group, A Prentice-Hall Company, 11480 Sunset Hills Road, Reston, Virginia 22090.

Of course, we cannot diagnose your problem sight unseen, but we do have a few suggestions. First, you can easily pry off the keycaps with a paper clip. (Be careful: The spring may jump out.) Check to see that the contacts are clean, with no particles preventing closure of the two contacts.

If that doesn't help, and if you don't mind opening your computer, the keyboard is easily accessible. (Be sure the computer isn't plugged into the wall socket.) Check the keyboard cable to see that it is not twisted, frayed, or loose. If this isn't the cure, you'll have to return your computer to an authorized Atari Service Center. Call Atari Customer Service for the name of the dealer nearest you, toll-free: (800) 538-8543.

Expanding Atari 800 To 64K

I have an Atari 800 which I recently expanded to 48K by adding an Intec 32K board. This addition leaves an open slot in my memory compartment. Can I now just add a 16K board in this open slot to bring my machine up to 64K? Or do I need some additional hardware or software to make this upgrade possible?

Neil G. Wyatt II

It's not that easy. The 6502 in your computer can access 64K at a time, but that includes both ROM and RAM. Your machine needs 16K of space for ROM and other system memory, leaving you with a maximum of 48K. Adding another 16K would only hopelessly confuse your machine. There are companies which manufacture 64K RAM boards. Some of these let you rotate segments of your memory with a large bank of extra memory. Others let you temporarily make the ROMs disappear, revealing underlying RAM. This is the technique used in the new Atari XL computers, and on the Commodore 64. At least one company lets you plug your existing RAM chips into a bare board to save you money in the upgrade.

No Easy Conversion From VIC To 64

Being the previous owner of a VIC-20, I have several VIC programs that I would like to use with my new 64. Is there a conversion factor for memory locations available?

Bill Powell

Translating programs from the VIC to the 64 is not always simple. There are many significant differences between the two computers. For example, the SID (Sound Interface Device) chip in the 64 is much more complex than the VIC chip (Video Interface Chip) in the VIC, so sound routines have to be rewritten.

Color and screen memory in the 64 are both 1024 bytes long, while in the VIC they are 512 bytes. Also, the screen and color memory locations in the 64 and VIC are different. This means that all POKEs and PEEKs to screen and color memory have to be changed. There is also the problem of "shifted" ROM. BASIC is stored in permanent memory beginning at address $A000 in the 64, while BASIC starts at address $C000 in the VIC. For example, the print fixed-point value *routine in BASIC ROM is at $BDCD in the 64, and $DDCD in the VIC.*

The way to convert your programs is to go through them one line at a time, and rewrite where necessary.

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. ©

The END *of* DINKETY-DINK.

Announcing the first computer music program that actually sounds like music.

LET'S FACE IT. Up till now, music programs for your home computer have all sounded, well, pretty lame. There were the ones that resembled little electronic music boxes, remember? And then there were those that sounded like so many burps.

Enter Music Construction Set.™ It's the first music program that really makes use of the power of that machine you've got. If you're a serious student, this means you'll be able to work with an intricacy and range of sound quality you've never heard before on a computer. And if you know nothing about music, you'll find something even more important. Namely, that this thing is simple enough to be a lot of fun.

Take a good look at this screen because it, you, and a joystick are the whole story here.

That's you at the right end of the staff of notes – the little hand. Move the joystick, and you move the hand. Use it to carry notes up to the staff. Lay in rests, signatures, clefs, then point to the little piano in the lower right and listen, because you'll hear the whole thing played back.

Move those little scales in the middle up and down to vary the music's speed, sound quality, and volume. Use the scissors to cut out whole measures, then use the glue pot to paste them in somewhere else. Got a printer? Great. Print the score out and show it off to your friends.

But what if you're not up to writing your own stuff yet? No problem. There are twelve pieces of music already in here, from rock 'n roll to baroque. They're fun to listen to, and even more fun to change. (Apologies to Mozart.)

The point is, the possibilities are endless. But if you're still skeptical, visit your nearest Electronic Arts dealer and do the one thing guaranteed to send you home with a Music Construction Set in tow.

Boot one up. Point to the piano. And listen.

NOW AVAILABLE FOR ATARI HOME COMPUTERS AND THE COMMODORE-64

ELECTRONIC ARTS™

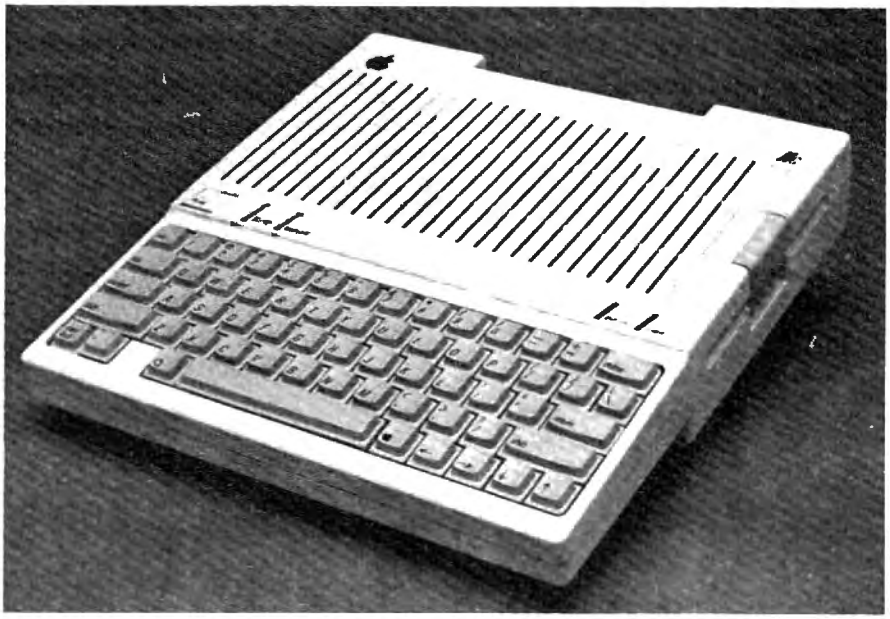
MUSIC CONSTRUCTION SET is now available for Apple II, II+, IIe, and Commodore 64 computers. The Apple version, with a Mockingboard,™ plays chords of up to six notes each. The Commodore version plays chords of up to three notes each. Apple is a registered trademark of Apple Computer. Commodore is a registered trademark of Commodore Business Machines, Inc. For more information about Electronic Arts, write us at 2755 Campus Drive, San Mateo, CA 94403 or call (415) 571-7171.

Evolutionary To The Core:

The Apple IIc Heads For Home

Selby Bateman, Features Editor

Apple Computer has made 1984 its year of surprises, first with the "revolutionary" Macintosh and now with the "evolutionary" Apple IIc. The new 7½-pound portable has already achieved critical acclaim and impressive early sales. Is it the computer for you?



The $1295 Apple IIc, with a half-height 5¼-inch built-in disk drive on the right side, 128K RAM, and a 63-key keyboard, is a smaller, enhanced Apple IIe. (All photos courtesy of Apple Computer, Inc.)

"The IIc is not a home computer," says Apple President John Sculley. "It's for the serious user in the home."

Sculley isn't just playing word games with that comment. It is as succinct a statement of Apple's plans for the IIc as you'll find. And it addresses the biggest challenge and the greatest opportunity for the Cupertino, California, company: To convince a huge untapped home market that the IIc is not a low-end computer. And at the same time, Apple is targeting owners of low-end microcomputers who want more power, more software, and more portability.

$15 Million In Advertising

By now you may have heard or read something about the IIc's power (128K RAM), price ($1295), portability (notebook size, 7½ pounds, built-in low-profile 5¼-inch disk drive), and compatibility (it runs thousands of Apple II software programs). And you've probably seen some of the $15 million in advertising that Apple has spent these past few months.

Still, for most people, the important question is: What is Apple offering you in the IIc that you can't get with the IBM, Commodore, Atari, Radio Shack, and other Apple computers?

If market researchers are correct, the number of personal computers in the home will go up from eight million units now to about 50 million by 1988. This would put at least one computer in two out of every three U.S. households in the next four years.

Two Steves In A Garage

To see how Apple plans to exploit that potential market with its new IIc, you have to go back to 1977. Two young men, Steve Jobs and Steve Wozniak, emerged from their garage workshop with the Apple II, the first fully assembled personal computer. It created a sensation, and the two Steves haven't looked back since. The Apple II begat the Apple II+, which begat the Apple III—a business machine. Then, in 1983, the company introduced the Apple IIe (the *e* stands for *enhanced*), a 64K RAM personal computer which continued the evolution of the Apple II line. Almost two million computers in the Apple II family have been sold.

At that point the problems started. Or as John Sculley puts it: "The Apple IIe was a very important technical improvement on the Apple II+, but the real difference in 1983 was that

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.

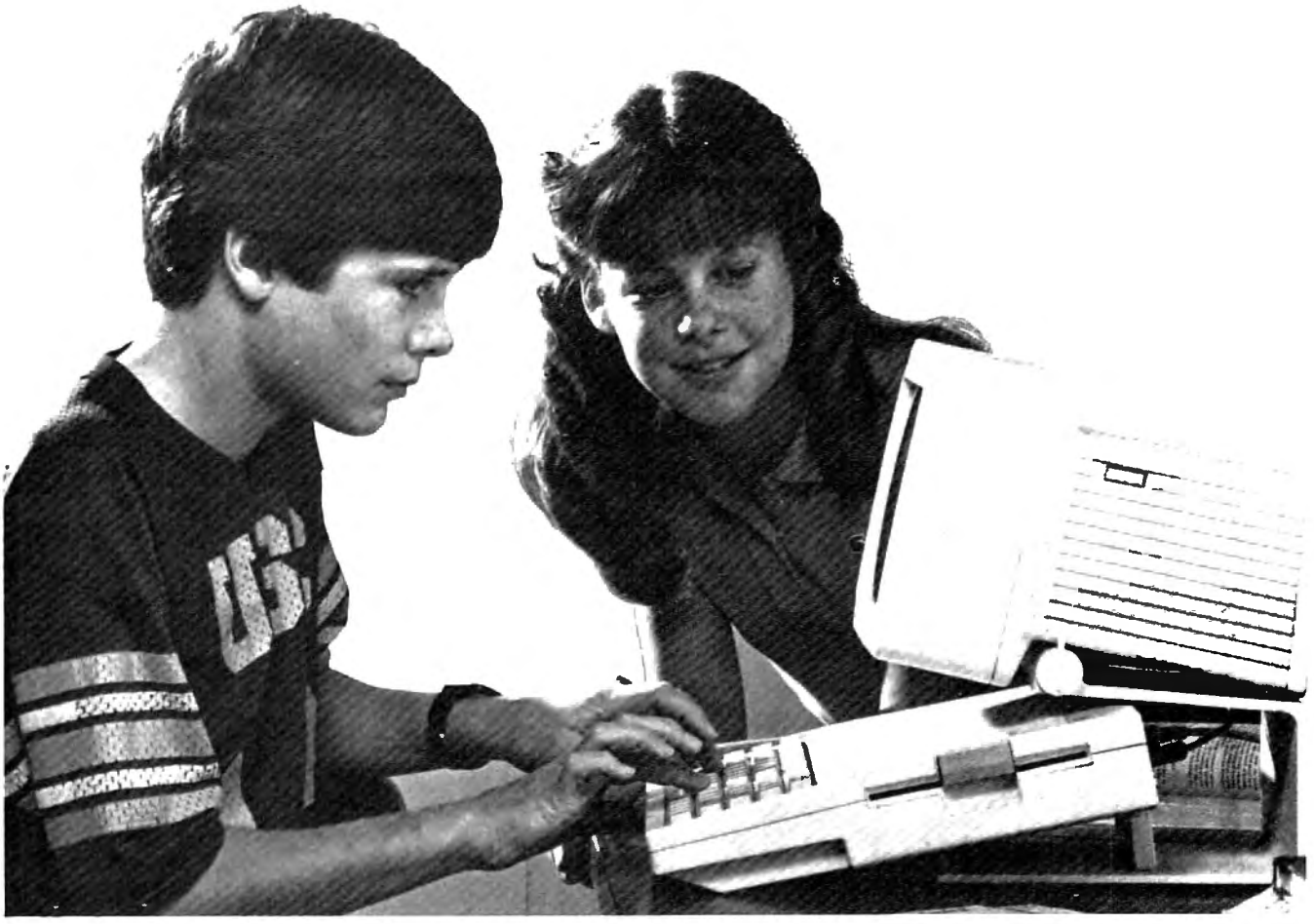
Behind the lower door are a top shelf for paper, feeding the printer, and a bottom shelf to receive printer copy as well as additional storage.

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



The IIc is about the size of a notebook, and can be configured with optional 9-inch monochromatic monitor (as shown here) and with a variety of compatible peripherals.

for the first time, the Apple II faced real competition."

Apple found itself sandwiched between the low price of the Commodore 64 (and several other low-end computers) and the IBM PC, which quickly established itself as a high-end standard.

"While IIe sales continued to grow," Sculley said, "Apple was quickly becoming positioned as a single-product company with declining importance in business and in the home. That's a very dangerous situation."

The First 100 Days Of The Macintosh

But what a difference a year makes. With the introduction of the Macintosh in January of this year, Apple offered a revolutionary personal computer aimed primarily at the business and college markets. It was designed to be the least intimidating and easiest to use computer on the market. The $2495 128K computer, which has no built-in programming language and features a mouse input device, sold more than 70,000 units in its first 100 days.

By contrast, the original Apple II took 2½ years to sell 50,000 units, and the IBM PC took over 7½ months to sell the same number.

Then, this April, Apple introduced the IIc at a day-long exposition in San Francisco. The event featured a high-tech sound and light show, demonstrations of the new product, presentations of compatible software from a variety of companies, and an unplanned earthquake that rocked the city. Within several hours that day, Apple took more than 50,000 IIc orders from more than two thousand retail dealers. Apple had come some distance from two young men in a garage.

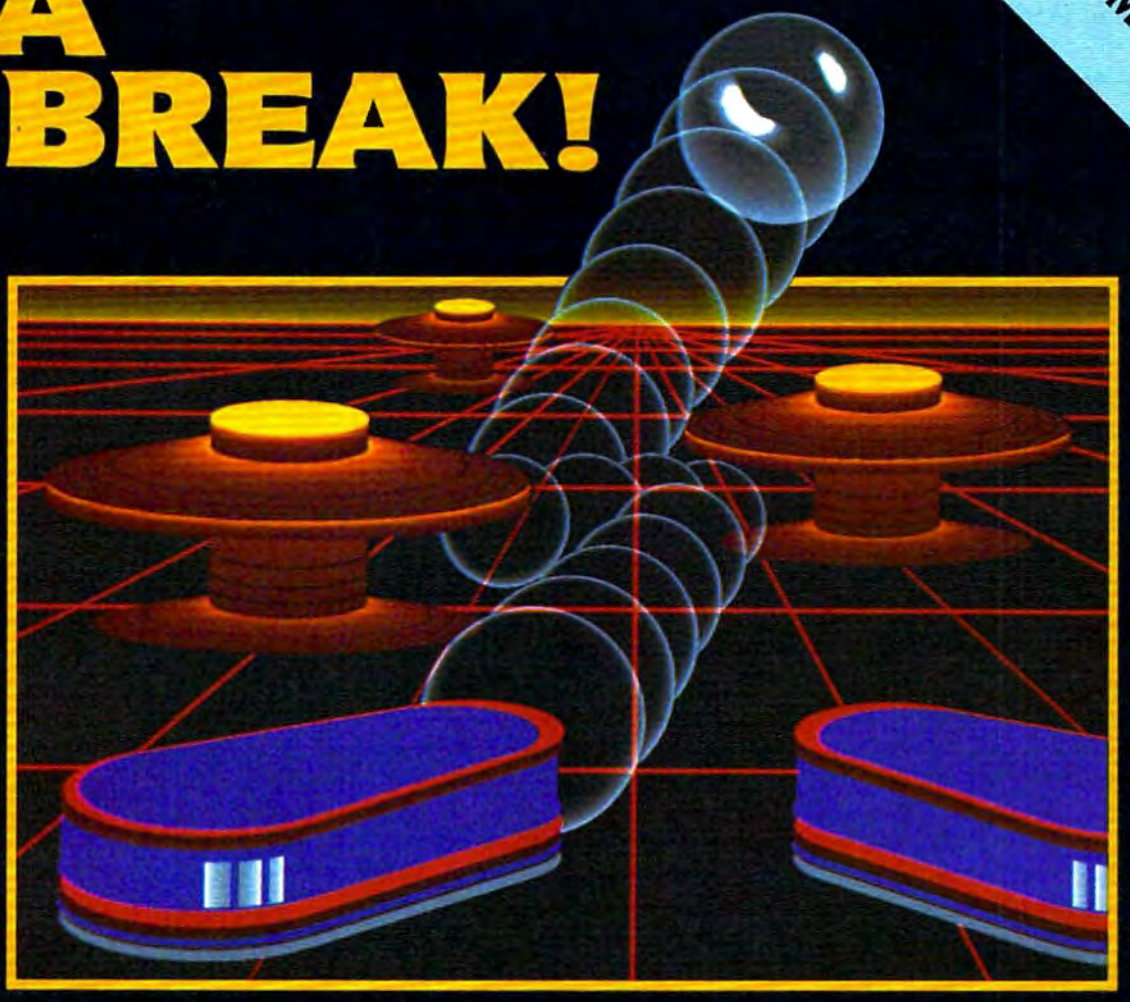
Just Another Appliance

Amidst the carnival hoopla of the IIc's debut, Apple officials restated their argument that the new machine is an evolutionary twist in the Apple II line and that it addresses a new segment of the buying public with a different concept of what a computer should be.

"We should not be judging how convenient the Apple IIc is versus other competitors," says Sculley, "but relative to other consumer appliances that are used by people who have never used any personal computers before. The IIc was designed from the start to demystify the intimidation that personal computers present to so many people." In other words, as with the Macintosh, Apple is trying to make the IIc as easy to use as a toaster, a television, or a stereo system. And the company is trying to do that in the same way that items such as cars,

TAKE A BREAK!

For Apple, Atari, Commodore 64, and IBM-PC



WITH NIGHT MISSION PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of Night Mission PINBALL, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put Night Mission PINBALL in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with Night Mission PINBALL from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.

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. Illinois residents add 5% sales tax. 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

cameras, and pocket calculators have evolved into mass market products.

Apple estimates that about two-thirds of IIc sales will be to the home market, with the rest going to schools and small businesses.

The Snow White Look

To underscore the novelty of the IIc, Apple has altered the physical appearance of the new machine and all of the optional peripherals that are made for it. The new look is called "Snow White," and it features an ivory color, rounded corners, and a sleek, narrow case with a fine-grained texture. Frog Design, a West German firm which also designed the Sony Walkman, created the new casing for Apple. It marks the biggest exterior design change in the Apple II line, and will be the look of all future Apple products.

The computer, including the keyboard, internal circuit board, built-in disk drive, and rear panel peripheral ports, weighs only about 7½ pounds. A recessed handle mounted on the top rear of the computer swings out for carrying and also locks in place in order to position the machine at a correct typing angle. The body of the computer is only 2½ inches high, 12 inches wide, and 11½ inches long—little more than the size of a notebook.

To emphasize the IIc's simplicity, everything a user needs to get started comes in one box, except a display screen. The absence of a monitor with the new computer will allow consumers to choose between television, composite monitor, and RGB (red-green-blue) displays. And, by September, a new flat liquid crystal display will be available, says Steve Jobs, Apple's chairman of the board. It also keeps the initial price lower and allows dealers to sell the computer in a variety of configurations.

By September, Apple promises this flat LCD screen, which fits on top of the main IIc unit and has full 80-character by 24-line display.

There aren't any ROM cartridge slots either. Jobs calls those cartridges clumsy, and prefers the cheaper, more effective, and more flexible floppy disks.

70 Fewer Chips

Internally, the IIc has a 65C02 central processing unit (CPU)—an eight-bit microprocessor—which Apple says is an enhanced version of the 6502B CPU used by the Apple IIe. Both of those chips are based on the original 6502 manufactured by MOS Technology. The C in 65C02 is taken from CMOS (Complimentary Metal-Oxide Semiconductor), which designates the production process of the chip. The new version reportedly has cooler operation and requires less power, allowing the IIc to use a battery pack that is to be available later.

The 65C02 chip also reportedly includes 27 new programming instructions. If programmers use these instructions in their IIc software packages, those programs will not work on earlier Apple II models with the original 6502 or 6502B chip. Upward compatibility from the earlier Apple II machines to the IIc is not a problem, however, Apple says.

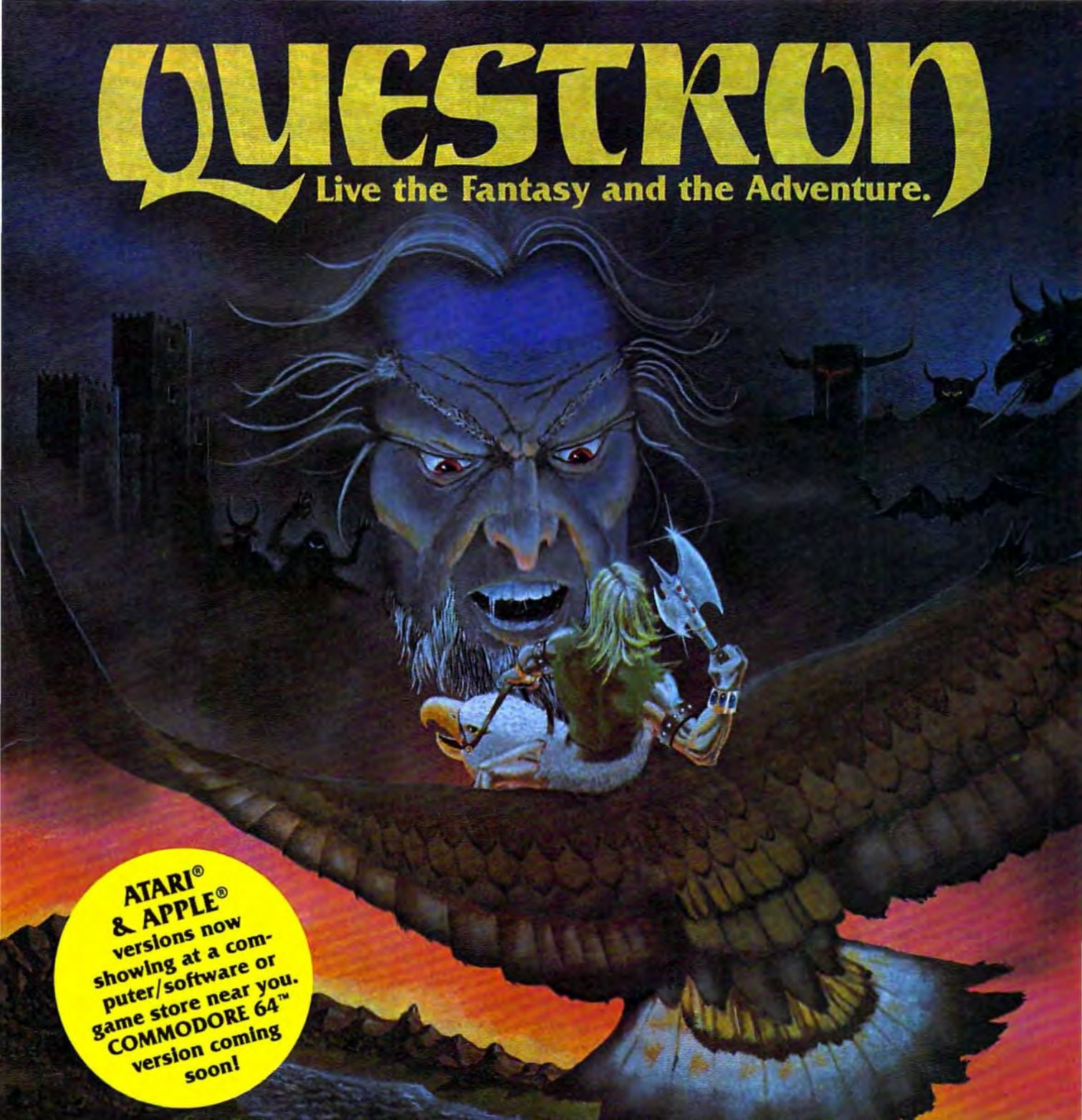
There are only 40 chips within the IIc, quite a reduction from the 110 which are found in the IIe. Apple engineers combined the functions of some chips onto custom large-scale integrated chips to achieve this. They also combined functions: The mouse peripheral port on the rear of the unit, for example, can accept a mouse, joystick, or hand controllers. Although there are fewer chips, the machine has double the usable memory of the Apple IIe—128K, or the equivalent of about 50 double-spaced typed pages of work area. There is 16K of ROM (Read Only Memory), containing among other things the standard Applesoft BASIC programming language.

Ultrahigh Resolution

There are three graphics modes available with the IIc. First, a low-resolution graphics mode of 40 x 48 pixels with 16 available colors and, second, a high-resolution mode of 280 x 192 with 6 colors—both of which are comparable to the IIe's

QUESTRON

Live the Fantasy and the Adventure.



ATARI® & APPLE® versions now showing at a computer/software or game store near you. COMMODORE 64™ version coming soon!

STRATEGIC SIMULATIONS INC. PRESENTS A FANTASY ADVENTURE GAME: QUESTRON™
ONE OF THE FINEST CHAPTERS IN THE NEVER-ENDING SAGA OF THE BATTLE BETWEEN GOOD AND EVIL
Starring YOU as THE HERO • MESRON, THE GOOD WIZARD • MANTOR, THE EVIL SORCEROR
AND HIS HORDES OF HERO-CRUNCHING MONSTERS • Written and directed by CHARLES DOUGHERTY

On 48K disk for your Apple® II with Applesoft ROM card, Apple II+, IIe, or Apple III. Also for Atari® home computers.

SSI

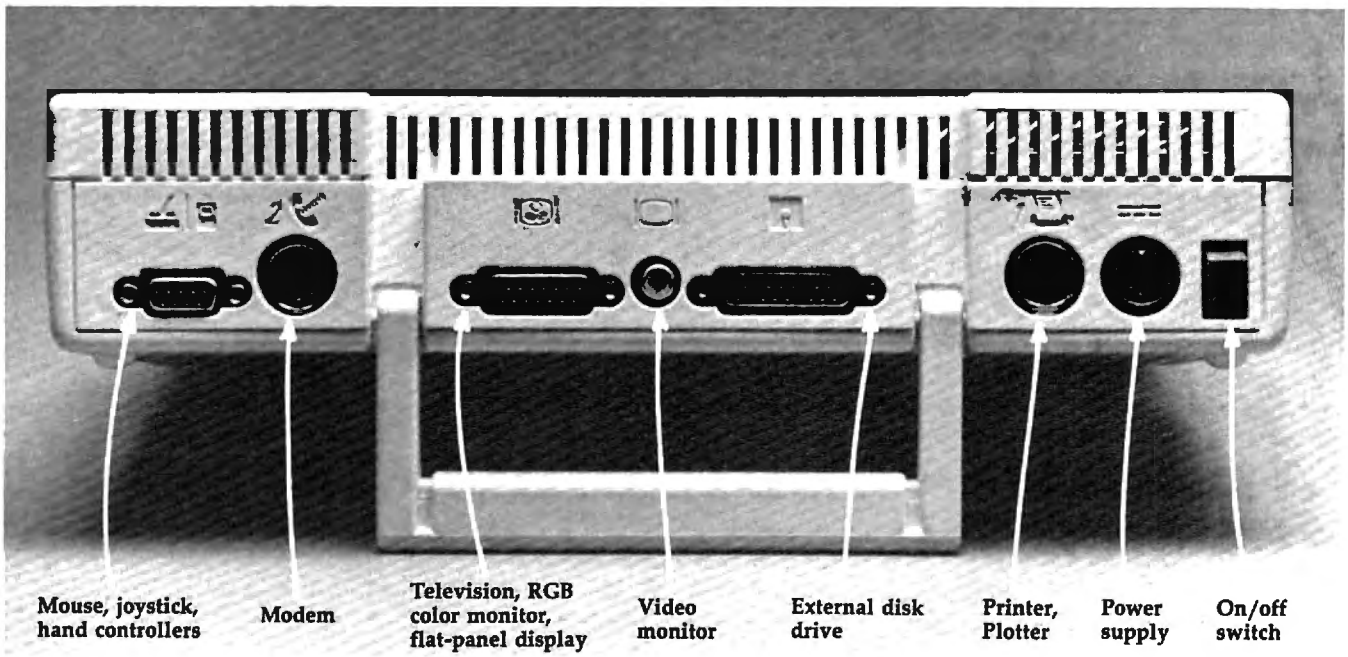
PG | THIS GAME RATED POSITIVELY GREAT. Ideal for Fantasy Adventurers of all ages.

APPLE & ATARI are trademarks of Apple Computer, Inc. and Atari, Inc., respectively. **STRATEGIC SIMULATIONS INC** COMMODORE 64 is a trademark of Commodore Electronics, Ltd.

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. **QUESTRON™** goes for **$49.95**, plus $2.00 for shipping and handling.

To order by mail, send your check to: STRATEGIC SIMULATIONS INC, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. (California residents, please 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.



The peripheral ports on the rear of the main panel are virtually foolproof for installation, and feature icons above each socket to show the user the various uses.

graphics modes. The IIc also has an ultrahigh-resolution monochromatic mode of 560 x 192, which approaches but does not match the Macintosh's 512 x 342 pixel monochrome display. Apple says that in the future the ultrahigh-resolution mode will be able to support 16-color graphics.

The IIc also has an internal speaker almost identical to that in the IIe. It allows five octaves of sound. The IIc has a volume control button on the side of the case, which the IIe does not. And there is a plug for headphones in the new machine, next to the volume control button.

The keyboard on the IIc represents a departure from the IIe, although the basic layout and size are virtually identical. There are 63 sculpted keys, but rather than the smooth up and down action of the IIe, the IIc's keys have a breakover effect—a definite toggle—that gives a tactile sense when a key has been hit as well as an audible click. The effect is not unlike the keyboard action on the IBM PC, although the breakover action on the IIc is not as pronounced.

Spillproof

Above the keyboard, starting on the left, are three partially recessed switches: a rectangular reset button, which on the IIe is to be found on the right side of the keyboard; a button that changes the video display from 40 to 80 columns (or vice versa) for text; and another button which will alter the keyboard from the standard QWERTY key configuration found on most typewriters to a DVORAK keyboard. The DVORAK keyboard, which has a faster and more logical key layout than QWERTY, is growing in popularity. Apple will also have replacement key caps for those who wish to install them in the DVORAK layout.

An Apple spokesperson says that six different keyboard prototypes were tested for the IIc, and that on the recommendations of touch typists who tested the various keyboards, the present style was adopted.

Another interesting keyboard feature is a layer of plastic beneath the keys, which will prevent any spilled liquids from penetrating into the computer itself. At typing angle, liquids will drain toward air vents at the front of the machine. An Apple spokesperson called this the "drool" cover.

Foolproof Peripheral Ports

Built into the right side of the main unit is a low-profile 5¼-inch disk drive, which has a 140K capacity. A total of 137K of that is available with ProDOS (a Disk Operating System) and with Pascal DOS. 124K of RAM is available with DOS 3.3.

The system uses a 12-volt detachable power supply, which will allow an unmodified IIc to run off a car's cigarette lighter, a battery pack, or international electrical outputs. This power supply, in the same Snow White style, is packaged in the IIc box along with the main unit, cables, RF modulator for TV hookup, and a disk-based owner's manual/computer literacy course aimed at the first-time user.

But one of the IIc's most attractive features for consumers new to computers could well be its back panel of peripheral ports. This also represents a major departure from the IIe. Instead of the add-on card slots

PICKING DAISIES?

BROTHER HR15

Costs about the same—but it's slower, noisier, and needs its own brand of ribbon. To be fair, it's lighter.* (But JUKI eclipses the BROTHER totally!)

SILVER REED EXP500

Okay, it's lighter—but it's more than a whole word slower per second, it's noisier, lacks a buffer memory, and prints only a 10"-wide line.* (JUKI triumphs again!)

SILVER REED EXP550

You pay about $100 more, and it's slower, noisier, has no buffer memory, and lacks the refinement of our linear stepper carriage motor. A little wider print line, yes. A bargain, no.* (JUKI by a mile.)

QUME LP20

Costs about $300 more, needs its own brand of ribbon, and takes only a 96-character wheel. Is it worth it for just 2 more characters per second and a wee bit quieter machine?* (Sorry, QUME. JUKI gets the trophy.)

DIABLO 620

Costs about twice as much, weighs 19 lbs. more, and requires its own brand of ribbon. Pretty steep for a slightly quieter machine and 2 more characters per second.* (The winner: JUKI.)

JUKI 6100

CONSIDER THESE FEATURES: Compatible with most personal computers (IBM, Apple, Kaypro, etc.), prints graphics, 2K buffer (expandable to 8K), bidirectional tractor feed option, proportional spacing, lightweight, 11" print line, uses 100-character drop-in daisywheel and inexpensive, easy-to-find IBM Selectric II® ribbon! Interchangeable interface and easy-to-read manual. **Feature for feature, dollar for dollar, JUKI—the best all-round letter-quality printer anywhere!**

THERE ARE LOTS OF DAISYWHEEL PRINTERS IN THE FIELD. PICK SMART. PICK JUKI 6100.

*Comparison based upon manufacturer's specifications rather than actual testing.

JUKI®

JUKI INDUSTRIES OF AMERICA, INC.

ACORN DATA PRODUCTS
7042 S. Revere Pkwy, Ste 50
Englewood, CO 80112
(303) 799-8900
Serving
MT, WY, CO, UT, NM

BUTLER ASSOCIATES, INC.
82A Winchester St.
Newton, MA 02161
(617) 964-5270
Serving
ME, NH, VT, MA, CT, RI

CM DISTRIBUTION
7023 Little River Trpk
Annandale, VA 22003
(703) 750-3885
Serving
MD, DE, DC, VA

COMPUTER SERVICES INTL
560 Sylvan Ave
Englewood Cliffs, NJ 07632
(201) 569-6300
Serving
METRO NY E. PA, NJ

CYPRESS DISTRIBUTING CO.
1266 Lincoln Ave., Ste 109
San Jose, CA 95125
(408) 297-9600
Serving
N. CA, NV, AZ

GENTRY ASSOCIATES INC.
7665 Currency Dr.
Orlando, FL 32809
(305) 859-7480
Serving
TN, NC, SC, MS, LA, AL, FL, GA

INFORMATION SYSTEMS INC.
2420 E. Oakton St., Unit K
Arlington Heights, IL 60005
(312) 228-5460
Serving
WI, IL, MN, IA, MO, NE, ND, SD, KS

MICRO SOURCE OF TEXAS INC.
670 International Pkwy
Richardson, TX 75081
(214) 690-5111
Serving
TX, OK, AR, LA

OSSMANN COMPUTER TECHNOLOGIES
6666 Old Collamer Rd.
East Syracuse, NY 13057
(315) 437-6666
Serving
UPSTATE NY

SOUTHERN MICRO DISTRIBUTORS
8708 Royal Lane
Irving, TX 75063
(214) 258-8636
Serving
TX, OK, AR, LA

STAR-TRONIC DISTRIBUTING CO.
23976 Freeway Park Dr.
Farmington Hills, MI 48024
(313) 477-7586
Serving
MI, IN, OH, KY, W. PA, WV

TECHNOLOGY MARKETING CORP.
2300 Valley View Lane, Ste 109
Dallas, TX 75234
(214) 243-7994
Serving
TX, OK, AR, LA

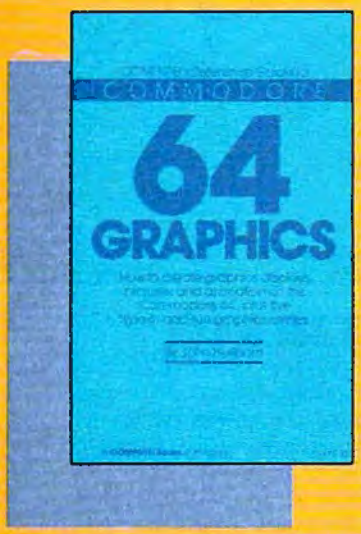
VITEK
930G Boardwalk Ave
San Marcos, CA 92069
(619) 744-8305
Serving
S. CA

WESTERN MICRO TECHNOLOGY
10040 Bubb Road
Cupertino, CA 95104
(408) 725-1660
Serving
N. CA, NV, AZ

NATIONAL HEADQUARTERS
JUKI INDUSTRIES OF AMERICA, INC.
DA DIVISION
299 Market St.
Saddle Brook, NJ 07662
(201) 368-3666

WEST COAST
JUKI INDUSTRIES OF AMERICA, INC.
CALIFORNIA DIVISION
3555 Lomita Blvd
Torrance, CA 90505
(213) 325-3093

COMPUTE! Books

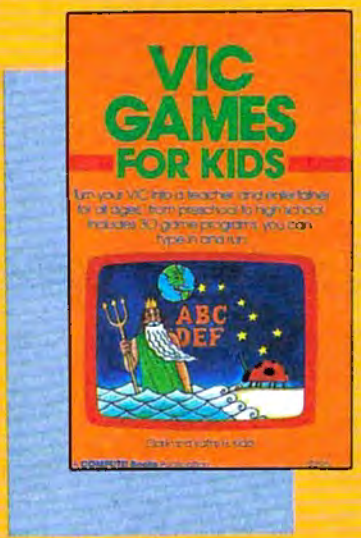


COMPUTE!'s Reference Guide To Commodore 64 Graphics

A complete tutorial on Commodore 64 graphics. Noted Commodore author John Heilborn explains how to program sprites, multicolored screens, animation, custom characters, and more. Beginners will like the step-by-step instructions and clear example programs. Advanced programmers can build up their tool kit with the character editors, sprite editors, screen design program, and other useful utilities.

218 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-29-9

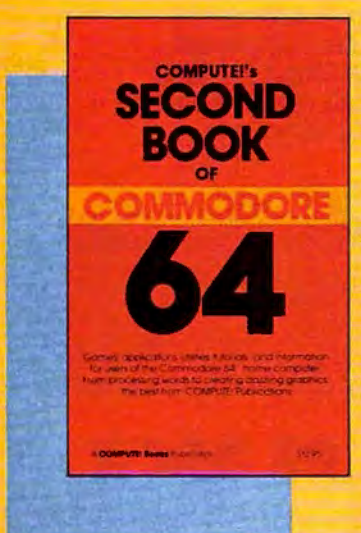


VIC Games For Kids

Contains 30 games written just for kids (although adults will enjoy them too). This book is an inexpensive source of educational software for children. The games are designed to teach math, geography, history, and other topics. Children learn while they're having fun. They will return to these games again and again.

240 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-35-3

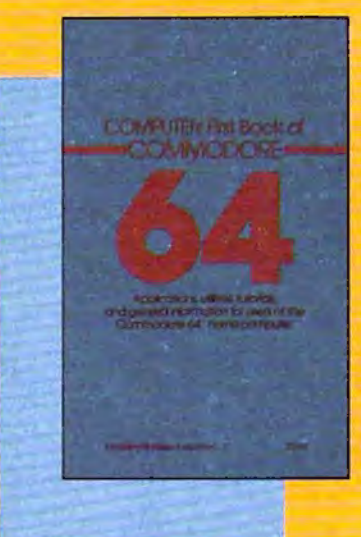


COMPUTE!'s Second Book of Commodore 64

Continues in the tradition of the best-selling *First Book of Commodore 64* in presenting quality programs and articles, many revised or never before published. There's something for almost any 64 user: arcade and text adventure games, an impressive word processor, a program which adds 41 new BASIC commands, an electronic spreadsheet, sound and graphics tutorials, and information on saving, copying, and retrieving files.

288 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-44-2

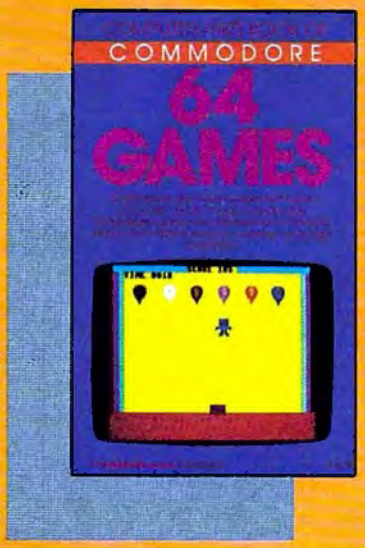


COMPUTE!'s First Book Of Commodore 64

An excellent resource for users of the 64, with something for everyone: BASIC programming techniques, a memory map, a machine language monitor, and information about writing games and using peripherals. Many ready-to-type-in programs and games.

264 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-20-5

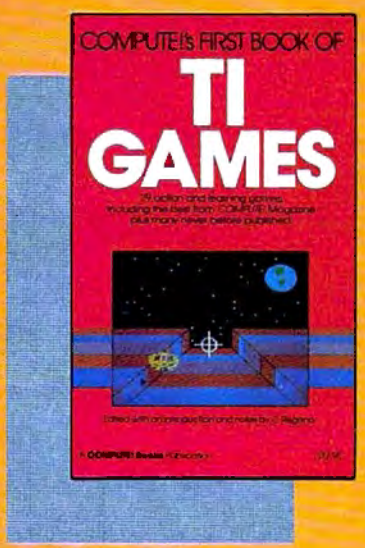


COMPUTE!'s First Book Of Commodore 64 Games

Packed full of games: "Snake Escape," "Oil Tycoon," "Laser Gunner," "Zuider Zee," and many more. Machine language games requiring fast hands and a good eye, as well as strategy games which will exercise your mind. Introductory chapters and annotated listings provide ideas and techniques for writing games. An excellent introduction for 64 owners who want to begin writing games.

217 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-34-5

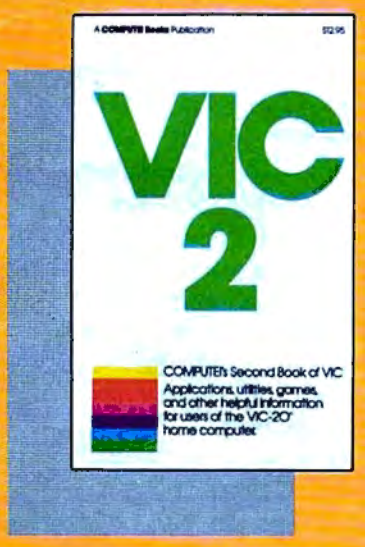


COMPUTE!'s First Book Of TI Games

Although this book is packed with ready-to-type-in games (29 in all), it is more than just a book of games. It is designed to teach game programming techniques. Introductory chapters explain the special features of the TI-99/4 and 99/4A, giving advice on coding techniques. Most games include an explanation of how the program works. Contains mazes, chase games, old favorites, thinking games, creative challenges, and more.

211 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-17-5

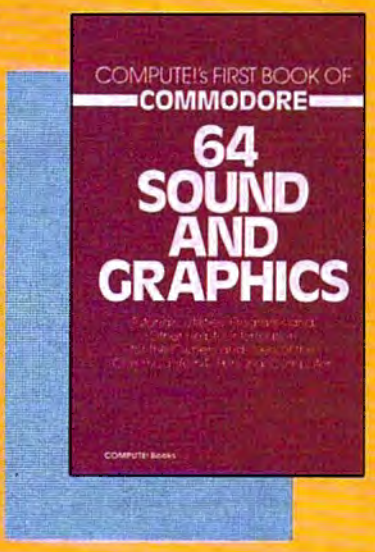


COMPUTE!'s Second Book Of VIC

This is just the book to follow the best-selling *First Book of VIC*: clear explanations of programming techniques, an extensive memory map, a mini word processor, a system for creating sound effects, a custom character maker, a machine language assembler, and "Gumball," an extraordinary all-machine-language game.

274 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-16-7



COMPUTE!'s First Book Of 64 Sound And Graphics

Clear explanations of the 64's sound and graphics capabilities. Includes many tutorials and example programs: "MusicMaster," a complete music synthesizer; "High-Resolution Sketchpad," an all-machine-language program for making computer art; and "Ultrafont Character Editor," one of the best character editors available. The appendices feature useful reference charts and conversion tables.

275 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-21-3

COMPUTE!™ Publications, Inc. abc
One of the ABC Publishing Companies
Post Office Box 5406, Greensboro, North Carolina 27403

The Apple IIc, with optional 9-inch monitor, Scribe thermal transfer printer, AppleMouse II, external disk drive, joystick, modem, and carrying case.

found on the back of the IIe, the new computer has a row of plug-in sockets which can simultaneously support a display screen, printer, modem, mouse pointer, joystick, and a second disk drive. And each socket is configured so that only the correct connector can be plugged into it, thus preventing a beginner from making a major error during setup. Each of the sockets, including two high-speed serial ports, is identified by a picture on the case.

Also, Apple computers have traditionally been hobbyists' and hackers' playgrounds, easily opened and modified. Not so with the IIc. The main unit is sealed, representing Apple's philosophy that this computer is for all of those who don't want to have to learn about expansion cards, complex interfaces, and the like.

Flat LCD Display

The IIc is expandable, with a variety of optional accessories. The Scribe printer ($299), for example, is a 13-pound thermal transfer printer that uses regular paper, prints text and graphics in six colors plus black, and has a low-resolution speed of 80 characters per second and a near-letter-quality speed of 50 characters per second.

Other accessories include an 11-pound Apple Monitor IIc ($199) with a nine-inch green phosphor display (a handle is also included for the monitor); the AppleMouse IIc ($99), similar to the mouse used with the Macintosh and Lisa 2 computers (requires no add-on card); an external 5¼-inch disk drive ($329) with 140K capacity; a IIc monitor stand ($39); and a carrying case ($39) with room for the power pack and the mouse.

By September, Apple promises to have a flat liquid crystal display (LCD) screen for the IIc that will fit onto the top of the machine and make the IIc even more portable. Priced at about $600, the LCD screen will hold as much information as a regular monitor—80 characters wide by 24 lines long. Prototypes of the new screen were being demonstrated at the introduction of the IIc in San Francisco. Although the screen image was inferior to the 9-inch monitors in both luminance and ease of use, there's little doubt that the flat screen will be a popular option when available.

In addition, several Apple II-family peripherals will run on the IIc including the Imagewriter dot-matrix printer ($595), both the 300-baud ($225) and 1200-baud ($495) modems, the Apple joystick ($59.95), hand controllers ($34.95), and color plotter ($779).

Thousands Of Programs For The Home

If Apple considers its hardware improvements the key to breaking down buyer resistance among noncomputer users, it's counting on the thousands of compatible software programs available for the IIc to help bring the computer home.

According to Apple, more than 10,000 programs have been written for the Apple II over the past seven years and more than 90 percent of the programs still available will run on the IIc. That is a wealth of business, educational, home productivity, and entertainment software that not even IBM can come close to offering.

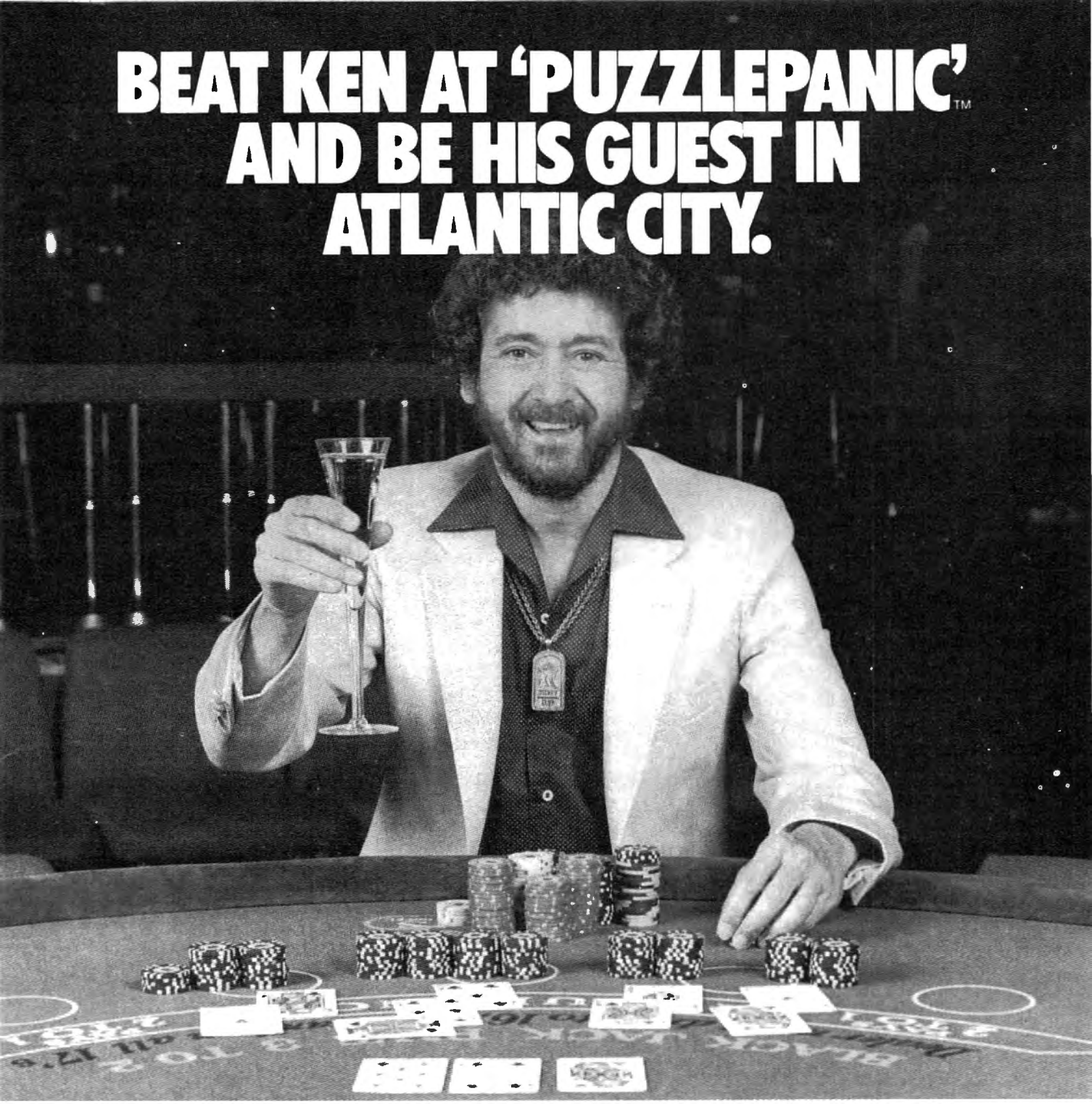
Any existing Apple II software that might not be compatible would result from a software manufacturer's use of a copy protection mechanism or reserved memory locations not recommended by Apple.

To drive home the point of software availability, Apple has been working with more than 100 software companies to make sure there are plenty of new programs designed to take advantage of the IIc's larger memory, ultrahigh resolution, and built-in mouse technology.

AppleWorks For The IIc

Apple also introduced four of its own programs for the IIc: *AppleWorks*, an integrated word processing, spreadsheet analysis, and data base management package; *Apple Access II*, telecommunications software; *Apple Logo II*, a graphics-oriented program-

BEAT KEN AT 'PUZZLEPANIC'™ AND BE HIS GUEST IN ATLANTIC CITY.



Ken Uston doesn't think you can solve his computer puzzle, "PUZZLEPANIC." But if you can, you'll have a chance to be his guest, all expenses paid, for a weekend in Atlantic City. If you're over 21 years old, Ken will show you the ins and outs of blackjack at Resorts International's fabulous casino. If you're under 21 years of age, Ken will take you on in the hotel's tremendous and well-equipped arcade, challenging you to every game in the place. Either way you can't lose. Conquer "PUZZLEPANIC" and take on the world famous Ken Uston in the bargain, but hurry, contest ends August 31, 1984.

Here's what you have to do: Solve each puzzle, figure out which symbol will correctly take you to the next puzzle (over 40 puzzles in all) then, use these symbols to find the solution to the Grand Puzzle. Ken will pick a winner from a drawing of all correct entries.

To enter, send your symbol chart listing the correct symbol for each puzzle, along with your name, address and telephone number to: PUZZLEPANIC CONTEST, c/o EPYX, Inc., 1043 Kiel Court, Sunnyvale, CA 94089.

EPYX™ COMPUTER SOFTWARE

Strategy Games for the Action-Game Player

CONDITIONS: All entries must be postmarked by August 31, 1984. Incomplete entries will not be considered. Winner will be selected from a random drawing of all correct entries to be held on or before September 30, 1984. Winner will be notified by mail within 30 days of drawing. Contest void where prohibited by law. Contest restricted to Continental United States.

ming language; and The Apple Education Classics, two popular educational packages—*Elementary, My Dear Apple* and *The Shell Games.*

The 21 third-party programs featured by Apple at the introduction of the IIc included such packages as *Bank Street Writer* by Brøderbund, Inc.; *Financial Cookbook* by Electronic Arts; *Dollars and Sense* by Monogram/Tronix Publishing, Inc.; *Crypto Cube* by Design-Ware, Inc.; *Fact and Fiction Toolkit* by Scholastic Wizware; *Mastering the SAT* by CBS Software, Inc.; *MasterType* by Scarborough Systems; *Rocky's Boots* by The Learning Company; and *Stickybear Shapes* by Xerox Education Publications, among others.

Since something like 70 percent of the computers in schools today are Apple II's and IIe's, the company believes many parents will opt for the IIc because their children are familiar with those machines.

IIe Or IIc?

Apple dealers are less certain about the impact that the IIc will have on sales of the IIe. Concurrent with the announcement of the new computer, Apple cut the price of a IIe almost in half, down to $995. That, according to Apple's Sculley, means that the IIe can be viewed as a thousand-dollar entry machine with exceptional expandability. Admitting that the two machines have a somewhat overlapping target market, Apple nonetheless expects the IIe to continue to have sales in educational (with even deeper discounts), business, and home markets. "IIe or IIc. That is the question" is the advertising phrase Apple has adopted as it attempts to sell—and point out the differences between—the two computers.

With two very different product lines—the Apple II family and the Lisa/Macintosh family—Apple is most obviously squared off against both the IBM PC and the PCjr. Ads for the IIc criticize the PCjr's limited software and its chiclet-style keyboard. As this is written, IBM has already announced to its stockholders that changes will be made in the PCjr, although no specifics were given. While IBM is not about to step away from any of its machines, Apple's giant archrival is scrambling to meet this surprising double onslaught from Apple.

In price and features, the $1295 IIc is closer to the Expanded Model ($1269) PCjr than the PC. The PCjr, like the IIc, has 128K (a 64K plug-in board); switchable 40/80-column video capability; and a built-in 5¼-inch disk drive. However, the IIc is packaged with an RF modulator for television hookup; you must buy a $30 RF modulator for the PCjr as well as cables. The PCjr operates with DOS 2.1, which costs another $65.

Aging Technology

By choosing to use a "c" in the new computer's name, Apple has been able to offer its marketing team a lot to play with: compact, convenient, complete, and comprehensible. The company's advertising is reflecting all of these concepts in an attempt to market the IIc as the first serious *people's* computer.

Many people were ready to write off Apple in 1983, calling the Apple II line an example of overpriced, aging technology. But with the IIc, Apple is betting that it will shake off that label—confounding the company's critics, battling IBM successfully on two fronts, and living up to the "Apple II Forever" slogan it adopted for the IIc's introduction.

Hopes have never been higher at Apple, nor the stakes more important. But, as John Sculley says, "If we're right, and we think we are, Silicon Valley will never be the same again."©

By The Year 2000, The World May Catch Up With The Way CompuServe's New Electronic Mall™ Lets You Shop Today.

Introducing the first computer shopping service that brings you convenience, savings and enjoyment.

Here's your chance to expand the practical uses of your personal computer.

Sign up for CompuServe and shop in our new Electronic Mall. It's easy to use. It tells you more about the products you're buying. It lets you order faster. And it's totally unique.

CompuServe's new Electronic Mall™ offers you all these shopping innovations.

– It's enormous! So it gives you in-depth information on thousands of goods and services, and lets you buy even hard-to-find merchandise. – Its unique "Feedback" service lets you ask the merchants themselves specific questions. – It's incredibly efficient in ordering the products and services you want. – Its special discount opportunities make it economical, purchase after purchase. – And its name-brand merchants assure you of top-quality merchandise.

Make the CompuServe Electronic Mall 15-Minute Comparison Test.

What you can do in 15 minutes shopping the Electronic Mall way.

- Call up on your computer screen full descriptions of the latest in computer printers, for instance.
- Pick one and enter the order command.
- Check complete descriptions of places to stay on your next vacation.
- Pick several and request travel brochures.
- Access a department store catalog and pick out a wine rack, tools, toys... any thing!
- Place your order.

What you can do in 15 minutes shopping the old way.

- Round up the family and get in the car.

The Electronic Mall, a valuable addition to the vast world of CompuServe.

CompuServe's Consumer Information Service brings you shopping information, entertainment, personal communications and more.

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

To receive your illustrated guide to CompuServe 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

The Electronic Mall™ is a cooperative venture between CompuServe Inc. and L.M. Berry & Company.

An H & R Block Company

How To Choose A Home Data Program

Kathy Yakal
Editorial Assistant

Whether or not you realize it, you've probably created and used data bases hundreds of times: every time you filled out a form for a doctor or employer or bank, or bought a new address book. Data base software for personal computers can make the creation, maintenance, and access of data files—information processing—far more efficient.

Have you ever sat near the reference desk of a public library and watched people do research? There's usually one scholarly looking gentleman with eyeglasses perched atop his head, a chewed-down pencil stub behind one ear, and crumpled yellow reference slips falling from his pockets as he looks in the card catalog.

Evolving Methods Of Research

As the computer begins to offer new ways to sift information, however, the techniques of these scholarly gentlemen may eventually become a thing of the past.

Doubtless, computer terminals with improving search software will continue to proliferate in public libraries, airports, department stores—anywhere information needs to be processed.

Our awareness of these data bases is focused on the data itself, not on the software necessary to store and search for it. Once we've learned how to use a particular data base, we tend to forget about the middleman, the data base software.

But if you're thinking of changing the way you file personal information by setting up a data base on your home computer, you may want to consider what kind of software will best suit your needs.

Searchware?

There is some confusion about what the term *data base* actually refers to. It is often used interchangeably to mean both the software used to store the data and the data itself.

Technically, *data base* means the information itself. A data base manager (a computer program) controls and processes that data. It's the manager that you can buy, and that's what we'll be discussing here. The data base is something you enter yourself—a list of your library books, for example.

Impulse buying is rare among software shoppers. Unless there's a two-for-$5 bin, the consumer is usually very cautious, finding out as much as possible about the software before a purchase.

It's especially important to determine your needs prior to purchasing a data base manager. "Maybe that seems too obvious," says Steve Bellinghausen, "but a lot of people don't do that."

Bellinghausen is distribution manager for Professional Software, Inc., publisher of *DataPlus-PC*, a recently released data base manager for the IBM-PC.

A Consumer's Quandary

How do consumers go about matching their needs with an appropriate data base program? In the past, retailers have helped. "Hardware and software dealers traditionally have performed some kind of consulting role to end users," says Bellinghausen. "As new products and companies flood the market, that's becoming increasingly difficult to do."

Bellinghausen describes a scenario where the unprepared consumer walks into a computer dealership and says he needs a data base manager. The dealer points to a flashy display and tells the customer that it's the

best-selling data base on the market.

So he buys it, only to find out that the program does far more than he'll ever need from it, and he's probably wasted a few hundred dollars. "You don't need a sledgehammer to drive a thumbtack," says Bellinghausen. "Or vice versa. You don't want to go the other way, either."

Though it may be easy to end up buying data base software that does more than the consumer needs or not enough, Bellinghausen thinks that asking yourself and the retailer a few simple questions can prevent that. "The consumer should be sharp enough to figure out what he wants to do with it," he says.

Let's look at some of those questions.

An Electronic Filing Cabinet

How many different files do you anticipate creating? If you're buying a data base manager for one use only, like cataloging a stamp collection, this isn't important. But some data base managers allow you to store only one file on a disk. So if you have several small files, you may be wasting disk space if you buy a program with that restriction.

How large do you expect your files to be? With even the most limited data base manager, you can always create new files if you run out of space. But if you have to do any kind of search, you may not be able to merge your files and run a search on the complete file. Try to estimate your storage needs generously.

What provisions does the data base manager make for defining the individual fields (subsections) within each record? Most programs allow you to design the format for each record, to designate how many fields per record, how many characters per field, whether letters only or numbers only or both will be allowed, and so on. Though each data base manager has its limits, a few packages are extremely limiting.

Will you be needing complicated sorts and searches? Most home applications don't require anything very intricate. But if you want to do more than, say, alphabetize, or retrieve by city or state, you'll need a more powerful data base manager.

How about printing reports? You probably want some kind of printer capabilities, which all data base software has. But the extent to which you can design specialized reports varies. Anticipate your future needs.

Remember: You should be able to have all your questions answered either by examining the outside of the package, reading the software documentation, or asking specific questions of the dealer. If you're planning a substantial investment in a data base manager, it might even be worth writing to the publisher if you can't get an important question answered.

A Few Bonuses

Though it's not absolutely necessary, it can be helpful to have a data base that is compatible with a word processing program. The reason for this, believes Bellinghausen, is that mailing lists are "far and away the most widely used application."

Another feature that few data base programs offer is the ability to go back and change field specifications after you've already entered a number of records. To illustrate the value of this, let's set up an imaginary file, a personal mailing list.

Addressing The Problem

It would seem like the best way to set up a file structure for this application would be to imitate the way that an address book is arranged: one line (field) for name, one for street address, one for city, state, and zip, and one for phone number. Maybe an extra line in case the address runs long.

You then specify that each field can accept both alpha and numeric characters, and allow ample characters per field. You transfer all the information from your address book and various scraps of paper lying around on your desk at home.

Then in November you start thinking about sending Christmas cards. You remember that little notebook that you've used to keep track of cards sent and received over the last five years.

At this point it becomes clear that you should have specified extra fields in your address file for the Chrismas list. Also, it would have been nice to have specified fields to keep track of birthdays.

If your data base software does not allow you to go back and add new fields to existing records, your options are to either set up a new file and reenter all of your records, or keep one set of records in a drawer and one on a disk.

Making It Easier

"I used to use data base software on my Atari," said one home computer owner we questioned. "But now I just use a word processing program to keep track of names and addresses."

Your data management needs may not be extensive enough to warrant buying a large, sophisticated data base manager. Or maybe there are some specific applications you could use data base software for, but don't want to take the time to work with a multipurpose data manager.

An alternative to generic data base programs—those that require you to set up your own

The ABC's Of Data Bases

Charles Brannon, Program Editor

There are several "generic" applications for microcomputers. Electronic spreadsheets such as *VisiCalc* helped to spark the microcomputer revolution. Word processing has made many a computer purchase easily justified. And data base programs are now one of the hottest items on the market. The best-selling software, such as *Lotus 1-2-3*, incorporates all three of these applications. Although packages such as *Lotus* 1-2-3 (first sold for the IBM PC) have more sophistication and scope (as well as a much higher price tag) than similar programs on home computers, there's still a lot you can do with even a bargain-basement data base.

Just as spreadsheets have made financial analysis easier, faster, and more flexible, just as word processors have blurred the distinction between rough and final drafts, data bases can make all your record keeping simple, streamlined, and fast. But first you have to translate the concepts of manual record keeping to the computer's way of doing things. Since you're bound to encounter new vocabulary and principles as you convert to electronic record keeping, it's useful to have a background in these things as you're searching for the right package for your needs. First, a clarification: Sometimes the program that manages the data is called a *data base*. The set of all your data is also referred to as the *data base*. Context usually makes the meaning clear.

Files, Records, Fields

To understand some of the features of data base management, think of how records are organized if no computer is involved. Short items are usually stored on index cards, then shoved in a box. This whole box of cards would be called a *file* on a data base. Each card is called a *record*. Records are further subdivided into *fields*. Before you can enter any information, you have to set up or define the data base by entering the name, type, and length of each field. To illustrate these subdivisions of a file, we can look at a common type of data base, the mailing list. Each record in such a file would be an individual mailing label. And, within each record, the fields would probably be: *name, address, city, state,* and *zip code*.

Many data bases ask you to estimate the length and type of each field. This way, the data base program can tally up a total for the length of each record. The memory (computer RAM or disk drive) is then subdivided into records of that length. By contrast, other data bases will assume a fixed record length (usually the size of one disk sector), then let you divide the record up into fields. You still have to estimate the length and type of each field. Some data bases can modify the fields at any time, but many data bases can't. It's often wise to set aside some extra fields in case you later want to start including additional information in each record, like telephone numbers.

Field length is based on what kind of information it will be expected to contain. People's names will generally be less than 20 characters. An address can be longer, say 30 characters. Most cities can be spelled in under 15 characters. Using the official abbreviations, the *state* field takes only two characters.

Another aspect of fields is their type—*Name, address, city,* and *state* are all string or alphanumeric fields. The zip code, however, is always a number and so it could be assigned to a *numeric* field. Numeric fields store their numbers in a fixed number of bytes, in the computer's internal floating point format. What this means to you is that no matter how long or short the number is when written out (5.2 or 5,200,000), it will only require, say, five bytes to store. Advanced data bases have several other types. A field requiring a yes/no answer can be stored with only 1 bit (1=yes, 0=no). You could pack eight yes/no answers in one byte. This is often called a boolean, bit, or binary field. Some data bases might support a byte field, which can hold only numbers in the range 0–255.

The *key field* is the primary field you use when accessing records. If the key field is *name*, you can look up any mailing label by the addressee's name. Many data bases will let you sort the entire file. You choose a key field to sort by. If you sort by *zip code*, the file will be ordered according to the zip

STOP unexplained data loss, BLOCK power line hash and AVOID damaging transient voltages with GESP-753!

Now you can plug into the double-duty protection for your sensitive personal computer! General Electric's GESP-753 Voltage Surge Suppressor guards valuable equipment two ways: against voltage spike and noise—all in one device!

With a GESP-753 Voltage Surge Suppressor, you can also guard against potentially damaging transient voltages and prevent the generation of additional noise from being sent down the power line by your equipment. Reduce unexplained data loss and block interference that raises havoc with audio or TV reception.

How spikes, surges and other gremlins attack.

Power switching devices (such as heating/cooling equipment going on and off)...utility transformers stepping up power to meet peak load needs...and/or an electrical storm in the area. Each and

all of these can play havoc with any sensitive electronic equipment—and your data input in particular—by triggering a momentary surge, a transient voltage spike, radio-frequency or electromagnetic interference.

Simple installation.

Just plug the GESP-753 into the upper outlet of any standard 110v wall grounding receptacle. A stabilizer pin fits into the grounding contact of the unused outlet to hold the unit in place. Once plugged in, a built-in light is illuminated to remind you that the protection circuits are working. Then, simply plug in the power supply cord of the equipment you want protected; up to three grounded outlets per GESP-753.

Technical Specifications

- Clamping level 325v
- Attenuation
  Up to 40 db
  (200 Khz—30 Mhz)
  Over 35 db
  (0.5 Mhz—5 Mhz)
- Response time 1 nanosecond
- Energy dissipation 50 joules

For the names of distributors and dealers in your area where you may obtain the GESP-753, circle the number shown below on the reader's service card.

We bring good things to life.

GENERAL ELECTRIC

Microwave Products Department, 316 E. Ninth St., Owensboro, KY 42301

code when printed out. Some data bases let you sort or search with multiple keys. For example, you could print out a list, alphabetized by name, of all addressees living in California. The sort would be keyed to the *name* field, and the printout would be selective by only printing and sorting those fields whose *state* field is CA.

A Range Of Features

Features vary from one data base to another. Their primary purpose is to let you store and retrieve records. But once you have a large data base, you should be able to manipulate and interrogate the data base, with all the speed and power that the computer can bring to bear. We've already mentioned sorting and printing. If you want to keep a mailing list, be sure the data base can print out mailing labels! You should also be able to remove (delete) a record once it is no longer needed. If you are manually keeping the list in a certain order, you want to be able to insert new records between existing ones. It's very handy to have a printout of just one field from every record. A directory function like this can usually print out the key field from each record for future reference. Some data bases permit you to perform math on numeric fields, even across the entire file. Others offer sophisticated report generation, where you can design a custom printout, complete with rows, columns, and calculations like totals and percentages.

You may want to be able to access your data base file from within other programs. To do this, a word processor, for example, must be compatible with the data base, so that information can be retrieved and inserted into the text held by the word processor. Or, if your data base can create files that are compatible with the word processor, you're in business. If this is important to you, try out both the word processor and the data base together before you buy either.

If you have a cassette system, you'll probably want to buy a memory-storage data base. These store the records in RAM. The advantage here is speed. You can sort and search for records much faster in memory than you can with a tape drive. At the end of the session, you save out the entire data base to tape. The disadvantage here is that this limits the amount of data which can be stored to the amount of RAM you have in your computer. Also, memory-based data bases often slow to a crawl when their memory is full.

If you have a lot of data to store, you'll probably need a disk-based data base. The records are stored directly on disk, and any record can be called up without reading through the entire file. You can usually use the whole disk for a single data base, or even link the data base to a second disk or disk drive. The disadvantage with a disk-based data base is the speed of disk access time, which is generally much slower than a memory data base.

Be critical of the data-entry mode. You'll be using that part of the data base more than any other as you type in all the data. Grade a program's entry mode in terms of how easy it is to learn, how easily you can edit and make changes, and how it reacts to errors you make. Does it check to make sure the field you've entered is of the correct type for that field? Does it warn you if you've typed too much for one field, or does it just chop off the extra characters?

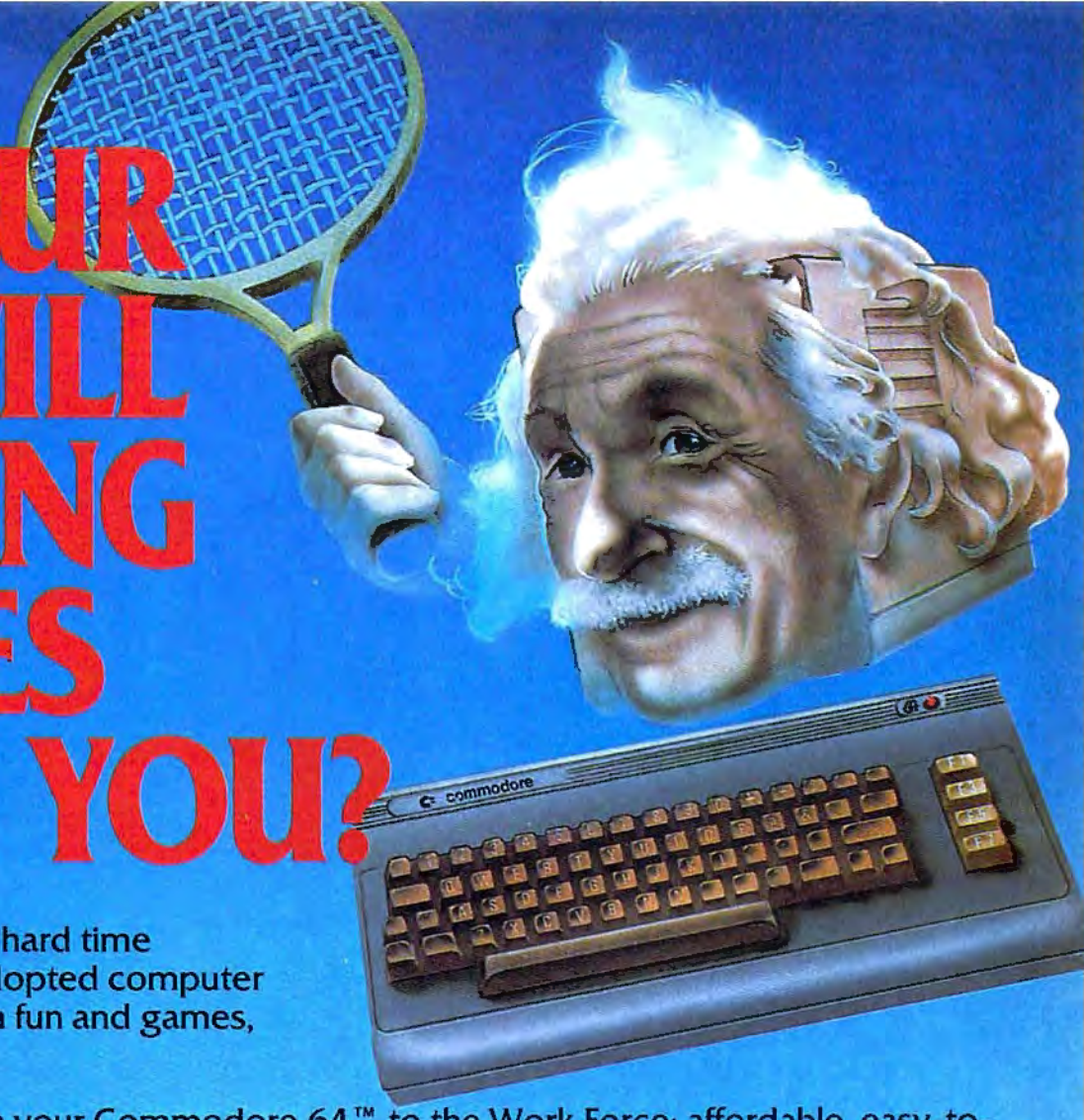
Some data bases, like *dBASE II*, are so sophisticated and flexible that they are practically a programming language for data base management. Many people buy templates for them. Templates are like programs for the data base. The template sets up all the fields, and includes the search and calculation descriptions. A template can also control the kinds of printouts allowed. You don't need to be a programmer to use an advanced data base, but you do have to learn the commands and protocols of that particular program. But once you've set up a certain kind of file, a template could create replications of that file type automatically the next time you want to build a similar data base.

files—is application-specific data base software.

Batteries Included offers such a series for the Commodore 64. At $29.95 each, these "mini-data bases" offer tailor-made filing systems ranging from Electronic Address Book to Recipes to Audio/Video Catalog. Eight different packages are currently available.

No one could call data base software faddish. After all, it facilitates one of the fundamental computer functions—information processing. And, in one form or another, data bases have been around for a long time, albeit in low-tech forms like filing cabinets and boxes of index cards. Data managing software offers a fast, effective method of storing, sorting, and searching all kinds of information. ©

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

The Promise Of Things To Come:

Atari's New Lease On Life

Fred D'Ignazio, Associate Editor
Selby Bateman, Features Editor

James Morgan, chairman and chief executive officer of Atari, Inc.

Out of the ashes of a disastrous 1983, a slimmer and more serious Atari, Inc., is fashioning a comeback under the guiding hand of new chairman and CEO James Morgan. In this, the first of a two-part look at Atari and its new products, Morgan talks candidly to COMPUTE! *about his company's mistakes, its strengths, its hopes.*

When an especially strong earthquake recently shook the California city of Sunnyvale, most of the residents shrugged, smiled nervously, and tried not to think about the next one.

But among the hundreds of people who work for Atari in more than two dozen nondescript buildings there, the quake appeared to be hardly noticed. When you've already had the world turned upside down and are feverishly working to restore your corporate footing, a little more trembling scarcely seems worth worrying about.

In retrospect, the earthshaking that Atari took from the end of 1982, through 1983, and into the early part of 1984, seems to have had the same sort of explosive force that first powered the company into becoming a billion-dollar organization. Almost overnight, Atari went from being king of the videogame and home computer market to being every analyst's example of the boom-and-bust potential inherent in the computer revolution.

World-Class Problems

The litany of problems was indeed world-class: over half a billion dollars in losses for the first three-quarters of 1983, premature announcements of several products that never appeared, the unsuccessful launch of the 1200XL computer, layoffs of hundreds of Atari employees, and a management team wracked with dissension, low morale, and a lack of corporate focus.

Enter James Morgan, a former Phillip Morris marketing executive, who replaced Ray Kassar in September 1983. His mandate from parent company Warner Communications was as simple to state as it was difficult to carry out: Turn Atari around.

"Before I came, this company thought it was a toy company, IBM, and everything in between," says Morgan. "And it was devoting people and resources to all of that."

Energy, Hope, And Resolve

Morgan has not gone about his cleanup at Atari quietly. Instead, he has become one of the most outspoken critics of the company's past policies. He often sounds more like an irate consumer than a computer company president.

In his effort to reshape Atari, Morgan laid off an additional 250 employees last winter, including Chris Crawford, Atari's highly regarded research-and-development director. Also, Atari's chief scientist, Alan Kay, left the company in the spring to join Apple Computer as an Apple Fellow.

Despite these changes—and in some cases because of them—Morgan appears to have brought new energy, new hope,

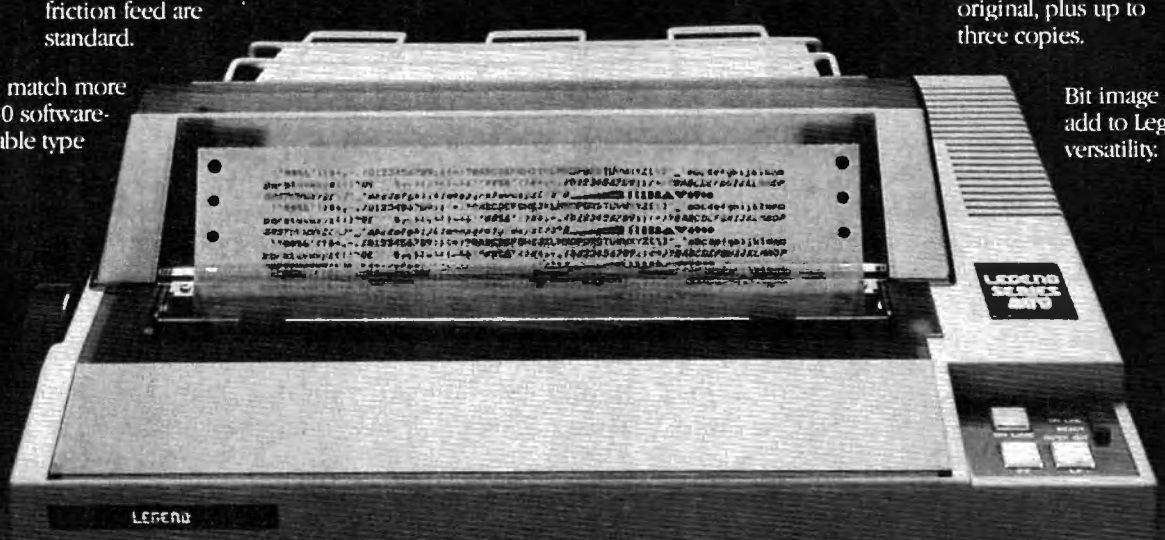
The Making Of A Legend.

Both tractor & friction feed are standard.

Prints a crisp original, plus up to three copies.

Mix & match more than 40 software-selectable type styles.

Bit image graphics add to Legend's versatility.



Centronics parallel interface standard; RS232C optional.

Serial impact dot matrix in two models: 80 & 100 cps.

New square dot technology for higher resolution & near letter-quality print (Model 800).

Up to 142 columns in compressed printing mode.

Bi-directional logic seeking for fast output.

Features like these make our new dot matrix impact printer a Legend. While a low price makes it a near miracle! Imagine, all this and more for less than $350. That puts you into our 80-cps Legend 800 model. And if you're looking for something even faster, look into our 100-cps Legend 1000.

See them both at a dealer near you. Or drop us a line for facts by mail: CAL-ABCO/PERIPHERALS DIVISION, 14722 Oxnard Street, Van Nuys, CA 91401. Telephone (818) 994-0909. Toll free 1-800-321-4484. Telex 662436. Dealer inquiries invited.

LEGEND

PERIPHERAL PRODUCTS

Toll Free 1-800-321-4484

and a new resolve to Atari's efforts. With an enthusiasm that has been missing for over a year, Atari employees and executives this spring were eagerly preparing for June's Consumer Electronics Show in Chicago and for the 12 months following—commonly acknowledged as the period during which Atari must show the world and Warner Communications that it is back on the right track.

"The Public Still Loves Us"

Morgan is excited—not only about Atari's future plans—but about the company's current strengths, which he believes have been largely overlooked.

"The financial analysts and the computer press have been disenchanted with Atari for several months, but the public still loves us. This gives us a franchise from the consumer to develop the type of microprocessor products that the consumer will want," he says.

"Before we could announce any products though, we had to get a sense of our own self-identity. Who are we? What are our strengths?"

All has not been bleak for Atari. The 600XL and 800XL computers have sold well. In fact, Morgan told analysts earlier this year that Atari could have sold about 40 percent more computers during the Christmas rush if they had been available to ship. And despite dire predictions about the death of the videogame machine, Atari seems confident that this market is stronger than some analysts have estimated.

AtariSoft And Atari Learning Systems

Atari's market share began climbing this past spring. And the company's software division, AtariSoft, and its educational division, Atari Learning Systems, both appear to be doing well.

The AtariLab computer science kit, with its temperature module, brings science into the real world for computer users and is one indication of Atari's commitment to quality educational software.

Linda Gordon, who directs the Atari Learning Systems Group, has a strong team, including Dorothy K. Deringer, formerly program officer with the National Science Foundation. In the burgeoning educational software field, Atari expects this division of the company to offer some of the most innovative and high-quality products for schools and home learning that will be available in the industry.

Products like the recently released AtariLab, a computerized science kit, and a series of other products similar in scope and quality (being introduced at CES) are creating excitement and momentum within the entire Atari organization.

Morgan is quick to point out what he feels are a few of Atari's underlying strengths. "First, the combination of color graphics and sound in Atari computers is better than in our competitors' computers. Second, more people are familiar with Atari than with any other computer company. Remember, 16 million Americans have an Atari computer—a 2600 video-computer system—in their home," he says.

"Third, when people think of Atari, they think of entertainment. That is a tremendous advantage, but not just so we can sell more videogames. Computers can make learning more entertaining. They can even make work more entertaining—as well as more productive."

The Computer Of 1990

But Morgan is frank about what he feels Atari must do in the future to reestablish itself as a creative and credible force in the microcomputer field. A committee Morgan chairs at Atari, called "The Computer of 1990," meets frequently to brainstorm about future directions. Division heads and product managers reportedly have more communication with one another than in the past. And products or strategies that once went unquestioned, have all undergone Morgan's scrutiny.

For example, the popular

Atari Program Exchange (APX), a division of the company which purchased, produced, and marketed consumer-written programs for Atari computers, has been drastically reshaped.

"Atari has redeployed some of its resources and programs so that they are more consistent with the current goals of the company," says Morgan. "In the case of APX, Atari has discontinued the mail-order portion of the program. Atari lost money in this portion of the business.

"Moreover, Atari had to come to grips with the fact that Atari is not in the mail-order business. However, APX will continue to review products sent to Atari by outside programmers," he says. "If the programs are topnotch, they will be added to the main Atari catalogue. Otherwise, they will not be sold by Atari in any fashion."

The Fate Of The 1450XLD?

Morgan also took a hard look at Atari's plans for a high-end computer. The 1400XL and the 1450XLD, announced at the June 1983 Consumer Electronics Show (CES), were never released. The 1400 was unceremoniously dropped, and the 1450, although exhibited at the January CES, was not yet on the market.

"Atari will sell a high-end computer in 1984," Morgan now says, "but the specific product features of that high-end machine still are under review. We showed the 1450XLD at the Consumer Electronics Show in January of 1984 to demonstrate our intent to market a high-end machine this year."

In fact, by the time you read this, Atari may well be marketing such a computer. And this points to one of the major changes Morgan has instituted at Atari: "We want Atari to be seen as the consumer's friend," he says. "That means we don't announce any products unless we are willing to back them 100 percent."

Enhancing Lives Through Interactive Electronics

Morgan also makes it clear that Atari has no intention of abandoning the computer market.

"That's the real tragedy of Atari. Despite a record of several excellent computers, we are still known as a videogame company" he says. "But we're going to change that. Over the next 18 months, we will be introducing a host of new products that will create an awareness and acceptance of Atari as being a superior computer manufacturer."

While Atari's product line will be more focused than in the past, the company's new strengths will have a broader base, Morgan suggests. "Our goal isn't to just produce computers. It is to produce products that enhance consumers' lives through interactive electronics."

"Invisible" Computers

"To think this way, we have to think beyond user friendliness and beyond desktop computers. We have to think of products that are *invisible.*

"For example, a truly friendly product should not separate you from the task at hand. It should be like a refrigerator—you just reach inside the door and get what you need. After all," he says, "the product, any product, is not a hero. It is just a medium. It is the carrier of what is important."

Morgan clearly expects June's CES show in Chicago to be a major first step in the company's introduction of new products aimed at carrying Atari back to critical and financial success. But he has not limited Atari to the introduction of products at trade shows.

Tuning In To The Consumer

In early May, Atari announced new Lucasfilm games—*Ballblazer* and *Rescue on Fractalus*—which Atari has developed in association with the special-effects wizards at the well-known motion picture company. And by the time you read this, Atari is scheduled to have premiered a new high-end game machine, the 7800 Pro System. Both of these new products were scheduled to be shown at CES in June as well.

"Our major priority at Atari is to tune in to the consumer. Ultimately, the home computer is not an entity unto itself. It is not a question of what a computer can do. It is a question of what a consumer does with it," Morgan says.

"In my opinion, we still have not given consumers a compelling reason to buy a computer. And we haven't spent enough time molding our products to consumers' desires."

Atari's "Smart" Telephone

"For example, most people like to communicate with other people," he says. "That is a real need and a real desire. And computers can help people communicate. But it's not easy. You have to type all sorts of special codes and commands, just to get started. Instead, it should be just as easy as using a phone. You should be able to press a couple of buttons and communicate."

Morgan says that AtariTel, the company's telecommunications division, will introduce "smart telephones" in the second half of 1984. "These telephones will be microprocessor-based. We currently are deciding how we will market the product," he adds.

While redirecting Atari's efforts, Morgan has also studied the microcomputer industry as well. And one of the major

problems still troubling the industry, he emphasizes, is that home computer technology is ahead of the average consumer without matching the consumer's real needs. The challenge, therefore, is for computer manufacturers to translate this new technology, while at the same time giving prospective buyers genuine reasons to purchase a computer.

Alan Alda Is The Bridge

Assisting in Atari's efforts to explain its computers is actor Alan Alda, who represents what Morgan calls a "bridge" to adults by selling the application of Atari technology, and the ease of use.

"He [Alda] always picks one activity, like word processing, or education, and shows you how you can do it on an Atari. Alan doesn't want to make adults buy computers because they feel guilty. He wants them to buy a computer because they're excited about doing something they have seen *him* do," says Morgan.

"Also, he never sells RAM, ROM, or CPUs. The CPU is the least important element in the computer. It is like the engine of a car. Most people buy a car without opening the hood," he says. "There is a common understanding among car owners that the engine will work, and it will get them where they're going. The CPU is like the engine. You've got to have it, but you don't sell computers because of it alone."

Morgan's Open Letter

Morgan's impact at Atari has not only been felt directly by his employees. He has also gone out of his way to be accessible to industry analysts, the press, and—most importantly—the thousands of loyal Atari owners who are both a present and future market and a formidable, knowledgeable circle of critics.

A personal and candid letter from Morgan to Atari owners popped up on the message section of CompuServe earlier this year, for example. In the note, Morgan thanked them for their support and criticism, explained his view of Atari's past problems, and requested their continued interest in Atari's future.

This kind of attention to personal detail, and the simultaneous redirection of Atari's efforts, have done much to restore the morale among Atari employees and have helped give the company valuable time in which to develop, and properly introduce, new products.

Frank Questions And Open Communication

Morgan seems to understand that his role must be multifaceted. "I act as a catalyst to the Atari management team, which has the real job of running this company," he says. "I try to set the tone for the management committee and I help point the group in certain directions.

"As someone who joined this industry as an outsider, I have been able to take a fresh look at the entire consumer electronics field in general and this company, in particular. I'm not afraid to ask frank questions, and also question why we do things the way we do. I encourage all Atari employees to examine their own work in the same way.

"I believe strongly that one of my biggest contributions to Atari will be the implementation of a corporate culture here that inspires teamwork and open communications," Morgan says. "I want to encourage people to take calculated risks and not be afraid to fail. That's part of being an excellent company. If we become an excellent company, then sales and profits will follow."

(Next month, COMPUTE! will take an in-depth look at Atari's new product line from the Summer Consumer Electronics Show.)

Share the Olympic Experience.

"Standing on the top of the Olympic victory stand is like stretching one's body on the top of the world. It is a moment where the individual man or woman gets introduced to the whole planet. It is a moment that is his or hers alone."

—Olga Connolly
Gold Medalist, 1956 Olympics

This summer, the Olympic torch will return to Los Angeles after 52 years. The stage is set. Some 10,000 athletes from 150 countries will battle for the gold in the historic Games of the XXIII Olympiad. And whether or not you plan to attend the Games, you can participate in the drama of this once-in-a-lifetime spectacle.

You and your friends will have virtual front-row tickets with the *Official Olympic Guide to Los Angeles*, a lavish 300-page magazine that captures it all—the thrills of great Olympic moments past and present, future stars and hopefuls, records to break, maps, TV and day-by-day events schedules.

Everything from the magic of opening ceremonies to the final lap of the marathon, along with an indispensable guide to Southern California, including top attractions, dining, nightlife, tips and trips. All in breathtaking full-color photography and illustrations.

You, and your out-of-town friends, will share all the action of this memorable event with this special collector's issue that will become a cherished keepsake of the legendary rekindling of the Olympic flame in Los Angeles.

Order now and be ahead of the crowd. Send in your check or money order. Or call 557-7587 and charge it to your MasterCard or Visa. Official Olympic Guide, 1888 Century Park East, Suite 920, Los Angeles, California 90067.

Send me _____ copy/copies of the* Official Olympic Guide to Los Angeles *at $5.95 each, plus $1.00 shipping and handling. Enclosed is my check or money order.

NAME _____

ADDRESS _____

CITY _____

STATE _____

☐ **Yes! I'd like the beautifully designed gift box for an additional $1.00. Please send me _____ boxes. I've enclosed an extra $ _____.**

Statistics For Nonstatisticians

A Burke Luitich

Basic statistical methods can help you make logical decisions in everyday situations.

For the most part, elementary statistical methods measure a group of similar things to see how these measurements vary when compared to some standard. Another use for statistics is to see how creating a group of objects can cause variations in these objects.

This program, "Statistics," takes your raw data and returns figures which you can use to make everyday decisions, for example, about the best way to build a wall or how much cash you'll need when you go shopping.

As a first example, let's look at two ways to cut a 2 x 4, by using a power table saw and a handsaw. We set the table saw guide to one foot and cut five pieces. We cut five more pieces using a handsaw, then measure the actual lengths of all ten pieces to see how accurately we made the cuts.

If nothing unusual is allowed to affect the cutting, we can expect the length of the pieces to vary depending on the process used. Statisticians call this an *unbiased random sample*.

Assume the measurements are as follows:

Table saw lengths (feet)	Handsaw lengths (feet)
1.05	1.22
.98	.91
1.03	.80
1.07	1.28
.96	.88

The Same Mean

A look at the values alone suggests that cutting with the handsaw is a far less consistent method than using the table saw. However, if you add up the lengths for each method and divide by 5 (the total cuts for each) you will find that both methods give the same *mean* (average) length of 1.018 feet.

Just finding an average length doesn't tell us much. What we need to know is how widespread the values are likely to be, and which method gave us the most lengths that were nearer our standard of one foot. In statistical terms, we need to calculate the *range* and the *standard deviation*.

We find the range by subtracting the shortest length from the longest, for each cutting method. For the handsaw the range is .48 feet (1.28 –.80), and for the table saw the range is .11 feet (1.07 – .96). Immediately, we can see that the table saw cut more consistently, because the range, or variation, is smaller.

We can use the standard deviation and the mean length to predict how often a given length is likely to occur. You don't have to worry about how to calculate a standard deviation; the program does this for you. If you type in the above lengths for the handsaw, the program will return a standard deviation of .217 feet. The standard deviation for the table saw is .047 feet.

Degree Of Accuracy

If we made a large number of cuts, then measured and graphed the lengths, the graph would form a bell curve, or normal distribution. By combining the standard deviation and the mean length, we get a range of lengths that includes 68.3 percent of all lengths (again, you don't have to know the theory; just use the number). To illustrate, first take the mean length, 1.018 feet, and subtract from it the standard deviation for the handsaw, .217 feet, to get .801 feet. Then add the standard deviation to the mean length to get 1.235 feet. This means that 68.3 percent of our lengths fall in the range between .801 and 1.235 feet.

By adding and subtracting the standard deviation (.047 feet) with the mean length of the table saw cuts (1.018 feet), we find that 68.3 percent (roughly two-thirds) of these lengths fall in the range from .971 to 1.065 feet.

If you want a wider sample, you must increase

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

the number of standard deviations. To include 95.4 percent of all lengths, use two standard deviations. For the handsaw, we now have .434 feet, two standard deviations. Combining it with the mean length, we get a range of .584 to 1.452 feet. Our table saw range becomes .924 to 1.102 feet (1.018 plus and minus .094).

Food For Thought

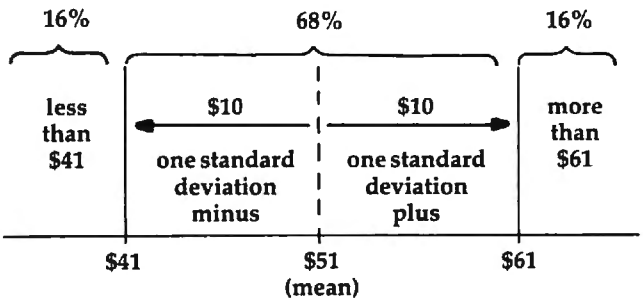
You can use the same methods to calculate a food budget. In this case, your data consists of the amounts you spent on groceries over a 13-week period (one-fourth of a year):

Week	Amount	Week	Amount
1	$42	8	47
2	50	9	65
3	75	10	49
4	37	11	43
5	51	12	52
6	45	13	54
7	56		

If you type this data into the Statistics program, you will find that your mean amount spent was about $51; that your spending varied from $37 to $75, for a range of $38; that you spent more than $50 (your median amount) as often as you spent less than that; and your standard deviation is about $10.

Applying The Statistics

Combining one standard deviation and the mean (or average) amount spent, we find that two-thirds of the weeks you spend between $41 and $61 at the grocery store. One-sixth of the time you spend less than $41; one-sixth of your bills are more than $61. So, if you budget $61 for groceries, you'll have enough 84 percent of the time.



If you want to be sure you'll have enough in case prices rise, you might want to use two standard deviations. By adding two standard deviations ($20) to the mean amount ($51), you will find that, to be about 98 percent sure, you should budget $71 each week.

There are other factors to be considered, of course, such as vacations, birthday parties, or visiting relatives, that can affect your food budget. The Statistics program does not take these kinds of things into account. But it does give you a tool which takes some of the guesswork out of everyday decision-making.

The Statistics program requests input of the size of the sample, or number of items to be entered (line 410), then requests the values of the sample measurements (lines 500–550). All the statistics referred to in this article are then calculated, that is, mean, standard deviation, median, and range.

Lines 325–350 and 4900–5610 give the user a thumbnail sketch of the information to be calculated and a description of each of the statistics. While the sample size is limited to 100 for the VIC version (other versions allow up to 300), this should be more than adequate in most cases.

Error Correction

An error correction routine is included in lines 555–580 and 5900–6190. This provides for the change of any entry before the calculation. While the program is running, a delay of up to two minutes will be experienced while the program performs several sorts on the data. This is normal for BASIC and may be longer for sample sizes in the 80 to 100 range or greater.

Program 1 requires at least 3K of expansion memory in the VIC computer. If the instructions, error correction routine, and headings are eliminated, the program will run on an unexpanded VIC. Specifically, the following lines should be deleted if the program is to run without memory expansion: 95–180, 325–350, 555–580, 4900–5610, and 5900–6190.

Further reductions can be made by reducing the sample size, redimensioning the array in line 90 to the new sample size (SA), and changing the value of 100 in line 420 to the new maximum sample size.

Statistics for a sample of 100 readings requires about 30–45 minutes to calculate by hand. This program requires about 8–10 minutes, including input.

Program 1: VIC Statistics

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

```
90 DIM SA(100)                          :rem 185
95 REM GENERAL INTRODUCTION             :rem 242
100 PRINT "{CLR}":POKE 36879,126:PRINT "
    {BLK}"                              :rem 207
110 FOR K=1 TO 3:PRINT:NEXT K           :rem 147
120 PRINT TAB(4);"{4 DOWN}{RVS}VIC STATIS
    TICS{OFF}"                          :rem 208
130 PRINT TAB(9);"{DOWN}FOR"            :rem 249
140 PRINT TAB(2);"{DOWN}{RVS}NON-STATISTI
    CIANS{OFF}"                         :rem 171
180 FORK=1TO2000:NEXTK                   :rem 98
190 PRINT"{CLR}"                        :rem 254
200 PRINT TAB(4);"{2 DOWN}THIS PROGRAM"
                                        :rem 108
210 PRINT TAB(3);"CALCULATES THE":rem 188
220 PRINT "{2 SPACES}FOLLOWING VALUES"
                                        :rem 230
```

Strengthen your hand with Superbase 64™

The complete information control system for the Commodore 64.

The World Famous Commodore 64.

No matter what your business or interest, with Superbase 64 you have a totally flexible 'record' system, as big as you want it, as fast as you need it.

Create your own formats, enter your records, change layouts and datafields.

Superbase gives you unrivalled control in home or office, business or professional practice, with a range of features including:

YOUR OWN RECORDS

- Design your layout using text, numeric, calculated result, date, linking and key fields
- Record size up to 1100 characters spread over up to 127 items
- Number of records limited **only by your equipment**. Can be thousands
- As many databases as you want – each with up to 15 files
- Learn fast through built-in HELP screens – then add your own notes

FAST ACCESS

- Keyed records for high speed retrieval – 3 secs for any file size
- Select and search for any name, description, number, date, etc. in multiple combinations
- Sort records into any order
- Display selections or generate printed reports
- Browse through records matching on any criteria

TOTAL CONTROL

- Links to other programs and EASY SCRIPT for mailshots, high-quality letters, quotes, tables, etc.
- Effective management of invoices, addresses, stock, membership, appointments – any and every kind of record
- English like commands for easy conversational programming, plus built-in BASIC

DATABASE MANAGEMENT

- Easy to understand menus
- Add or amend fields, or alter length – no file rebuilding needed
- Update files with automatic batch processing option
- Calendar arithmetic for effective time management
- Display quantities, values, totals, as you enter them. Formulas for on-screen result calculation

Superbase 64™

Precision Software

Precision Software (USA) Inc.,
Suite 1100
820, Second Avenue,
New York, N.Y. 10017
Tel: (212) 490 1825

OFFICIALLY APPROVED BY COMMODORE

commodore

```
230 PRINT "FROM DATA YOU INPUT:" :rem 123
240 PRINT TAB(4);"{2 DOWN}1. MEAN"
                                  :rem 160
250 PRINT TAB(4);"{DOWN}2. STANDARD"
                                  :rem 193
260 PRINT TAB(7);"DEVIATION"      :rem 166
290 PRINT TAB(4);"{DOWN}3. MEDIAN":rem 35
300 PRINT TAB(4);"{DOWN}4. RANGE":rem 219
310 PRINT"{3 DOWN} PRESS {RVS}C{OFF} TO C
    ONTINUE"                      :rem 20
320 GET Z$:IF Z$<>"C" THEN 320    :rem 255
325 REM INSTRUCTION REQUEST        :rem 9
330 PRINT"{CLR}":POKE214,10:PRINT:POKE211
    ,5:PRINT"DO YOU NEED"         :rem 99
340 PRINT"{DOWN} INSTRUCTIONS (Y/N)?":GOS
    UB7010                        :rem 22
350 IF Z$="Y" THEN GOSUB 5010     :rem 244
355 REM DATA ENTRY                :rem 45
370 PRINT"{CLR}ENTER YOUR DATA ONE"
                                  :rem 199
380 PRINT"VALUE AT A TIME,"       :rem 26
390 PRINT"THEN PRESS RETURN."     :rem 55
410 POKE214,10:PRINT:POKE211,0:INPUT"ENTE
    R SAMPLE SIZE";N              :rem 26
420 IFN>100ORN<2THEN5710          :rem 155
430 PRINT:PRINT"{CLR}{4 DOWN}{RIGHT}IF YO
    U MAKE AN ERROR"              :rem 230
440 PRINT"{2 DOWN}{RIGHT}CONTINUE WITH EN
    TRY."                          :rem 9
450 PRINT"{2 DOWN}{RIGHT}YOU WILL BE ABLE
     TO"                          :rem 28
460 PRINT"{2 DOWN}{RIGHT}CORRECT LATER."
                                  :rem 98
470 PRINT"{2 DOWN}{RVS}{BLK}{4 RIGHT}PRES
    S ANY KEY{OFF}"               :rem 148
480 GETZ$:IFZ$=""THEN480          :rem 141
500 FOR I=1 TO N                   :rem 36
520 PRINT"{CLR}":POKE214,11:PRINT:POKE211
    ,0:PRINT"DATA ENTRY #";I;:INPUTR$
                                  :rem 130
530 SA(I)=VAL(R$)                 :rem 173
550 NEXT I                         :rem 34
555 REM ERROR CORRECTION REQUEST   :rem 46
560 PRINT"{CLR}":POKE214,11:PRINT:POKE211
    ,0:PRINT"DO YOU WISH TO MAKE" :rem 68
570 PRINT"ANY CORRECTIONS(Y/N)?":GOSUB 70
    10                            :rem 136
580 IF Z$="Y" THENPRINT"{CLR}":GOTO6000
                                  :rem 80
585 REM CALCULATION OF MEAN AND STD. DEVI
    ATION                         :rem 250
587 PRINT"{CLR}":POKE214,9:PRINT:POKE211,
    5:PRINT"{RVS}PLEASE WAIT{OFF}":rem 48
588 POKE214,11:PRINT:POKE211,3:PRINT"STAT
    ISTICS BEING"                 :rem 151
589 POKE214,13:PRINT:POKE211,6:PRINT"CALC
    ULATED"                       :rem 255
590 FOR I=1 TO N                   :rem 45
600 SUM=SUM+SA(I)                  :rem 22
610 NEXT I                         :rem 31
620 MEAN=SUM/N                    :rem 104
630 FOR I=1 TO N                   :rem 40
640 DFF=DFF+(SA(I)-MEAN)↑2        :rem 255
650 NEXT I                         :rem 35
660 SDDEV=SQR(DFF/(N-1))          :rem 146
665 REM SORT OF DATA INTO NUMERIC ORDER
                                  :rem 69
670 FL=0                          :rem 187
680 FOR I=1 TO N-1                :rem 139
690 IF SA(I)<=SA(I+1) THEN 740     :rem 41
700 Q=SA(I)                        :rem 83
710 SA(I)=SA(I+1)                 :rem 141
720 SA(I+1)=Q                     :rem 177
730 FL=1                          :rem 154
740 NEXT I                         :rem 35
750 IF FL=1 THEN 670              :rem 247
755 REM CALCULATION OF RANGE      :rem 182
760 RG=SA(N)-SA(1)                :rem 233
765 LR=SA(1)                      :rem 147
767 HR=SA(N)                      :rem 174
805 REM CALCULATION OF MEDIAN     :rem 243
810 IF N/2 <>INT(N/2) THEN 814     :rem 8
811 IF SA(N/2)=SA(N/2+1) THEN MDD=SA(N/2)
                                  :rem 191
812 IF SA(N/2)<>SA(N/2+1) THEN MDD=(SA(N/
    2)+SA(N/2+1))/2               :rem 202
813 GOTO 1310                     :rem 154
814 MDD=SA(INT(N/2+1))            :rem 219
1290 REM PRINT RESULTS TO SCREEN :rem 210
1310 PRINT"{CLR}CALCULATION RESULTS"
                                  :rem 138
1320 PRINT"**********************";
                                  :rem 110
1330 PRINT "{DOWN}SAMPLE SIZE";SPC(3);N
                                  :rem 212
1340 PRINT"{DOWN}MEAN (X BAR)";SPC(2);INT
     (MEAN*10000+.5)/10000         :rem 78
1350 PRINT"{DOWN}STD. DEV.";SPC(5);INT(SD
     DEV*10000+.5)/10000           :rem 46
1360 PRINT"{DOWN}MEDIAN";SPC(8);INT(MDD*1
     0000+.5)/10000                :rem 25
1370 PRINT"{DOWN}RANGE";SPC(9);RG:rem 153
1375 PRINT"{DOWN}LOWEST SAMPLE";SPC(1);LR
                                  :rem 206
1380 PRINT"{DOWN}HIGHEST SAMPLE";HR
                                   :rem 81
1480 POKE214,21:PRINT:POKE211,4:PRINT"
     {RVS}PRESS ANY KEY{OFF}";    :rem 142
1490 GET A$:IF A$="" THEN 1490    :rem 191
1495 REM REQUEST TO CONTINUE OR END
                                   :rem 96
1510 PRINT"{CLR}":POKE214,11:PRINT:POKE21
     1,0:PRINT"DO YOU WISH TO PROCESS"
                                  :rem 113
1520 PRINT"MORE DATA (Y/N)?":GOSUB 7010
                                  :rem 206
1530 IF Z$<>"Y" THEN POKE36879,27:PRINT"
     {CLR}":END                   :rem 160
1540 CLR:GOTO 330                 :rem 180
4900 REM INSTRUCTION SUBROUTINE    :rem 35
5010 PRINT"{CLR}{DOWN}THE MAXIMUM NUMBER"
                                    :rem 3
5020 PRINT "OF ENTRIES YOU CAN"    :rem 22
5030 PRINT "MAKE IS 100. THE"     :rem 243
5035 PRINT "MINIMUM IS 2."        :rem 182
5040 PRINT "{DOWN}**********************"
                                  :rem 29
5050 PRINT "{DOWN}THE MEAN IS THE":rem 43
5060 PRINT "ARITHMETIC AVERAGE"   :rem 129
5070 PRINT "OF THE NUMBERS YOU"    :rem 44
5080 PRINT "ENTER."                :rem 74
5090 PRINT "{DOWN}**********************"
                                   :rem 34
5100 PRINT "{DOWN}STANDARD DEVIATION"
                                  :rem 156
5110 PRINT "IS A MEASURE OF HOW"   :rem 10
5120 PRINT "WIDELY YOUR NUMBERS" :rem 210
5130 PRINT "SPREAD FROM THE AVG.":rem 122
5140 PRINT"{2 DOWN}{2 RIGHT}PRESS 'M' FOR
      MORE"                       :rem 57
5150 GET A$:IF A$<>"M" THEN 5150  :rem 67
5160 PRINT "{CLR}{DOWN}SINCE THE VALUES Y
```

SEE THE LIGHT!

Quality Software that Outshines the Rest

Programs for the Commodore 64

dMOS™

Database Management Operating System

The BEST data base management tool for the collection, arrangement and display of alphanumeric data.

A unique pattern matching and searching capabilities make dMOS™ the easiest DB system available on the market.

Features:

- Map search technique to achieve a "logical AND", while searching between fields.
- Display records found, or NOT found by a search.
- Rearrange fields.
- Suppress fields and field titles.
- Insert short (10 character) text.
- Selectively print records.

$39.95

SoftLoc™

The Program Security System

Set up program security in minutes.

Lock up your personal, financial or business records.

3 types of protection:

- Re-encodes program.
- Modifies diskette directory.
- Selectable and reassignable 7 digit access code.

$29.95

and announcing. . .

Missing Key™

A reset System
Restores your BASIC Program.

After programming for hours you press RUN for a final check of your work — the computer locks up. You press RUN, STOP. . . nothing — you press RESTORE. . . nothing — you look for the missing key but it isn't there. You have to turn off your computer and lose hours of work!!

Now Add the Missing Key™:

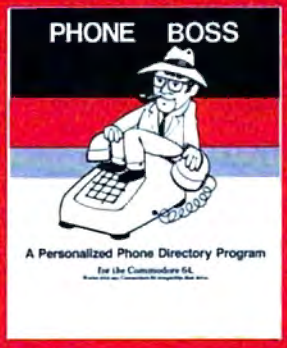
- Press the "Missing Key™" and the computer resets itself from any lockup, and your BASIC program is restored.
- Load and run the program included.
- Takes nothing away from your computer, neither memory nor a plug-in port.
- Attaches to your C64 keyboard or any other convenient location.
- Will not void your CBM warranty.

$29.95

Phone Boss™

Personal Phone Directory Program

Designed to store and organize your personal phone listings.



The user has complete control of 15 category titles and entries.

$29.95

Software by SoftPeople ■ Programs for the Commodore 64 ■ Connecting People with Great Ideas.

SoftPeople Inc.

2042 Marshall Ave. St. Paul, MN 55104 **(612) 644-1551**

by Softpeople, Inc.

MasterCard VISA

Dealer Inquiries Invited
1-800-447-3273
Special P.O.P Packages & Prices Available

```
     OU"                              :rem 97
5170 PRINT "ENTER TEND TO FORM A"     :rem 95
5180 PRINT "BELL CURVE(NORMAL"        :rem 52
5190 PRINT "DISTRIBUTION), THE"       :rem 118
5200 PRINT "STD. DEVIATION IS A"      :rem 49
5210 PRINT "MEASURE OF THE AREA"      :rem 58
5220 PRINT "UNDER THE BELL CURVE."
                                      :rem 203
5230 PRINT "{DOWN}NO. OF STD.{4 SPACES}%
     {SPACE}AREA"                     :rem 99
5240 PRINT "DEV.(+/-)"                :rem 129
5245 PRINT "-----------{4 SPACES}------"
                                      :rem 158
5250 PRINT TAB(4);"{DOWN}1";TAB(16);"68.3
     "                                :rem 142
5260 PRINT TAB(4);"{DOWN}2";TAB(16);"95.5
     "                                :rem 146
5270 PRINT TAB(4);"{DOWN}3";TAB(16);"99.7
     "                                :rem 154
5280 PRINT TAB(4);"{DOWN}4";TAB(16);"99.9
     "                                :rem 158
5290 PRINT"{DOWN}{2 RIGHT}PRESS 'M' FOR M
     ORE";                            :rem 105
5300 GET A$:IF A$<>"M" THEN 5300      :rem 61
5430 PRINT"{CLR}{DOWN}**************************
     **"                              :rem 179
5435 PRINT "{DOWN}THE MEDIAN IS THE"
                                      :rem 191
5440 PRINT "VALUE AT THE MID-"        :rem 152
5450 PRINT "POINT OF YOUR DATA."      :rem 85
5460 PRINT"{3 DOWN}PRESS 'M' FOR MORE"
                                      :rem 21
5470 GET A$:IF A$<>"M" THEN 5470      :rem 77
5490 PRINT "{CLR}{2 DOWN}THE RANGE IS THE
     "                                :rem 35
5500 PRINT "DIFFERENCE BETWEEN"       :rem 112
5510 PRINT "YOUR LOWEST DATA"         :rem 227
5520 PRINT "VALUE AND THE HIGHEST.":rem 8
5530 PRINT "{DOWN}THE RANGE IS A QUICK-"
                                      :rem 132
5540 PRINT "AND-DIRTY ESTIMATE"       :rem 135
5550 PRINT "OF THE SPREAD."           :rem 3
5560 PRINT "{2 DOWN}THE STD. DEVIATION"
                                      :rem 96
5570 PRINT "IS MORE RELIABLE."        :rem 223
5580 PRINT "{4 DOWN}{2 RIGHT}PRESS 'S' TO
      START"                          :rem 128
5590 PRINT "{5 RIGHT}THE PROGRAM"     :rem 46
5600 GET A$:IF A$<>"S" THEN 5600      :rem 73
5610 RETURN                           :rem 172
5700 REM ERROR TRAP FOR TOO LARGE A SAMPL
     E                                :rem 184
5710 POKE214,10:PRINT:POKE211,18:PRINT"
     {4 SPACES}"                      :rem 132
5720 GOTO 410                         :rem 156
5730 REM DISPLAY CORRECTION OPTIONS
                                      :rem 237
5740 PRINT"{CLR}":POKE214,21:PRINT:POKE21
     1,0:PRINT"{RVS}{BLK}A=AMEND T=TABLE
     {SPACE}Q=QUIT{OFF}";:RETURN      :rem 36
5810 GETZ$:IFZ$=""OR(Z$<>"A"ANDZ$<>"T"AND
     Z$<>"Q")THEN5810                 :rem 35
5820 KL=0:IFZ$="T"THENKL=1:GOTO 6130
                                      :rem 96
5830 IFZ$="Q"THEN587                  :rem 130
5840 PRINT"{CLR}"                     :rem 53
5900 REM ERROR CORRECTION SUBROUTINE
                                      :rem 68
6000 POKE214,11:PRINT:POKE211,0:PRINT"REM
     EMBER INCORRECT{4 SPACES}SAMPLE # (Y
     /N)?":GOSUB 7010                 :rem 10
6020 IFZ$="N"THEN6130                 :rem 157
6030 PRINT"{CLR}":POKE214,6:PRINT:POKE211
     ,1:INPUT"THE SAMPLE #";EN$       :rem 118
6040 EN=VAL(EN$):IFEN>NOREN<1OREN<>INT(EN
     )THEN 6030                       :rem 131
6070 POKE214,8:PRINT:POKE211,1:PRINT"SAMP
     LE";EN                           :rem 179
6075 POKE214,10:PRINT:POKE211,1:PRINT"VAL
     UE=";SA(EN)                      :rem 190
6080 POKE214,12:PRINT:POKE211,1:INPUT"YOU
     R NEW VALUE";C:SA(EN)=C          :rem 184
6090 POKE214,14:PRINT:POKE211,1:PRINT"
     {2 DOWN}MORE CHANGES (Y/N)?":GOSUB 7
     010                              :rem 136
6110 IFZ$="Y"THENPRINT"{CLR}":GOTO 6000
                                      :rem 123
6120 GOTO 587                         :rem 166
6130 GOSUB5740:POKE214,0:PRINT:POKE211,0
                                      :rem 150
6140 IFKL=0THENPRINT"THESE ARE THE FIRST
     {3 SPACES}TEN VALUES.":K=1:GOTO 6150
                                       :rem 47
6145 PRINT"THESE ARE THE NEXT{4 SPACES}TE
     N VALUES."                       :rem 247
6150 POKE214,3:PRINT:POKE211,1:PRINT"ENTR
     Y";SPC(5);"VALUE"                 :rem 82
6160 FF=0:FORK=KTOK+9:FF=FF+1         :rem 247
6165 IFK>100THENK=K+9:NEXT:GOTO587
                                      :rem 113
6170 POKE214,FF+5:PRINT:POKE211,1:PRINTK;
     TAB(11);SA(K)                     :rem 15
6180 NEXT K                            :rem 89
6190 GOTO 5810                        :rem 215
7010 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")TH
     EN 7010                          :rem 201
7020 RETURN                           :rem 169
```

Program 2: 64 Statistics

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

```
80 POKE53280,0:POKE53281,0            :rem 189
90 DIM SA(300)                        :rem 187
95 REM GENERAL INTRODUCTION           :rem 242
120 PRINT"{CLR}":POKE214,10:PRINT:POKE211
    ,14:PRINT"{RVS}STATISTICS{OFF}"
                                      :rem 147
130 POKE214,12:PRINT:POKE211,17:PRINT"
    {RVS}FOR{OFF}"                    :rem 215
140 POKE214,14:PRINT:POKE211,10:PRINT"
    {RVS}NON-STATISTICIANS{OFF}"      :rem 231
180 FORK=1TO2000:NEXT                  :rem 23
200 PRINT"{CLR}":POKE214,4:PRINT:POKE211,
    14:PRINT"{CYN}THIS PROGRAM"       :rem 165
210 POKE214,5:PRINT:POKE211,13:PRINT"CALC
    ULATES THE"                       :rem 219
220 POKE214,6:PRINT:POKE211,12:PRINT"FOLL
    OWING VALUES:"                    :rem 213
240 POKE214,10:PRINT:POKE211,15:PRINT"1.
    {SPACE}MEAN"                      :rem 202
250 POKE214,12:PRINT:POKE211,15:PRINT"2.
    {SPACE}STANDARD"                  :rem 254
260 POKE214,13:PRINT:POKE211,18:PRINT"DEV
    IATION"                           :rem 245
290 POKE214,15:PRINT:POKE211,15:PRINT"3.
    {SPACE}MEDIAN"                     :rem 99
300 POKE214,17:PRINT:POKE211,15:PRINT"4.
    {SPACE}RANGE"                      :rem 29
310 POKE214,22:PRINT:POKE211,10:PRINT"PRE
    SS 'C' TO CONTINUE"               :rem 108
320 GETZ$:IFZ$<>"C"THEN 320           :rem 255
322 REM INSTRUCTIONS REQUEST           :rem 89
```

```
325 SUM=0:MEAN=0:DFF=0:SDDEV=0:RG=0
                                    :rem 152
330 PRINT"{CLR}":POKE214,12:PRINT:POKE211
    ,5:PRINT"DO YOU NEED INSTRUCTIONS (Y/
    N)?"                            :rem 128
340 GOSUB 7010                      :rem 223
350 IF Z$="Y"THEN GOSUB 5010        :rem 244
355 REM DATA ENTRY                   :rem 45
410 PRINT"{CLR}":POKE214,10:PRINT:POKE211
    ,6:INPUT"ENTER SAMPLE SIZE";N:rem 190
420 IF N>300ORN<=1THENFORI=1488TO1498:POK
    EI,32:NEXT:GOTO410              :rem 139
425 PRINT"{CLR}"                    :rem 255
430 POKE214,2:PRINT:POKE211,1:PRINT"IF YO
    U MAKE AN ERROR, CONTINUE WITH"
                                    :rem 119
440 POKE214,4:PRINT:POKE211,1:PRINT"DATA
    {SPACE}ENTRY. YOU CAN CORRECT LATER."
                                    :rem 75
500 FOR I=1 TO N                    :rem 36
520 POKE214,9:PRINT:POKE211,10:PRINT"DATA
     ENTRY #{4 SPACES}{4 LEFT}";I;:INPUTR
    $                               :rem 96
530 SA(I)=VAL(R$)                   :rem 173
550 FORJ=1450TO1468:POKEJ,32:NEXTJ:NEXT I
                                    :rem 17
555 REM ERROR CORRECTION REQUEST    :rem 46
560 PRINT"{CLR}":POKE214,12:PRINT:POKE211
    ,2                              :rem 176
561 PRINT"WISH TO MAKE ANY CORRECTIONS (Y
    /N)?"                            :rem 2
570 GOSUB 7010                      :rem 228
580 IFZ$="Y"THENPRINT"{CLR}":GOTO6000
                                    :rem 80
585 REM CALCULATION OF MEAN AND STD. DEVI
    ATION                           :rem 250
587 PRINT"{CLR}":POKE214,11:PRINT:POKE211
    ,14:PRINT"{RVS}PLEASE WAIT{OFF}"
                                    :rem 137
588 POKE214,13:PRINT:POKE211,6:PRINT"STAT
    ISTICS BEING CALCULATED"        :rem 110
590 FOR I=1 TO N                     :rem 45
600 SUM=SUM+SA(I)                    :rem 22
610 NEXT I                           :rem 31
620 MEAN=SUM/N                      :rem 104
630 FOR I=1 TO N                     :rem 40
640 DFF=DFF+(SA(I)-MEAN)↑2          :rem 255
650 NEXT I                           :rem 35
660 SDDEV=SQR(DFF/(N-1))            :rem 146
665 REM SORT OF DATA INTO NUMERIC ORDER
                                    :rem 69
670 FL=0                            :rem 187
680 FOR I=1 TO N-1                  :rem 139
690 IF SA(I)<=SA(I+1) THEN 740       :rem 41
700 Q=SA(I)                          :rem 83
710 SA(I)=SA(I+1)                   :rem 141
720 SA(I+1)=Q                       :rem 177
730 FL=1                            :rem 154
740 NEXT I                           :rem 35
750 IF FL=1 THEN 670                :rem 247
755 REM CALCULATION OF RANGE        :rem 182
760 RG=SA(N)-SA(1)                  :rem 233
765 LR=SA(1)                        :rem 147
767 HR=SA(N)                        :rem 174
805 REM CALCULATION OF MEDIAN       :rem 243
810 IF N/2 <>INT(N/2) THEN 814       :rem 8
811 IF SA(N/2)=SA(N/2+1) THEN MDD=SA(N/2)
                                    :rem 191
812 IF SA(N/2)<>SA(N/2+1) THEN MDD=(SA(N/
    2)+SA(N/2+1))/2                 :rem 202
813 GOTO 1310                       :rem 154
814 MDD=SA(INT(N/2+1))              :rem 219
1290 REM PRINT RESULTS TO SCREEN    :rem 210
1310 PRINT"{CLR}":POKE214,3:PRINT:POKE211
    ,10:PRINT"CALCULATION RESULTS"
                                     :rem 69
1320 POKE214,4:PRINT:POKE211,9:PRINT"****
    ****************"              :rem 146
1330 POKE214,6:PRINT:POKE211,7:PRINT"SAMP
    LE SIZE";SPC(10);N              :rem 122
1340 POKE214,8:PRINT:POKE211,7       :rem 25
1345 PRINT"MEAN(X BAR)";SPC(10);INT(MEAN*
    10000+.5)/10000                 :rem 113
1350 POKE214,10:PRINT:POKE211,7      :rem 67
1355 PRINT"STD. DEVIATION";SPC(7);INT(SDD
    EV*10000+.5)/10000              :rem 186
1360 POKE214,12:PRINT:POKE211,7      :rem 70
1365 PRINT"MEDIAN";SPC(15);INT(MDD*10000+
    .5)/10000                       :rem 59
1370 POKE214,14:PRINT:POKE211,7:PRINT"RAN
    GE";SPC(16);RG                  :rem 110
1375 POKE214,16:PRINT:POKE211,7:PRINT"LOW
    EST SAMPLE VALUE";SPC(2);LR :rem 245
1377 POKE214,18:PRINT:POKE211,7:PRINT"HIG
    HEST SAMPLE VALUE";SPC(1);HR :rem 34
1480 POKE214,22:PRINT:POKE211,13:PRINT"
    {RVS}{YEL}PRESS ANY KEY{OFF}{CYN}"
                                    :rem 193
1490 GET A$:IF A$="" THEN 1490      :rem 191
1495 REM REQUEST TO CONTINUE OR END
                                     :rem 96
1510 PRINT"{CLR}":POKE214,12:PRINT:POKE21
    1,3                             :rem 221
1520 PRINT"WISH TO PROCESS MORE DATA (Y/N
    )?"                              :rem 73
1530 GOSUB 7010                      :rem 17
1535 IFZ$="N"THENPRINT"{CLR}":END :rem 78
1540 FORI=1TON:SA(I)=0:NEXT:GOTO 325
                                    :rem 180
4900 REM INSTRUCTION SUBROUTINE      :rem 35
5010 PRINT"{CLR}":POKE214,3:PRINT:POKE211
    ,6                              :rem 175
5020 PRINT"THE MAXIMUM NUMBER OF ENTRIES
    {SPACE}YOU"                      :rem 12
5030 POKE214,5:PRINT:POKE211,1:PRINT"CAN
    {SPACE}MAKE IS 300. MINIMUM NUMBER I
    S 2."                            :rem 73
5050 POKE214,9:PRINT:POKE211,5:PRINT"THE
    {SPACE}MEAN IS THE ARITHMETIC AVERAG
    E"                              :rem 137
5070 POKE214,11:PRINT:POKE211,1:PRINT"OF
    {SPACE}THE NUMBERS YOU ENTER."
                                    :rem 135
5100 POKE214,15:PRINT:POKE211,5:PRINT"STA
    NDARD DEVIATION IS A MEASURE OF"
                                    :rem 198
5120 POKE214,17:PRINT:POKE211,1:PRINT"HOW
     WIDELY YOUR NUMBERS SPREAD FROM"
                                    :rem 104
5130 POKE214,19:PRINT:POKE211,1:PRINT"THE
     AVERAGE."                       :rem 91
5140 PRINTTAB(9);"{2 DOWN}{RVS}{GRN}PRESS
     ANY KEY FOR MORE{OFF}{CYN}" :rem 50
5150 GET A$:IF A$="" THEN 5150      :rem 185
5155 PRINT"{CLR}"                    :rem 52
5160 POKE214,1:PRINT:POKE211,6:PRINT"SINC
    E THE VALUES YOU ENTER TEND TO"
                                    :rem 140
5180 POKE214,3:PRINT:POKE211,1:PRINT"FORM
     A BELL CURVE (NORMAL DIST.), THE"
                                    :rem 193
```

```
5200 POKE214,5:PRINT:POKE211,1:PRINT"STD.
     DEVIATION IS A MEASURE OF THE AREA"
                                    :rem 84
5220 POKE214,7:PRINT:POKE211,1:PRINT"UNDE
     R THE BELL CURVE."             :rem 79
5230 POKE214,9:PRINT:POKE211,5:PRINT"NO.O
     F STD.DEV.";SPC(6);"% AREA"    :rem 16
5245 POKE214,10:PRINT:POKE211,5:PRINT"---
     ----------";SPC(6);"------":rem 254
5250 PRINTTAB(10);"{DOWN}1";SPC(15);"68.3
     "                              :rem 201
5260 PRINTTAB(10);"{DOWN}2";SPC(15);"95.5
     "                              :rem 205
5270 PRINTTAB(10);"{DOWN}3";SPC(15);"99.7
     "                              :rem 213
5280 PRINTTAB(10);"{DOWN}4";SPC(15);"99.9
     "                              :rem 217
5290 PRINTTAB(9);"{2 DOWN}{RVS}{GRN}PRESS
      ANY KEY FOR MORE{OFF}{CYN}"   :rem 56
5300 GET A$:IF A$="" THEN 5300      :rem 179
5430 PRINT"{CLR}"                   :rem 48
5435 POKE214,3:PRINT:POKE211,5:PRINT"THE
     {SPACE}MEDIAN IS THE VALUE AT THE"
                                    :rem 37
5450 POKE214,5:PRINT:POKE211,1:PRINT"MID-
     POINT OF YOUR DATA."           :rem 222
5490 POKE214,9:PRINT:POKE211,5:PRINT"THE
     {SPACE}RANGE IS THE DIFFERENCE BETWE
     EN"                            :rem 205
5510 POKE214,11:PRINT:POKE211,1:PRINT"YOU
     R LOWEST DATA VALUE AND THE HIGHEST.
     "                              :rem 253
5530 POKE214,13:PRINT:POKE211,1:PRINT"THE
      RANGE IS A QUICK-AND-DIRTY ESTIMATE
     "                              :rem 12
5540 POKE214,15:PRINT:POKE211,1:PRINT"OF
     {SPACE}THE SPREAD. THE STD. DEVIATIO
     N IS"                          :rem 238
5560 POKE214,17:PRINT:POKE211,1:PRINT"MOR
     E RELIABLE."                   :rem 247
5580 PRINTTAB(4);"{2 DOWN}{RVS}{GRN}PRESS
      'S' TO START THE PROGRAM{OFF}{CYN}"
                                    :rem 21
5600 GET A$:IF A$=""OR(A$<>"S")THEN 5600
                                    :rem 33
5610 RETURN                         :rem 172
5810 GETZ$:IFZ$=""OR(Z$<>"C"ANDZ$<>"N"AND
     Z$<>"Q")THEN 5810              :rem 31
5820 FL=0:IFZ$="N"THENFL=1:GOTO6130
                                    :rem 80
5830 IFZ$="Q"THEN587                :rem 130
5840 PRINT"{CLR}"                   :rem 53
5900 REM ERROR CORRECTION SUBROUTINE
                                    :rem 68
6000 POKE214,12:PRINT:POKE211,3:PRINT"REM
     EMBER INCORRECT SAMPLE # (Y/N)?"
                                    :rem 140
6010 GOSUB 7010                     :rem 15
6020 IFZ$="N"THEN6130               :rem 157
6030 PRINT"{CLR}":POKE214,6:PRINT:POKE211
     ,6:INPUT"WHAT IS THE SAMPLE NUMBER";
     EN$                            :rem 241
6040 EN=VAL(EN$):IFEN>NOREN<1OREN<>INT(EN
     )THEN6030                      :rem 131
6070 POKE214,8:PRINT:POKE211,6:PRINT"SAMP
     LE";EN;SPC(5);"VALUE=";SA(EN):rem 75
6080 POKE214,11:PRINT:POKE211,6:INPUT"ENT
     ER YOUR NEW VALUE";C           :rem 8
6090 SA(EN)=C                       :rem 199
6096 POKE214,14:PRINT:POKE211,6:PRINT"ANY
      MORE CHANGES (Y/N)?"          :rem 215
6100 GOSUB 7010                     :rem 15
6110 IF Z$="Y" THEN PRINT"{CLR}":GOTO6000
                                    :rem 123
6120 GOTO587                        :rem 166
6130 PRINT"{CLR}"                   :rem 46
6132 POKE214,21:PRINT:POKE211,0     :rem 65
6134 PRINT"{RVS}{WHT}{2 SPACES}C=CHANGE D
     ATA{2 SPACES}N=NEXT TABLE{2 SPACES}Q
     =QUIT{2 SPACES}{OFF}";         :rem 198
6137 POKE214,2:PRINT:POKE211,5      :rem 26
6140 IFFL=0THENPRINT"THESE ARE THE FIRST
     {SPACE}TEN VALUES.":K=1:GOTO6150
                                    :rem 42
6145 PRINT"THESE ARE THE NEXT TEN VALUES.
     "                              :rem 247
6150 POKE214,6:PRINT:POKE211,10:PRINT"ENT
     RY";SPC(12);"VALUE"            :rem 179
6160 FF=0:FOR K=K TO K+9:FF=FF+1    :rem 247
6165 IFK>300THENK=K+9:NEXT:GOTO587
                                    :rem 115
6170 POKE214,FF+7:PRINT:POKE211,10:PRINTK
     ;TAB(27);SA(K)                 :rem 72
6180 NEXT K                         :rem 89
6190 GOTO5810                       :rem 215
7010 GETZ$:IF Z$=""OR(Z$<>"Y"ANDZ$<>"N")T
     HEN 7010                       :rem 201
7020 RETURN                         :rem 169
```

Program 3: Atari Statistics

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

```
JJ 100 GRAPHICS 0:CLOSE #1:OPEN #1,4,0
       ,"K:":POKE 752,1:SETCOLOR 2,0,0
CE 110 DIM SA(300)
GF 120 REM ■GENERAL INSTRUCTIONS■
DM 130 POSITION 15,6:? "■STATISTICS■":PO
       SITION 18,10:? "■FOR■":POSITION 1
       1,14:? "■NON-STATISTICIANS■"
AG 140 FOR K=1 TO 300:SA(K)=0:NEXT K
IJ 150 ? "{CLEAR}":POSITION 14,3:? "TH
       IS PROGRAM":POSITION 13,4:? "CA
       LCULATES THE":POSITION 12,5:? "
       FOLLOWING VALUES"
FH 160 POSITION 9,6:? "FROM THE DATA Y
       OU INPUT:"
CH 170 POSITION 14,9:? "1. MEAN":POSIT
       ION 14,11:? "2. STANDARD":POSIT
       ION 17,12:? "DEVIATION"
AD 180 POSITION 14,14:? "3. MEDIAN":PO
       SITION 14,16:? "4. RANGE"
LJ 190 POSITION 9,20:? "PRESS 'C' TO C
       ONTINUE"
DF 200 GET #1,A:IF A<>67 THEN 200
HF 210 REM ■REQUEST INSTRUCTIONS■
JC 220 SUM=0:MEAN=0:DFF=0:SDDEV=0:RG=0
EM 230 ? "{CLEAR}":POSITION 5,12:? "Do
        you need instructions (y/n)?"
NF 240 GOSUB 1350
FP 250 IF A=89 THEN GOSUB 830
EI 260 REM ■DATA ENTRY■
FY 270 ? "{CLEAR}":POSITION 9,11:? "En
       ter sample size ";:INPUT N
HJ 280 IF N>300 OR N<=1 THEN FOR I=28
       TO 38:POSITION I,11:? CHR$(32):
       NEXT I:GOTO 270
EM 290 ? "{CLEAR}":POSITION 5,2:? "Ent
       er data one value at a time,
       {4 SPACES}then press return."
EC 300 POSITION 5,6:? "In case of an e
       rror, continue to{3 SPACES}ente
       r data. You will be able to
       {7 SPACES}correct later."
CD 310 FOR I=1 TO N
```

```
HJ 320 POSITION 7,12:? "DATA ENTRY # "
       ;I;:INPUT SA
JF 330 SA(I)=SA
LE 340 FOR J=22 TO 38:POSITION J,12:?
       CHR$(32):NEXT J:NEXT I
GH 350 REM ERROR CORRECTION REQUEST
HM 360 ? "{CLEAR}":POSITION 3,12:? "Wi
       sh to make any corrections (y/n
       )?"
OD 370 GOSUB 1350
EF 380 IF A=89 THEN GOTO 1120
PE 390 REM CALCULATION OF MEAN AND STD
       .DEVIATION
FJ 400 ? "{CLEAR}":POSITION 15,12:? "P
       LEASE WAIT":POSITION 6,14:? "ST
       ATISTICS BEING CALCULATED"
CE 410 FOR I=1 TO N
NI 420 SUM=SUM+SA(I):NEXT I
GH 430 MEAN=SUM/N
CH 440 FOR I=1 TO N
MA 450 DFF=DFF+(SA(I)-MEAN)^2:NEXT I
JA 460 SDDEV=SQR(DFF/(N-1))
JA 470 REM SORTING THE DATA
JL 480 FL=0
IK 490 FOR I=1 TO N-1
BC 500 IF SA(I)<=SA(I+1) THEN 550
FC 510 Q=SA(I)
IM 520 SA(I)=SA(I+1)
LA 530 SA(I+1)=Q
JJ 540 FL=1
CC 550 NEXT I
PF 560 IF FL=1 THEN 480
PB 570 REM CALCULATION OF RANGE
OJ 580 RG=SA(N)-SA(1)
NE 590 LR=SA(1):HR=SA(N)
HM 600 REM CALCULATION OF MEDIAN
AE 610 IF N/2<>INT(N/2) THEN 650
LN 620 IF SA(N/2)=SA(N/2+1) THEN MDD=S
       A(N/2)
MI 630 IF SA(N/2)<>SA(N/2+1) THEN MDD=
       (SA(N/2)+SA(N/2+1))/2
MA 640 GOTO 670
MJ 650 MDD=SA(INT(N/2+1))
KP 660 REM PRINT RESULT TO SCREEN
KA 670 ? "{CLEAR}":POSITION 10,2:? "CA
       LCULATION RESULTS":POSITION 9,4
       :? "**********************"
NN 680 POSITION 4,6:? "SAMPLE SIZE:":P
       OSITION 26,6:? N
FC 690 POSITION 4,8:? "MEAN (X BAR)":P
       OSITION 26,8:? INT(MEAN*10000+0
       .5)/10000
AC 700 POSITION 4,10:? "STD. DEVIATION
       ":POSITION 26,10:? INT(SDDEV*10
       000+0.5)/10000
CP 710 POSITION 4,12:? "MEDIAN":POSITI
       ON 26,12:? MDD
LH 720 POSITION 4,14:? "RANGE":POSITIO
       N 26,14:? RG
HB 730 POSITION 4,16:? "LOWEST SAMPLE
       VALUE":POSITION 26,16:? LR
KA 740 POSITION 4,18:? "HIGHEST SAMPLE
        VALUE":POSITION 26,18:? HR
CA 750 POSITION 13,22:? "PRESS ANY KEY
       "
DO 760 GET #1,A
KL 770 REM REQUEST TO CONTINUE OR END
DO 780 ? "{CLEAR}":POSITION 4,12:? "Wi
       sh to process more data (y/n)?"
OJ 790 GOSUB 1350
NF 800 IF A=78 THEN GRAPHICS 0:END
MG 810 FOR I=1 TO N:SA(I)=0:NEXT I:GOT
       O 220
BA 820 REM INSTRUCTION SUBROUTINE
MW 830 ? "{CLEAR}":POSITION 5,5:? "The
        maximum number of entries is
       {3 SPACES}300, while the minimu
       m number is 2."
IH 840 POSITION 5,9:? "The MEAN is the
        arithmetic average of the numb
       ers you enter."
PB 850 POSITION 5,13:? "The STANDARD D
       EVIATION is a measureof how wid
       ely your numbers spread fromthe
        average."
HB 860 POSITION 10,21:? "Press a key f
       or more"
EA 870 GET #1,A
EL 880 ? "{CLEAR}":POSITION 6,3:? "Sin
       ce the values you enter tend to
        form a bell curve (Normal dist
       .), the Std. Deviation"
ML 890 POSITION 17,5:? " is a measure
       of the{4 SPACES}area under the
       bell curve."
PG 900 POSITION 3,9:? "No of Std.Dev.(
       +/-)":POSITION 3,10:? "--------
       ----------"
NJ 910 POSITION 29,9:? "% Area":POSITI
       ON 29,10:? "-------"
GN 920 POSITION 4,12:? "(+/-) 1 Std.De
       v.":POSITION 31,12:? "68.3"
HF 930 POSITION 4,14:? "(+/-) 2 Std.De
       v.":POSITION 31,14:? "95.5"
IB 940 POSITION 4,16:? "(+/-) 3 Std.De
       v.":POSITION 31,16:? "99.7"
IJ 950 POSITION 4,18:? "(+/-) 4 Std.De
       v.":POSITION 31,18:? "99.9"
HC 960 POSITION 10,21:? "Press a key f
       or more"
EB 970 GET #1,A
LO 980 ? "{CLEAR}":POSITION 4,5:? "The
        MEDIAN is the value at the mid
       - point of your data."
NJ 990 POSITION 4,10:? "The RANGE is t
       he difference between your lowe
       st data value"
LH 1000 POSITION 25,11:? "and the high
        est. The range is a quick-and-
        dirty"
AB 1010 ? "estimate of the spread. The
        standard  deviation is more r
       eliable than the{3 SPACES}spre
       ad."
KD 1020 POSITION 10,20:? "Press 'S' to
        start"
EH 1030 GET #1,A:IF A<>83 THEN 1030
KF 1040 RETURN
CE 1050 REM DISPLAY CORRECTION OPTIONS
BC 1060 ? "{CLEAR}":POSITION 1,23:? "C
       ECHANGE DATA{3 SPACES}N:NEXT T
       ABLE{4 SPACES}Q:QUIT":RETURN
EF 1070 GET #1,A:IF A<>67 AND A<>78 AN
       D A<>81 THEN 1070
DP 1080 IF A=78 THEN 1280
AL 1090 IF A=81 THEN 390
MC 1100 ? "{CLEAR}"
PJ 1110 REM ERROR CORRECTION SUBROUTIN
       E
DF 1120 ? "{CLEAR}":POSITION 3,12:? "R
       emember incorrect sample # (y/
       n)?"
AD 1130 GOSUB 1350
DH 1140 IF A=78 THEN 1230
LN 1150 ? "{CLEAR}":POSITION 7,9:? "Wh
       at is the sample #";:INPUT EN
```

```
IK 1160 IF EN>N OR EN<1 OR EN<>INT(EN)
        THEN 1150
PO 1170 POSITION 7,11:? "Sample ";EN:P
        OSITION 22,11:? "Value ";SA(EN
        )
KN 1180 POSITION 7,13:? "Enter your ne
        w value":POSITION 7,14:INPUT C
        :SA(EN)=C
CE 1190 POSITION 7,19:? "Any more chan
        ges (y/n)?"
AM 1200 GOSUB 1350
DF 1210 IF A=89 THEN 1120
JC 1220 GOTO 400
LH 1230 GOSUB 1060:POSITION 5,2:? "The
        se are the first ten values:"
PM 1240 POSITION 11,5:? "ENTRY":POSITI
        ON 22,5:? "VALUE"
GM 1250 FOR K=1 TO 10
JJ 1260 POSITION 12,K+7:? K:POSITION 2
        4,K+7:? SA(K):NEXT K
ML 1270 GOTO 1070
GH 1280 POSITION 5,2:? "These are the
        next ten values:(DELETE)":IF K
        <=300 THEN GOSUB 1340
BO 1290 CT=8:FOR K=K TO K+9
FA 1300 IF K>300 THEN K=K+9:NEXT K:GOT
        O 400
KF 1310 POSITION 12,CT:? K:POSITION 24
        ,CT:? SA(K)
FB 1320 CT=CT+1:NEXT K
MI 1330 GOTO 1070
JJ 1340 FOR J=1 TO 10:POSITION 12,J+7:
        ? "(3 SPACES)":POSITION 24,J+7
        :? "(15 SPACES)":NEXT J:RETURN
FE 1350 GET #1,A:IF A<>89 AND A<>78 TH
        EN 1350
KK 1360 RETURN
```

Program 4: TI-99/4A Statistics

```
100 DIM SA(300)
110 CALL CLEAR
120 PRINT TAB(10);"STATISTICS"
130 PRINT : : :
140 PRINT TAB(13);"FOR"
150 PRINT : : :
160 PRINT TAB(7);"NON-STATISTICIANS
    "
170 PRINT : : : : : :
180 FOR K=1 TO 400
190 NEXT K
200 CALL CLEAR
210 PRINT "THIS PROGRAM CALCULATES
    THE": :
220 PRINT "FOLLOWING VALUES FROM DA
    TA": :
230 PRINT "YOU INPUT:"
240 PRINT : :
250 PRINT TAB(4);"1. MEAN"
260 PRINT : :
270 PRINT TAB(4);"2. STANDARD DEVIA
    TION"
280 PRINT : :
290 PRINT TAB(4);"3. MEDIAN"
300 PRINT : :
310 PRINT TAB(4);"4. RANGE"
320 PRINT : : :
330 PRINT TAB(2);"PRESS ANY KEY TO
    CONTINUE"
340 PRINT :
350 GOSUB 2170
360 SUM=0
370 MEAN=0
380 DFF=0
390 SDDEV=0
400 RG=0
410 REM INSTRUCTIONS REQUEST
420 PRINT TAB(6);"INSTRUCTIONS (Y/N
    )?"
430 PRINT : : : : : : : : : :
440 GOSUB 2170
450 IF (K<>89)*(K<>78)THEN 440
460 IF K=78 THEN 490
470 GOSUB 1330
480 REM DATA ENTRY
490 CALL CLEAR
500 PRINT TAB(3);"ENTER SAMPLE SIZE
    ";
510 INPUT N
520 IF (N>300)+(N<=1)THEN 490
530 CALL CLEAR
540 PRINT TAB(3);"ENTER YOUR DATA O
    NE VALUE": :
550 PRINT "AT A TIME, THEN PRESS":
    :
560 PRINT "RETURN.": : : :
570 PRINT TAB(3);"IF YOU MAKE AN ER
    ROR,": :
580 PRINT "CONTINUE WITH DATA ENTRY
    .": :
590 PRINT "YOU WILL BE ABLE TO MAKE
    ": :
600 PRINT "CORRECTIONS LATER.": : :
    : :
610 PRINT TAB(2);"PRESS ANY KEY TO
    CONTINUE": :
620 GOSUB 2170
630 FOR I=1 TO N
640 CALL CLEAR
650 PRINT "DATA ENTRY #";I;
660 INPUT R$
670 SA(I)=VAL(R$)
680 NEXT I
690 REM ERROR CORRECTION REQUEST
700 CALL CLEAR
710 PRINT TAB(3);"ANY CORRECTIONS (
    Y/N) ?"
720 PRINT : : : : : : : : : :
730 GOSUB 2170
740 IF K<>89 THEN 770
750 GOSUB 1800
760 REM CALCULATION OF MEAN AND STD
    . DEVIATION
770 PRINT TAB(9);"PLEASE WAIT": : :
780 PRINT "STATISTICS BEING CALCULA
    TED"
790 PRINT : : : : : : : : : :
800 FOR I=1 TO N
810 SUM=SUM+SA(I)
820 NEXT I
830 MEAN=SUM/N
840 FOR I=1 TO N
850 DFF=DFF+(SA(I)-MEAN)^2
860 NEXT I
870 SDDEV=SQR(DFF/(N-1))
880 REM SORT OF DATA INTO NUMERIC O
    RDER
890 FL=0
900 FOR I=1 TO N-1
910 IF SA(I)<=SA(I+1)THEN 960
920 Q=SA(I)
930 SA(I)=SA(I+1)
940 SA(I+1)=Q
950 FL=1
```

```
960 NEXT I
970 IF FL=1 THEN 890
980 REM CALCULATION OF RANGE
990 RG=SA(N)-SA(1)
1000 LR=SA(1)
1010 HR=SA(N)
1020 REM CALCULATION OF MEDIAN
1030 IF N/2<>INT(N/2)THEN 1090
1040 IF SA(N/2)<>SA(N/2+1)THEN 1060
1050 MDD=SA(N/2)
1060 IF SA(N/2)=SA(N/2+1)THEN 1080
1070 MDD=(SA(N/2)+SA(N/2+1))/2
1080 GOTO 1110
1090 MDD=SA(INT(N/2+1))
1100 REM PRINT RESULTS TO SCREEN
1110 CALL CLEAR
1120 PRINT TAB(5);"CALCULATION RESU
     LTS": :
1130 PRINT "**********************
     *****": : :
1140 PRINT "SAMPLE SIZE";TAB(19);N:
      :
1150 PRINT "MEAN (X BAR)";TAB(19);I
     NT(MEAN*10000+.5)/10000: :
1160 PRINT "STD. DEVIATION";TAB(19)
     ;INT(SDDEV*10000+.5)/10000: :
1170 PRINT "MEDIAN";TAB(19);INT(MDD
     *10000+.5)/10000: :
1180 PRINT "RANGE";TAB(19);INT(RG*1
     0000+.5)/10000: :
1190 PRINT "LOWEST VALUE";TAB(19);L
     R: :
1200 PRINT "HIGHEST VALUE";TAB(19);
     HR: : :
1210 PRINT TAB(8);"PRESS ANY KEY"
1220 GOSUB 2170
1230 REM REQUEST TO CONTINUE OR END
1240 PRINT " WISH TO PROCESS MORE D
     ATA": :
1250 PRINT TAB(12);"(Y/N)?": : : :
     : : : : : :

1260 GOSUB 2170
1270 IF K=78 THEN 1320
1280 FOR I=1 TO N
1290 SA(I)=0
1300 NEXT I
1310 GOTO 360
1320 END
1330 PRINT TAB(3):"THE MAXIMUM NUMB
     ER OF EN-": :
1340 PRINT "TRIES YOU CAN MAKE IS 3
     00.": :
1350 PRINT "THE MINIMUM NUMBER IS 2
      ": : :
1360 PRINT TAB(3);"THE MEAN IS THE
     ARITH ": :
1370 PRINT "METIC AVERAGE OF THE NU
     MBERS": :
1380 PRINT "YOU ENTER.": : :
1390 PRINT TAB(3);"STANDARD DEVIATI
     ON IS A": :
1400 PRINT "MEASURE OF HOW WIDELY Y
     OUR": :
1410 PRINT "NUMBERS SPREAD FROM THE
     ": :
1420 PRINT "AVERAGE.": : :
1430 GOSUB 2160
1440 CALL CLEAR
1450 PRINT TAB(3);"SINCE THE VALUES
      YOU ENTER": :
1460 PRINT "TEND TO FORM A BELL CUR
     VE": :
1470 PRINT "(NORMAL DISTRIBUTION)
     THE": :
1480 PRINT "STD. DEVIATION IS A MEA
     SURE": :
1490 PRINT "OF THE AREA UNDER THE B
     ELL": :
1500 PRINT "CURVE.": : :
1510 PRINT TAB(4);"NO. OF STD.
     {4 SPACES}% AREA"
1520 PRINT TAB(5);"DEV.(+/-)"
1530 PRINT TAB(4);"-----------
     {4 SPACES}------": :
1540 PRINT TAB(8);"1{11 SPACES}68.3"
1550 PRINT TAB(8);"2{11 SPACES}95.5"
1560 PRINT TAB(8);"3{11 SPACES}99.7"
1570 PRINT TAB(8);"4{11 SPACES}99.9"
     : : :
1580 GOSUB 2160
1590 PRINT TAB(3);"THE MEDIAN IS TH
     E VALUE AT": :
1600 PRINT "THE MID-POINT OF YOUR D
     ATA.": : :
1610 PRINT TAB(3);"THE RANGE IS THE
      DIF-": :
1620 PRINT "FERENCE BETWEEN YOUR LO
     WEST": :
1630 PRINT "DATA VALUE AND THE HIGH
     EST.": :
1640 PRINT "IT IS A QUICK-AND-DIRTY
     ": :
1650 PRINT "ESTIMATE OF THE SPREAD.
     ": :
1660 PRINT "STANDARD DEVIATION IS M
     ORE": :
1670 PRINT "RELIABLE, HOWEVER.": :
     : :
1680 PRINT TAB(3);"PRESS ANY KEY TO
      START"
1690 GOSUB 2170
1700 RETURN
1710 REM DISPLAY CORRECTION OPTION
1720 GOSUB 2170
1730 IF (K<>67)*(K<>78)*(K<>81)THEN
      1720
1740 FL=0
1750 IF K<>78 THEN 1780
1760 FL=1
1770 GOTO 1980
1780 IF K=81 THEN 770
1790 REM ERROR CORRECTION SUBR
1800 PRINT "REMEMBER INCORRECT SAMP
     LE #": :
1810 PRINT TAB(11);"(Y/N) ?": : : :
      : : : : : :
1820 GOSUB 2170
1830 IF K=78 THEN 1980
1840 INPUT "WHAT IS THE SAMPLE # ?
     ":EN$
1850 EN=VAL(EN$)
1860 IF (EN>N)+(EN<1)+(EN<>INT(EN))
     THEN 1840
1870 PRINT : :
1880 PRINT "SAMPLE";EN;"{3 SPACES}"
     ;"VALUE=";SA(EN)
1890 PRINT : :
1900 PRINT "ENTER YOUR NEW VALUE :
     "
1910 INPUT SA(EN)
```

```
1920 PRINT : : : : : :
1930 PRINT TAB(3);"ANY MORE CHANGES
     (Y/N)?": : : : :
1940 GOSUB 2170
1950 CALL CLEAR
1960 IF K=78 THEN 770
1970 GOTO 1800
1980 IF FL=1 THEN 2020
1990 PRINT "THESE ARE THE FIRST TEN
     ": :
2000 L=1
2010 GOTO 2040
2020 CALL CLEAR
2030 PRINT "THESE ARE THE NEXT TEN"
     : :
2040 PRINT "VALUES.": : :
2050 PRINT TAB(5);"ENTRY";TAB(15);"
     VALUE": :
2060 FF=0
2070 FOR L=L TO L+9
2080 FF=FF+1
2090 IF L>300 THEN 770
2100 PRINT TAB(5);L;TAB(15);SA(L)
2110 NEXT L
2120 PRINT : :
2130 PRINT "C=CHANGE DATA
     (3 SPACES)N=NEXT TABLE": :
2140 PRINT TAB(12);"Q=QUIT"
2150 GOTO 1720
2160 PRINT TAB(3);"PRESS ANY KEY FO
     R MORE";
2170 CALL KEY(0,K,S)
2180 IF S=0 THEN 2170
2190 CALL CLEAR
2200 RETURN
```

Program 5: PC/PCjr Statistics

```
100 DIM SA(300)
110 REM GENERAL INTRODUCTION
120 WIDTH 40
130 KEY OFF
140 CLS:LOCATE 9,15,0:PRINT "STATISTICS"

150 LOCATE 11,18:PRINT "FOR"
160 LOCATE 13,12:PRINT "NON-STATISTICIAN
S"
170 FOR K=1 TO 1500:NEXT K
180 CLS:LOCATE 5,8:PRINT"This program ca
lculates the":LOCATE 7,5:PRINT"following
 from data you input:"
190 LOCATE 10,10:PRINT "1. Mean":LOCATE
13,10:PRINT"2. Standard Deviation"
200 LOCATE 16,10:PRINT "3. Median":LOCAT
E 19,10:PRINT"4. Range"
210 GOSUB 1190
220 REM INSTRUCTION REQUEST
230 SUM=0:MEAN=0:DFF=0:SDDEV=0:RG=0
240 LOCATE 13,4 :PRINT "Do you need inst
ructions (Y/N) ?"
250 GOSUB 1160
260 IF A$="Y" OR A$="y" THEN GOSUB 730
270 REM DATA ENTRY
280 LOCATE 11,9:INPUT "Enter sample size
";N
290 IF N>300 OR N<=1 THEN 280
300 CLS:LOCATE 3,4:PRINT"If you make an
error, continue with":LOCATE 5,4:PRINT"t
he with data entry. You will be"
310 LOCATE 7,4:PRINT"able to correct it
later."
320 FOR I=1 TO N:LOCATE 13,22:PRINT STRI
NG$(19,32):LOCATE 13,10:PRINT"Data entry
 #    "STRING$(4,29)I;:INPUT R$
330 SA(I)=VAL(R$):NEXT I
340 REM ERROR CORRECTION REQUEST
350 CLS:LOCATE 12,2:PRINT"Wish to make a
ny corrections (Y/N)?"
360 GOSUB 1160
370 IF A$="Y" OR A$="y" THEN 950
380 REM CALCULATION OF MEAN AND STD. DEV
IATION
390 LOCATE 11,14:PRINT"PLEASE WAIT"
400 LOCATE 13,6:COLOR 0,7:PRINT"Statisti
cs being calculated":COLOR 7,0
410 FOR I=1 TO N:SUM=SUM+SA(I):NEXT I
420 MEAN=SUM/N
430 FOR I=1 TO N:DFF=DFF+(SA(I)-MEAN)^2:
NEXT I
440 SDDEV=SQR(DFF/(N-1))
450 REM SORT OF DATA INTO NUMERIC ORDER
460 FL=0:FOR I=1 TO N-1:IF SA(I)<=SA(I+1
) .THEN 480
470 Q=SA(I):SA(I)=SA(I+1):SA(I+1)=Q:FL=1

480 NEXT I
490 IF FL=1 THEN 460
500 REM CALCULATION OF RANGE
510 RG=SA(N)-SA(1):LR=SA(1):HR=SA(N)
520 REM CALCULATION OF MEDIAN
530 IF N/2<>INT(N/2) THEN 570
540 IF SA(N/2)=SA(N/2+1) THEN MDD=SA(N/2
)
550 IF SA(N/2)<>SA(N/2+1) THEN MDD=(SA(N
/2)+SA(N/2+1))/2
560 GOTO 590
570 MDD=SA(INT(N/2+1))
580 REM PRINT RESULTS TO THE SCREEN
590 CLS:LOCATE 3,10:PRINT"CALCULATION RE
SULTS"
600 LOCATE 5,4:PRINT"**********************
***************"
610 LOCATE 7,7:PRINT"Sample Size";SPC(10
);N
620 LOCATE 9,7:PRINT"Mean (X Bar)";SPC(9
);INT(MEAN*10000+.5)/10000
630 LOCATE 11,7:PRINT"Std. Dev.";SPC(12)
;INT(SDDEV*10000+.5)/10000
640 LOCATE 13,7:PRINT"Median";SPC(15);MD
D
650 LOCATE 15,7:PRINT"Range";SPC(16);RG
660 LOCATE 17,7:PRINT"Lowest Sample Valu
e";SPC(2);LR
670 LOCATE 19,7:PRINT"Highest Sample Val
ue";SPC(1);HR:GOSUB 1190
680 REM REQUEST TO CONTINUE OR END
690 LOCATE 12,3:PRINT"Wish to process mo
re data (Y/N) ?":GOSUB 1160
700 IF A$="N" OR A$="n" THEN CLS:END
710 FOR I=1 TO N:SA(I)=0:NEXT I:CLS:GOTO
 230
720 REM INSTRUCTION SUBR
730 LOCATE 3,5:PRINT "The maximum number
 of entries you"
740 LOCATE 5,2:PRINT"can make is 300. Th
e minimum is 2."
750 LOCATE 9,5:PRINT"The mean is the ari
thmetic average":LOCATE 11,2:PRINT"of th
e numbers you enter."
760 LOCATE 15,5:PRINT"Standard deviation
 is a measure":LOCATE 17,2:PRINT"of how.
```

```
widely your numbers spread from"
770 LOCATE 19,2:PRINT"the average."
780 GOSUB 1170
790 CLS:LOCATE 1,5:PRINT"Since the value
s you enter tend to":LOCATE 3,2:PRINT"fo
rm a bell curve (normal"
800 LOCATE 5,2:PRINT"distribution), the
standard deviation":LOCATE 7,2:PRINT"is
a measure of the area under the bell"
810 LOCATE 9,2:PRINT"curve.":LOCATE 11,9
:PRINT"No. of Std.     % Area"
820 LOCATE 13,10:PRINT"Dev. (+/-)":LOCATE
 14,9:PRINT"-----------    ------"
830 LOCATE 16,14:PRINT"1"SPC(10)"68.3":L
OCATE 18,14:PRINT"2"SPC(10)"95.5":LOCATE
 20,14:PRINT"3"SPC(10)"99.7":LOCATE 22,1
4:PRINT"4"SPC(10)"99.9"
840 GOSUB 1170
850 CLS:LOCATE 4,5:PRINT"The median is t
he value at the mid-":LOCATE 6,2:PRINT"p
oint of your data."
860 LOCATE 9,5:PRINT"The range is the di
fference between":LOCATE 11,2:PRINT"your
 lowest data value and the highest."
870 LOCATE 13,2:PRINT"The range is a qui
ck-and-dirty estimate":LOCATE 15,2:PRINT
"of the spread. The standard deviation"
880 LOCATE 17,2:PRINT"is a more reliable
 estimate of spread":LOCATE 19,2:PRINT"t
han range."
890 LOCATE 23,9:PRINT"Press any key to s
tart":GOSUB 1180:RETURN
900 REM DISPLAY CORRECTION OPTIONS
910 Z$=INKEY$:IF Z$="" OR (Z$<>"c" AND Z
$<>"C" AND Z$<>"n" AND Z$<>"N" AND Z$<>"
q" AND Z$<>"Q") THEN 910
920 CLS:FL=0:IF Z$="n" OR Z$="N" THEN FL
=1:GOTO 1060
930 IF Z$="q" OR Z$="Q" THEN 390
940 REM ERROR CORRECTION SUBR
950 LOCATE 12,3:PRINT"Remember incorrect
 sample # (Y/N) ?":GOSUB 1160
960 IF A$="N" OR A$="n" THEN 1060
970 LOCATE 7,6:INPUT "What is the sample
 number ";EN$
980 EN=VAL(EN$):IF EN>N OR EN<1 OR EN<>I
NT(EN) THEN 970
990 LOCATE 9,6:PRINT"Sample";EN;SPC(5);"
Value=";SA(EN)
1000 LOCATE 12,6:INPUT"Enter your new va
lue";C
1010 SA(EN)=C
1020 LOCATE 16,6:PRINT"Any more changes
(Y/N) ?"
1030 GOSUB 1160
1040 IF A$="Y" OR A$="y" THEN 950
1050 GOTO 390
1060 CLS:COLOR 0,7:LOCATE 22,2:PRINT"  C
=CHANGE DATA  N=NEXT TABLE  Q=QUIT  ":CO
LOR 7,0
1070 LOCATE 2,5
1080 IF FL=0 THEN PRINT"These are the fi
rst ten values.":K=1:GOTO 1100
1090 PRINT"These are the next ten values
."
1100 LOCATE 6,10:PRINT"Entry";SPC(12);"v
alue"
1110 FF=0:FOR K=K TO K+9:FF=FF+1
1120 IF K>300 THEN CLS:GOTO 390
1130 LOCATE FF+7,10:PRINT K;TAB(27);SA(K
)
1140 NEXT K
1150 GOTO 910
1160 A$=INKEY$:IF A$="" OR (A$<>"Y" AND
A$<>"y" AND A$<>"N" AND A$<>"n") THEN 11
60 ELSE CLS:RETURN
1170 LOCATE 24,10:PRINT"Press any key fo
r more";
1180 A$=INKEY$:IF A$="" THEN 1180 ELSE C
LS:RETURN
1190 LOCATE 22,7: PRINT "Press any key t
o continue"
1200 IF INKEY$="" THEN 1200 ELSE CLS:RET
URN
```

Program 6: Apple Statistics

```
100  HOME : DIM SA(300)
110  INVERSE : VTAB 9: HTAB 17: PRINT
     "STATISTICS": VTAB 13: HTAB 20: PRINT
     "FOR"
120  VTAB 17: HTAB 13: PRINT "NON-STAT
     ISTICIANS"
130  FOR K = 1 TO 300:SA(K) = 0: NEXT
     K: NORMAL
140  HOME : VTAB 2: PRINT "THIS PROGRA
     M CALCULATES THE FOLLOWING:"
150  VTAB 6: HTAB 15: PRINT "1. MEAN":
      VTAB 10: HTAB 15: PRINT "2. STAN
     DARD": VTAB 11: HTAB 18: PRINT "D
     EVIATION"
160  VTAB 15: HTAB 15: PRINT "3. MEDIA
     N": VTAB 19: HTAB 15: PRINT "4. R
     ANGE"
170  VTAB 23: HTAB 10: PRINT "PRESS 'C
     ' TO CONTINUE": VTAB 23: HTAB 17
180  GET Z$: IF Z$ <  > "C" THEN 180
190  REM  INSTRUCTIONS REQUEST
200 SUM = 0:MEAN = 0:DFF = 0:SDDEV = 0
     :RG = 0
210  HOME : VTAB 12: HTAB 5: PRINT "DO
      YOU NEED INSTRUCTIONS (Y/N)?": VTAB
     12: HTAB 35
220  GOSUB 1180
230  IF Z$ = "Y" THEN  GOSUB 700
240  HOME : REM  DATA ENTRY
250  VTAB 12: HTAB 9: INPUT "ENTER SAM
     PLE SIZE :";N$:N =  VAL (N$)
260  IF N > 300 OR N <  = 1 THEN  FOR
     I = 26 TO 35: VTAB 12: HTAB I: PRINT
      CHR$ (32): NEXT : GOTO 250
270  HOME : PRINT "IN CASE OF AN ERROR
     , CONTINUE WITH THE  ENTRY."
280  VTAB 2: HTAB 8: PRINT "YOU CAN CO
     RRECT LATER."
290  FOR I = 1 TO N: VTAB 12: HTAB 10:
      PRINT "DATA ENTRY #";I;: INPUT R
     $:SA(I) =  VAL (R$)
300  FOR J = 24 TO 39: VTAB 12: HTAB J
     : PRINT  CHR$ (32): NEXT J: NEXT
     I
310  REM  ERROR CORRECTION REQUEST
320  HOME : VTAB 12: PRINT "WISH TO MA
     KE ANY CORRECTIONS (Y/N)?": VTAB
     12: HTAB 35
330  GOSUB 1180
340  IF Z$ = "Y" THEN  GOTO 970
350  REM  CALCULATIONS OF MEAN AND STD
     . DEVIATION
360  HOME : VTAB 11: HTAB 16: PRINT "P
     LEASE WAIT"
370  VTAB 13: HTAB 8: PRINT "STATISTIC
     S BEING CALCULATED"
380  FOR I = 1 TO N:SUM = SUM + SA(I):
      NEXT I
```

```
390 MEAN = SUM / N
400  FOR I = 1 TO N:DFF = DFF + (SA(I)
     - MEAN) ^ 2: NEXT I
410 SDDEV =  SQR (DFF / (N - 1))
420  REM  SORTING THE DATA
430 FL = 0: FOR I = 1 TO N - 1: IF SA(
     I) <  = SA(I + 1) THEN 450
440 Q = SA(I):SA(I) = SA(I + 1):SA(I +
     1) = Q:FL = 1
450  NEXT I: IF FL = 1 THEN 430
460  REM  CALCULATION OF RANGE
470 HR = SA(N):LR = SA(1):RG = HR - LR

480  REM  CALCULATION OF MEDIAN
490  IF N / 2 <  > INT (N / 2) THEN 5
     20
500  IF SA(N / 2) = SA(N / 2 + 1) THEN
     MDD = SA(N / 2): GOTO 540
510 MDD = (SA(N / 2) + SA(N / 2 + 1)) /
     2: GOTO 540
520 MDD = SA( INT (N / 2 + 1))
530  REM  PRINT RESULTS TO SCREEN
540  HOME : VTAB 2: HTAB 10: PRINT "CA
     LCULATION RESULTS": VTAB 3: HTAB
     9: PRINT "**********************"
550  VTAB 6: HTAB 6: PRINT "SAMPLE SIZ
     E"; TAB( 28);N
560  VTAB 8: HTAB 6: PRINT "MEAN (X-BA
     R)"; TAB( 28); INT (MEAN * 10000 +
     .5) / 10000
570  VTAB 10: HTAB 6: PRINT "STD. DEVI
     ATION"; TAB( 28); INT (SDDEV * 10
     000 + .5) / 10000
580  VTAB 12: HTAB 6: PRINT "MEDIAN"; TAB(
     28);MDD
590  VTAB 14: HTAB 6: PRINT "RANGE"; TAB(
     28);RG
600  VTAB 16: HTAB 6: PRINT "LOWEST SA
     MPLE VALUE"; TAB( 28);LR
610  VTAB 18: HTAB 6: PRINT "HIGHEST S
     AMPLE VALUE"; TAB( 28);HR
620  VTAB 23: HTAB 10: PRINT "PRESS 'P
     ' TO PROCEED": VTAB 23: HTAB 17
630  GET Z$: IF Z$ = "" OR Z$ <  > "P"
      THEN 630
640  REM  REQUEST TO CONTINUE OR END
650  HOME : VTAB 12: HTAB 3: PRINT "WI
     SH TO PROCESS MORE DATA (Y/N)?": VTAB
     12: HTAB 34
660  GOSUB 1180
670  IF Z$ = "N" THEN  HOME : END
680  FOR I = 1 TO N:SA(I) = 0: NEXT I:
      GOTO 200
690  REM  INSTRUCTIONS SUBROUTINE
700  HOME : VTAB 4: PRINT "THE MAXIMUM
      SAMPLE SIZE IS 300, WHILE   THE
     MINIMUM IS 2."
710  VTAB 10: PRINT "THE MEAN IS THE A
     RITHMETIC AVERAGE OF   THE NUMBER
     S YOU ENTER."
720  VTAB 16: PRINT "THE STANDARD DEVI
     ATION IS A MEASURE OF  HOW WIDELY
      YOUR DATA SPREADS FROM THE   AVE
     RAGE."
730  VTAB 22: HTAB 12: PRINT "PRESS 'M
     ' FOR MORE": VTAB 22: HTAB 19
740  GET Z$: IF Z$ = "" OR Z$ <  > "M"
      THEN 740
750  HOME : PRINT "SINCE THE VALUES YO
     U ENTER TEND TO FORM A BELL CURVE
      (NORMAL DIST.), THE STD.   DEVIA
     TION IS A MEASURE OF THE AREA UND
     ERTHE BELL CURVE."
760  VTAB 8: PRINT "NO. OF STD.DEV. (+
     /-)": VTAB 9: PRINT "------------
     ----------"
770  VTAB 8: HTAB 28: PRINT "% AREA": VTAB
     9: HTAB 28: PRINT "--------"
780  VTAB 11: HTAB 2: PRINT "(+/-) 1 S
     TD.DEV."; TAB( 30);"68.3"
790  VTAB 13: HTAB 2: PRINT "(+/-) 2 S
     TD.DEV."; TAB( 30);"95.5"
800  VTAB 15: HTAB 2: PRINT "(+/-) 3 S
     TD.DEV."; TAB( 30);"99.7"
810  VTAB 17: HTAB 2: PRINT "(+/-) 4 S
     TD.DEV."; TAB( 30);"99.9"
820  VTAB 22: HTAB 12: PRINT "PRESS 'M
     ' FOR MORE": VTAB 22: HTAB 19
830  GET Z$: IF Z$ = "" OR Z$ <  > "M"
      THEN 830
840  HOME : VTAB 4: PRINT "THE MEDIAN
     IS THE VALUE AT THE MID-POINT OF
     YOUR DATA."
850  VTAB 8: PRINT "THE RANGE IS THE D
     IFFERENCE BETWEEN YOURLOWEST DATA
      VALUE AND THE HIGHEST. THE  RANG
     E IS A QUICK-AND -DIRTY ESTIMATE
     OF THE SPREAD."
860  VTAB 11: HTAB 13: PRINT "THE STD.
      DEVIATION IS MORE  RELIABLE."
870  VTAB 22: HTAB 5: PRINT "PRESS 'S'
      TO START THE PROGRAM": VTAB 22: HTAB
     12
880  GET Z$: IF Z$ = "" OR Z$ <  > "S"
      THEN 880
890  RETURN
900  REM  DISPLAY CORRECTION OPTIONS
910  HOME : INVERSE : VTAB 24: PRINT "
     C=CHANGE DATA      N=NEXT TABLE
      E=EXIT": NORMAL : RETURN
920  GET Z$: IF Z$ = "" OR (Z$ <  > "C
     " AND Z$ <  > "N" AND Z$ <  > "E"
     ) THEN 920
930  IF Z$ = "N" THEN 1120
940  IF Z$ = "E" THEN 350
950  GOTO 970
960  REM  ERROR CORRECTION SUBROUTINE
970  HOME : VTAB 12: HTAB 3: PRINT "RE
     MEMBER INCORRECT SAMPLE # (Y/N)?"
     : VTAB 12: HTAB 36
980  GOSUB 1180
990  IF Z$ = "N" THEN 1080
1000  HOME : VTAB 8: HTAB 6: PRINT "WH
     AT IS THE SAMPLE NUMBER";: INPUT
     E$:EN =  VAL (E$)
1010  IF EN > N OR EN < 1 OR EN <  >  INT
     (EN) THEN 1000
1020  VTAB 11: HTAB 6: PRINT "SAMPLE "
     ;EN; TAB( 22);"VALUE "SA(EN)
1030  VTAB 15: HTAB 6: PRINT "ENTER YO
     UR NEW VALUE";: INPUT C:SA(EN) =
     C
1040  VTAB 22: HTAB 6: PRINT "ANY MORE
      CHANGES (Y/N)?": VTAB 22: HTAB 2
     8
1050  GOSUB 1180
1060  IF Z$ = "Y" THEN 970
1070  GOTO 360
1080  GOSUB 910: VTAB 2: PRINT "THESE
     ARE THE FIRST TEN SAMPLE VALUES:"

1090  VTAB 6: HTAB 10: PRINT "ENTRY"; TAB(
     25);"VALUE"
1100  VTAB 7: HTAB 10: PRINT "-----"; TAB(
     25);"-----"
1110  FOR K = 1 TO 10: VTAB K + 8: HTAB
```

```
    11: PRINT K; TAB( 26);SA(K): NEXT
    K: GOTO 920
1120  VTAB 2: PRINT "THESE ARE THE NEX
    T TEN SAMPLE VALUES:": VTAB 2: HTAB
    38: PRINT " "
1130  IF K < 300 THEN  GOSUB 1170
1140 L = 9: FOR K = K TO K + 9: IF K >
    300 THEN K = K + 9: NEXT K: GOTO
    360
1150  VTAB L: HTAB 10: PRINT K; TAB( 2
    6);SA(K)
1160 L = L + 1: NEXT K: GOTO 920
1170  FOR M = 9 TO 18: VTAB M: HTAB 10
    : PRINT "    "; TAB( 26);"
                    ": NEXT M: RETURN
1180  GET Z$: IF Z$ = "" OR (Z$ <  > "
    Y" AND Z$ <  > "N") THEN 220
1190  RETURN
```

Program 7: Color Computer Statistics

```
100 DIM SA(300)
110 REM GENERAL INTRODUCTION
120 CLS:PRINT@ 171,"STATISTICS":PRI
    NT@ 238,"FOR":PRINT@ 295,"NON-S
    TATISTICIANS"
130 FOR K=1 TO 1000:NEXT K
140 CLS:PRINT@ 32,"THIS PROGRAM CAL
    CULATES THE(5 SPACES)FOLLOWING:
    "
150 PRINT@ 139,"1. MEAN":PRINT@ 203
    ,"2. STANDARD":PRINT@ 238,"DEVI
    ATION"
160 PRINT@ 299,"3. MEDIAN":PRINT@ 3
    63,"4. RANGE"
170 PRINT@ 454,"PRESS 'C' TO CONTIN
    UE"
180 Z$=INKEY$:IF Z$="" OR Z$ <>"C"
    THEN 180
190 SUM=0:MEAN=0:DFF=0:SDDEV=0:RG=0
200 CLS:PRINT@ 224,"DO YOU NEED INS
    TRUCTIONS (Y/N)?"
210 GOSUB 1050:IF Z$="Y" THEN GOSUB
     600
220 REM DATA ENTRY
230 CLS:PRINT@ 229,"ENTER SAMPLE SI
    ZE";:INPUT N$:N=VAL(N$)
240 IF N>300 OR N<=1 THEN  230
250 CLS:PRINT"  IN CASE OF AN ERROR
     CONTINUE. YOU CAN CORRECT LATE
    R."
260 FOR I=1 TO N:PRINT@ 224,"DATA E
    NTRY #";I;:INPUT R$:SA(I)=VAL(R
    $)
270 FOR J=242 TO 250:PRINT@ J,CHR$(
    32):NEXT J:NEXT I
280 REM ERROR CORRECTION REQUEST
290 CLS:PRINT@ 224,"WISH TO MAKE CO
    RRECTIONS (Y/N)?"
300 GOSUB 1050:IF Z$="Y" THEN GOSUB
     870
310 REM CALCULATIONS OF MEAN AND ST
    D. DEVIATION
320 CLS:PRINT@ 202,"PLEASE WAIT":PR
    INT@ 258,"STATISTICS BEING CALC
    ULATED"
330 FOR I=1 TO N:SUM=SUM+SA(I):NEXT
     I:MEAN=SUM/N
340 FOR I=1 TO N:DFF=DFF+(SA(I)-MEA
    N)^2:NEXT I:SDDEV=SQR(DFF/(N-1)
    )
350 REM SORTING THE DATA
360 FL=0:FOR I=1 TO N-1:IF SA(I)<=S
    A(I+1) THEN 380
370 Q=SA(I):SA(I)=SA(I+1):SA(I+1)=Q
    :FL=1
380 NEXT I:IF FL=1 THEN 360
390 REM CALCULATION OF RANGE
400 HR=SA(N):LR=SA(1):RG=HR-LR
410 REM CALCULATION OF MEDIAN
420 IF N/2 <> INT(N/2) THEN 450
430 IF SA(N/2)=SA(N/2+1) THEN MDD=S
    A(N/2):GOTO 470
440 MDD=(SA(N/2)+SA(N/2+1))/2:GOTO
    470
450 MDD=SA(INT(N/2+1))
460 REM PRINT RESULTS TO SCREEN
470 CLS:PRINT@6,"CALCULATION RESULT
    S":PRINT@37,"**********
    ***"
480 PRINT@ 100,"SAMPLE SIZE";TAB(20
    );N
490 PRINT@164,"MEAN (X-BAR)";TAB(20
    );INT(MEAN*10000+.5)/10000
500 PRINT@228,"STD.DEVIATION";TAB(2
    0);INT(SDDEV*10000+.5)/10000
510 PRINT@292,"MEDIAN";TAB(20);MDD
520 PRINT@356,"RANGE";TAB(20);RG:PR
    INT@397,"(";LR;"-";HR;")"
530 PRINT@456,"PRESS ANY KEY"
540 Z$=INKEY$:IF Z$=""THEN 540
550 REM REQUEST TO CONTINUE OR END
560 CLS:PRINT@224,"WISH TO PROCESS
    MORE DATA (Y/N)?"
570 GOSUB 1050:IF Z$="N" THEN CLS:E
    ND
580 FOR I=1 TO N:SA(I)=0:NEXT I:GOT
    O 190
590 REM INSTRUCTIONS SUBROUTINE
600 CLS:PRINT@65,"THE MAXIMUM NUMBE
    R OF ENTRIES  IS 300. THE MINIM
    UM IS 2."
610 PRINT@193,"THE MEAN IS THE ARIT
    HMETIC(5 SPACES)AVERAGE OF THE
    DATA."
620 PRINT@322,"THE STANDARD DEVIATI
    ON IS A(4 SPACES)MEASURE OF HOW
     WIDELY YOUR DATA SPREADS FROM
    THE AVERAGE."
630 PRINT@453,"PRESS ANY KEY FOR MO
    RE"
640 Z$=INKEY$:IF Z$=""THEN 640
650 CLS:PRINT" SINCE THE VALUES YOU
     ENTER TENDTO FORM A BELL CURVE
     (NORMAL(4 SPACES)DIST.), THE S
    TD.DEVIATION IS A  MEASURE OF A
    REA UNDER THE BELL  CURVE."
660 PRINT@192,"NO. OF STD.DEV. (+/-
    )";TAB(25);"% AREA"
670 PRINT@224,"----------------------
    -";TAB(25);"-------"
680 PRINT@290,"(+/-) 1 STD.DEV.";TA
    B(26);"68.3"
690 PRINT@322,"(+/-) 2 STD.DEV.";TA
    B(26);"95.5"
700 PRINT@354,"(+/-) 3 STD.DEV.";TA
    B(26);"99.7"
710 PRINT@386,"(+/-) 4 STD.DEV.";TA
    B(26);"99.9"
720 PRINT@453,"PRESS ANY KEY FOR MO
    RE"
730 Z$=INKEY$:IF Z$="" THEN 730
```

```
740 CLS:PRINT:PRINT:PRINT" THE MEDI
    AN IS THE VALUE AT THE MID-POIN
    T OF YOUR DATA."
750 PRINT@ 193,"THE RANGE IS THE DI
    FFERENCE(4 SPACES)BETWEEN YOUR
    LOWEST DATA VALUE  AND THE HIGH
    EST. IT IS A QUICK- AND-DIRTY E
    STIMATE OF THE"
760 PRINT@320,"SPREAD. STD. DEVIATI
    ON IS MORE  RELIABLE THAN RANGE
    ."
770 PRINT@449,"PRESS 'S' TO START T
    HE PROGRAM"
780 Z$=INKEY$:IF Z$="" OR (Z$<>"S")
     THEN 780
790 RETURN
800 REM DISPLAY CORRECTION OPTION
810 CLS:PRINT@480,"C=CORRECT
    (3 SPACES)N=NEXT TABLE  E=EXIT"
    ;:RETURN
820 Z$=INKEY$:IF Z$="" OR (Z$<>"C"
    AND Z$<>"N" AND Z$<>"E") THEN 8
    20
830 IF Z$="N" THEN 990
840 IF Z$="E" THEN 310
850 GOTO 870
860 REM ERROR CORRECTION SUBROUTINE
    "
870 CLS:PRINT@224,"REMEMBER INCORRE
    CT SAMPLE (Y/N)?"
880 GOSUB 1050:IF Z$="N" THEN 960
890 CLS:PRINT@165,"WHAT IS THE SAMP
    LE #";:INPUT E$:EN=VAL(E$)
900 IF EN>N OR EN<1 OR EN<>INT(EN)
    THEN 890
910 PRINT@229,"SAMPLE ";EN;TAB(20);
    "VALUE ";SA(EN)
920 PRINT@293,"ENTER YOUR NEW VALUE
    ":PRINT"(5 SPACES)";:INPUT C$:S
    A(EN)=VAL(C$)
930 PRINT@453,"ANY MORE CHANGES (Y/
    N)?"
940 GOSUB 1050:IF Z$="Y" THEN 870
950 GOTO 1040
960 GOSUB 810:PRINT@0,"THESE ARE TH
    E FIRST TEN VALUES:"
970 PRINT@69,"ENTRY";TAB(17);"VALUE
    "
980 I=1:FOR K=101 TO 389 STEP 32:PR
    INT@K,I;TAB(17);SA(I):I=I+1:NEX
    T K:GOTO 820
990 PRINT@0,"THESE ARE THE NEXT TEN
     VALUES":IF I<300 THEN GOSUB 10
    30
1000 FOR K=101 TO 389 STEP 32:IFI>3
    00 THEN 1020
1010 PRINT@K,I;TAB(17);SA(I):I=I+1
1020 NEXT K:GOTO 820
1030 FOR K=101 TO 389 STEP 32:PRINT
    @K,"(3 SPACES)";TAB(18);"
    (6 SPACES)":NEXT K:RETURN
1040 RETURN
1050 Z$=INKEY$:IF Z$="" OR (Z$<>"Y"
     AND Z$<>"N") THEN 1050
1060 RETURN
```

M'FILE For The Commodore 64

Stan Fenster

M'FILE is a set of data management programs for the Commodore 64. A 1541 disk drive and a printer are required. The manual indicates that there's also a tape version, but it was not tested for this review. Any serial port or user port (RS-232 or parallel) printer can be used.

The package contains a 7-inch x 9-inch loose-leaf manual, a software key, and a program disk. The programs on the disk can be copied to a backup disk. The key, which plugs into Control Port 1 (front joystick port), is required to run the program. The disk contains the main routines, a SORT program, a COPY program, and a TEXT'MERGE program. No data files go on this disk. Instead, each data file occupies a separate data disk. This might discourage the use of *M'FILE* for small jobs.

The manual is well written and generally easy to follow, though more examples would have been helpful. However, the menus within the program lead the user so effectively through all the necessary steps that you'd rarely need to refer to the manual.

Major Characteristics

Data is stored on the data disk, not in RAM. This allows for either 1000 records of 125 characters each or 500 records of 250 characters each. Each record is limited to 32 fields of no more than 78 characters each. Fields can be either numeric or alphanumeric. All alphabetic characters are in uppercase, but a new version which supports both upper- and lowercase has been announced. Numeric fields can be used in formulas allowing arithmetic calculations among fields. Within formulas, an if-then syntax is available. Up to ten user-defined interfield formulas can be used.

Selection of records for updates, reports, or text merges can be made using any or all fields. One field is designated as the keyfield. Searches using the keyfield are especially fast, and offer extensive comparison capabilities (=, <, >, <=, >=, or <>) between fields.

Report formats are very flexible, and up to 15 such formats can be saved for reuse later. Fields can be printed in columns or horizontally, and special ASCII control codes can be sent to the printer.

Main Menu Options

The main menu gives the following options:

1. Access a File.
2. Print a Report.
3. Create a File.
4. Sort a File.
5. Exit.

To get started, select Create a File. Next, format a data disk and set it up to receive the record specifications. After a

title is entered, the program leads the user through the steps of choosing field names, types, sizes, and decimal settings. When setup is complete, it is stored on disk, and the user proceeds to Access a File.

Access a File is used to enter new data, to change or inspect data, or to add fields and formulas.

Print a Report allows the user to create a new report format, to call up a saved report format, to add temporary fields or formulas, to display totals on the screen, or to set printer control characters. After a report format is chosen and edited as desired, it may be saved. The report is then printed. Many options are available in setting up the print format.

Use Of Function Keys

The screens displayed by *M'FILE* are either menus or data screens. Data screens have a line at the bottom indicating the functions available by use of function keys. These generally include f1=ACCEPT, f3=DELETE, and f5=EXIT. On a data entry or modification screen, f7=ADD/SUB (modify a numeric field) also appears.

When a record selection is made by record number, the f2 key advances the pointer to the next sequential record, and f4 moves to the previous sequential record. Note that if records have been deleted, there may be missing record numbers. When a record selection is done by searching, the f8 key advances to the next matching record. However, none of these even-numbered function keys is indicated in the screen prompts.

Sorting

Sorting is done by a separate program (SORT) loaded from the main menu. Sorting is performed on any single field, in either ascending or descending order. The result is a new, sorted data file on the disk. The exit from SORT is to the BASIC environment, not back to the main menu. To continue processing—for example, to prepare a report—the user must load and run *M'FILE* again.

Text Merging

A separate program, TEXT'MERGE, is provided to create files which can be used by most major word processing programs. The selection menu lists Quick Brown Fox, WordPro, EasyScript, Script 64, Totl Text, Busiwriter, Paper Clip, and Other. Other is intended to work with any word processors which can read sequential files. I found that I could not generate mailing lists and labels in the format I wanted with the *M'FILE* report generator. However, using TEXT'MERGE in conjunction with my word processor (EasyScript), it was easy to get single-column labels. Additional columns are not supported by *M'FILE* but I was able to do it with word processor manipulations—but only because my printer has reverse-line motion.

Copy Program

A separate program, COPY, is provided to make backup copies of *M'FILE* data disks, using a single disk drive. The file title of the new disk can be different from the old title. As with other backup programs, frequent disk swaps are required to make the backup disk. COPY also allows the user to copy the format only. This feature is useful for expanding the file onto a second disk, or creating a new file with the same format.

Other Features And Considerations

Another capability of *M'FILE* is a screen dump to the printer at almost any time, by pressing the British pound key.

When the file structure is accepted by the user, the file character limit is set at 125 if no more than 125 characters are required; otherwise, the limit is 250 characters. After this, the original field names and structures can't be changed, although new fields can be added later, as long as the maximum character limit (125 or 250) is not exceeded.

In report mode, additional temporary numeric fields can be added. Their contents are created by using formulas involving other numeric fields. The menu allows the user to add alphanumeric fields, but there is no way to enter data into them.

Reports have an optional total line at the end. However, the number of characters for each numeric field is the same for the totals as for individual entries. This leads to possible truncation in a total if extra digits would have been generated. I had this problem, and I found a solution by creating a temporary field equal to my original field, but with extra width. Then I used the new field instead of the old one and got a correct total. Of course, it would have been easier if the original field had been wider, but I didn't know that at the time. A related feature is Screen Totals. This allows display of totals on the screen, but has the same width limitation as printed totals.

Because the main data is kept on the data disk, there are frequent disk reads and writes. In particular, when an exit is made from the file access mode after changes are made, the data file on disk is rewritten, which takes a few minutes. When the user enters a new mode, a read is required, even if a write was just performed.

Field selection and searching are easy and fast. Sorting on keyfields is fast too, but limited in speed by disk accesses. Other sorts are slower.

Even with the limitations discussed above, *M'FILE* is a

powerful and versatile data management system.

M'FILE
by m'soft
Double E. Electronics, Inc.
12027 Pacific Street
Omaha, NE 68154
(402) 334-7870
$79.95

AtariWriter

John Heilborn

A few years ago, Atari introduced a program called *Atari Word Processor*. It was a fairly expensive word processor that had an enormous number of functions and features. It was versatile and almost totally menu-driven. In other words, while you were using it, the computer displayed a menu of program functions at the top of the screen.

Unfortunately, if you decided you wanted to do something that was on another, undisplayed, menu you would have to know how to find the menu (the menus were nested and could be accessed by pressing different options) and would have to go through all of the other menus to get there.

The old word processor was very versatile, but was also rather cumbersome.

A New Generation

Today there is a new Atari word processor called *AtariWriter*. It's obviously a descendant of the original *Atari Word Processor*, but all of the "fat" has been trimmed. *AtariWriter* is easy to use (no menus to wallow through), easy to understand (all of the commands are logically accessed), and very responsive (when you press a key, the computer responds quickly).

One additional feature that makes this program far better than the old one is that it is in a cartridge, not on disk, and it will work with as little as 16K memory. You can use it with *any* of the existing Atari computers and

you don't need to have a disk drive or a tape drive. Of course, if you run the word processor without a device to store your text, you will lose it after writing it. However, for people just using the system as an enhanced typewriter, this is enough.

Functions And Features

Although I am used to writing on an expensive professional word processing system, *AtariWriter* has all of the features that I would normally use on the larger system. I wrote this article on *AtariWriter* and found, in fact, that *AtariWriter* has some very nice extra features not usually found on other systems, such as a single toggle function which allows you to switch displayed characters from upper- to lowercase automatically. The program has more features than I can cover in a review, so I'll just give you the highlights.

When you turn *AtariWriter* on, it displays the Atari logo for a few moments and then switches to a menu of functions. The functions are:

- **[C]REATE** File is used to begin writing a new file. If you have some text in memory left over from another file and you select this option, the computer responds by asking you if you wish to delete the file in memory. This way you will not accidentally erase a file that you want to SAVE. If there is no file in memory, the computer simply goes to the editing page.
- **[D]ELETE** File erases a file from the disk drive. When you select this option, the computer asks for the name of the file you wish to delete. Once you have selected the file to delete, it checks with you one more time by asking, "ARE YOU SURE?" This makes it almost impossible to erase a file by accident.
- **[E]DIT** File is similar to [C]REATE File, but is used to continue working on an existing file.
- **[F]ORMAT** Disk erases all of the information a disk contains, so the program asks you, "ARE YOU SURE?" before executing this command too.
- **[I]NDEX** of Disk Files displays the names of all the programs and text files on your disk. After the files have been displayed, the computer asks if you want to print the index. If you press Y or enter YES (and have a printer connected), the index will be printed.
- **[L]OAD** File transfers any file you have on disk (or cassette) into the computer's memory.
- **[P]RINT** File prints the file that is currently in memory. This routine will not print a file directly from disk (or cassette). The file *must* first be transferred to memory.
- **[S]AVE** File transfers any file you have in memory onto the disk (or cassette).

Editing

In the editing mode, *AtariWriter* displays a blank screen with a

series of letters and numbers across the top. Below the blank screen is a black area with arrows indicating the tab positions and two indicators (L and C) which are used to keep track of the line and column of the cursor location.

Print Formatting Controls

The letters and numbers across the top of the screen are print controls and can be changed at any time during your editing session. This means that you can have text that varies in width, spacing, or any other parameter that can be set with these controls. The print functions that you can control are bottom margin, paragraph spacing, print style, paragraph indention, right justification, left margin position, right margin position, line spacing, top margin, and page length.

Block Functions

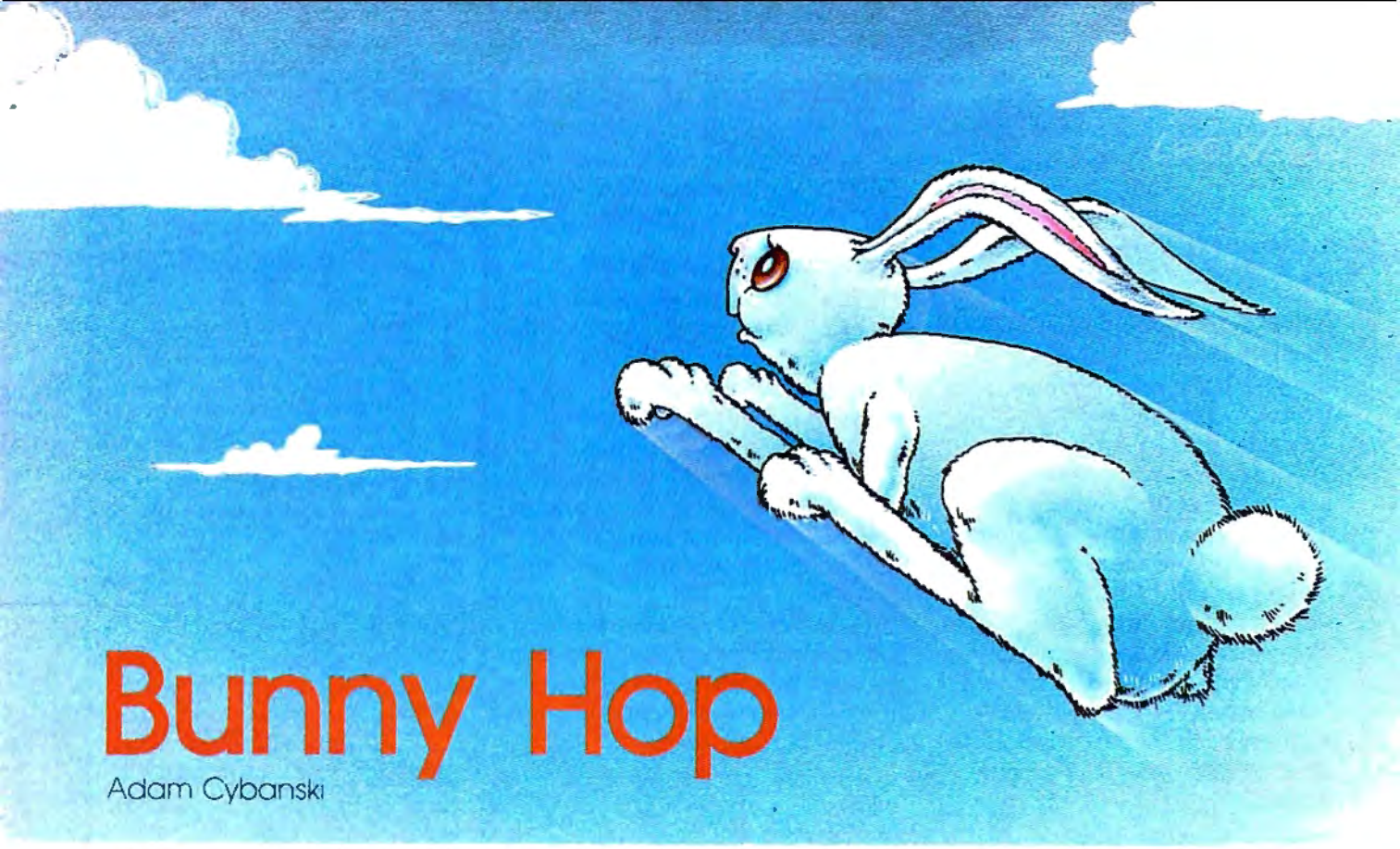
Block functions are controls that allow you to move or delete entire blocks of text. To move a block of text, you would simply mark the beginning and end of the block you wish to move (or delete); *AtariWriter* will do the rest for you automatically.

Search and Replace

With search and replace you can specify a word (or several words) that you want the computer to find. *AtariWriter* will then look through the entire document and locate each occurrence of the word (or words) you specify. Once each word has been found, you can continue editing from that point, replace that word or delete it.

AtariWriter is a very good, low-cost word processing system that can provide you with virtually every feature you could want from a word processor.

AtariWriter
Atari, Inc.
1265 Borregas Ave.
Box 427
Sunnyvale, CA 94086
(408) 745-2000
$99.95



Bunny Hop

Adam Cybanski

Airborne carrots and a cloud-hopping rabbit add a dash of whimsy to this joystick-controlled game for the Commodore 64, VIC, and Atari.

It's harvest time in that great big carrot patch in the sky and Copernicus, a rather hungry rabbit, is on the prowl for his favorite vegetable.

Unfortunately, due to a fluke of nature as well as incredibly bad air pollution, the carrots are suspended in midair. This, however, can be turned into an advantage for Copernicus, since he is able to hop around on the dense clouds in pursuit of his meal.

He may also jump around on platforms which have been dispersed across the sky for absolutely no reason at all.

Watch That Brier Patch

The only thing Copernicus cannot do is land in the brier patch below—he would be transported back to his initial position. This would not be so bad if he had an unlimited number of transport passes—unfortunately, he has not. After they are exhausted, he is condemned to the brier patch for the rest of his life.

If he clears the sky of carrots, then benevolent forces disperse a new set of carrots for him, and his friends in the brier patch become so happy for him that they usually jump around in a rabbit euphoria. Meanwhile, Copernicus continues to collect carrots until he runs out of transport passes.

One Tricky Carrot

Once in a while our hero eats a jumping bean which has been disguised as a carrot by some unknown troublemaker. This causes him to hop continuously, hampering his ability to gather carrots. In addition, the more he eats, the heavier he becomes, until he finds that he can no longer jump as high as he used to. But then, it just adds more challenge to his hare-raising adventures.

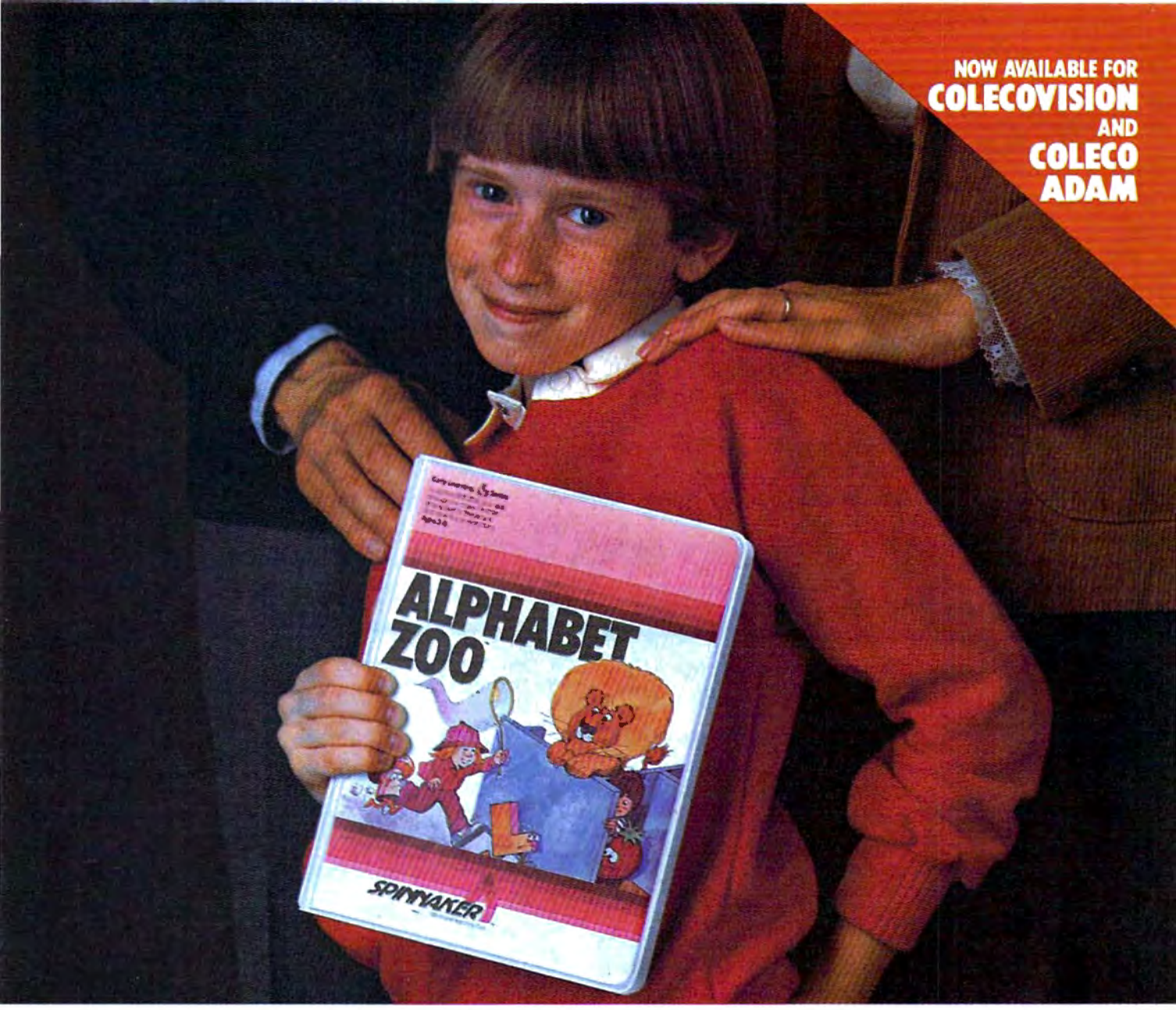
Joystick Controls

The object of the game is to collect all the carrots. Copernicus jumps when you press the fire button or push the joystick forward. He moves either left or right when you push the joystick in either of those directions.

Clearing the screen of carrots starts you on a new level. Every fifth level is a "jumper round" during which the rabbit jumps continuously. At the completion of every tenth level, a new rabbit is earned. Rabbits are displayed at the top center of the screen, and one is lost every time you fall into the brier patch at the bottom of the screen. When all rabbits are exhausted, the game is over.

You have 60 seconds to collect the carrots. If your time runs out, the clouds speed up and you forfeit an extra time bonus. If you still do not collect all the carrots within 30 seconds, the clouds and platforms will start disappearing.

The score is displayed on the top right. Try to beat a score of 10,000! The key to high scoring is to clear the carrots quickly and collect the extra time bonus.



Finally, computer games you want your kids to play.

Spinnaker makes computer games kids love to play.

But some of our biggest fans are parents.

Because on top of all the fun and excitement, our games have something more. True educational value. They help develop a child's learning skills, in all kinds of fun ways.

So Spinnaker games aren't just computer games. They're Learning Games.

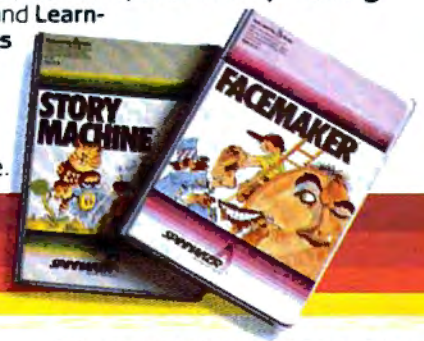
They're written by top educators who know how to make learning fun. And by expert game programmers, who use colorful graphics, animation and sound to make our games so exciting, your kids may not even realize they're learning. They're having too good a time!

That's why children love us. And parents love us.

And why we're already the leader in the field of home educational software.

So if you're looking for computer games that you'll like as much as your kids will, look for Spinnaker **Early Learning Games** (ages 3–8) and **Learning Discovery Games** (ages 6–12) at your local retailer.

Spinnaker. We're giving computer games a good name.



Disks for: Apple,® Atari,® IBM® PC and PCjr and Commodore 64.™

Cartridges for: ColecoVision,® Coleco Adam,™ Atari,® IBM® PCjr and Commodre 64.™

SPINNAKER™

We make learning fun.

Logic (VIC And 64 Versions)

Lines 5 to 20 set the custom characters. Lines 30 to 55 produce a short machine language routine to move the clouds. Lines 100 to 150 initialize the screen for each new level.

Lines 200 to 280 are the main game loop. Lines 400 to 410 are a subroutine which increases the game difficulty when the timer reaches zero. Lines 500 to 520 are called when the game ends.

Lines 600 to 635 control the interlevel features like the jumper round and game speed. Lines 700 and 705 are called when a carrot is gathered.

Lines 800 to 840 are a routine which is called when all carrots have been gathered. It controls the extra time bonus and the bouncing rabbit.

Lines 900 and 905 are called when a rabbit hits the brier patch.

Bunny Hop (VIC And 64 Versions)

Line 20 contains the variable I, which is the general speed of the game. Decreasing it makes the game faster but also causes Copernicus to remain on the screen for less time.

Line 25 has CT, which is the number of carrots you must collect initially. LI is one less than the number of rabbits you start off with. L is the number of cloud pieces that are placed in each row.

The platform pieces in line 115 and also in line 117 of the 64 version are character number 183.

Line 150 has the variable D, which contains the initial bunny character. It is either 33, facing right, or 34, facing left. Line 150 also increases CT by one. Changing this to CT=CT+0 will keep the number of carrots to be collected at four.

Use Either Joystick Port

The statement in line 200 of the 64 version, Y=PEEK(56320)AND PEEK(QQ) combines the joystick ports, enabling use of either.

Changing the number 36 to the number 1 in line 245 will enable a person to jump on the brier patch without losing a bunny.

Z of line 400 is the cloud speed switch. If Z is zero, the clouds remain slow. They speed up when Z is changed to −1.

Line 610 increments LE, which is the level counter. If LE is greater than 19, W is set to one instead of zero, which makes Copernicus' jumps shorter.

An Extra Rabbit

The tens in line 615 give you an extra rabbit every ten levels. Changing this to INT(LE/5)=LE/5 will give extra rabbits every five levels.

Line 620 makes V true (−1) every five levels, which produces a jumper round.

Line 625 resets the number of carrots to 4 every ten levels.

Line 630 decreases the number of cloud pieces per row. Changing it to L=L−0 will keep the number of clouds constant.

20 Points For A Carrot

You are given 20 points for a carrot in line 700. Changing this to FORX=1TO55 would give you 55 points for each carrot.

If you are tired of the large number of bunny hops at high levels, just revise line 800 with FORC=1TO2, which will perform only two hops each level throughout the game.

To change the number of extra bonus points, simply revise line 830 to something like SC=SC+2.

Program 1: 64 Bunny Hop

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

```
1 QQ=56321                                :rem 17
5 DATA0,0,0,0,0,0,0,0,8,12,6,189,127,124,
  56,70,16,48,96,189,254,62,28,34:rem 241
10 DATA0,102,255,255,102,0,0,0,66,40,134,
   51,136,102,102,102                     :rem 78
15 DATA0,48,188,255,255,188,48,0,0,102,25
   5,255,102,0,0,0                        :rem 228
16 POKE56334,0:POKE1,51                   :rem 35
17 FORX=0TO1023:POKEX+13312,PEEK(X+53248)
   :NEXT:POKE1,55:POKE56334,1             :rem 2
20 FORX=12544TO12599:READY:POKEX,Y:NEXT:S
   =54272:I=75                            :rem 243
23 POKES+5,9:POKES+6,9:POKES,115:POKES+1,
   3:POKE53280,5:POKE53281,6              :rem 190
25 POKE53272,28:K=54272:L=18:CT=3:LI=3:GO
   SUB600:POKES+24,15                     :rem 208
30 DATA160,40,177,253,200,145,253,136,136
   ,192,255,208,245,160,40,177,253,160
                                          :rem 207
31 DATA 0,145,253,160                     :rem 253
35 DATA40,169,32,145,253,96,160,41,177,25
   3,136,145,253,200,200,192,81,2:rem 216
40 DATA245,160,40,177,253,160,80,145,253,
   160,40,169,32,145,253,96,169,5,133,254
                                          :rem 110
45 DATA169,63,133,253,32,96,47,169,184,13
   3,253,32,68,47,169,47,133,253,169,6
                                          :rem 254
50 DATA133,254,32,96,47,169,168,133,253,3
   2,68,47,169,31,133,253,169,7,133,254
                                           :rem 32
53 DATA32,96,47,96                        :rem 180
55 FORX=12100TO12203:READY:POKEX,Y:NEXT
                                          :rem 227
100 PRINT"{HOME}{15 RIGHT}{WHT}{RVS}BUNNY
    {2 SPACES}HOP":PRINT"{WHT}{RVS}
    {2 RIGHT}TIME":PRINT"{7 LEFT}{RVS}SCO
    RE"                                    :rem 33
105 FORX=1944TO2023:POKEX,36:POKEX+K,5:NE
    XT                                    :rem 181
110 FORX=55496TO56215:POKEX,1:NEXT:I=I-3
                                          :rem 193
115 PRINT"{HOME}{7 DOWN}{YEL}{RVS}[Y]
```

IS YOUR CHILD TOP BANANA, OR JUST ONE OF THE BUNCH?

Kids everywhere are going ape over Artworx® Monkey Series educational software! Like all good arcade games, kids just can't stop playing them. Which is great, because while they're enjoying the antics of Marc the Monkey, they're learning. And growing.

Three Artworx monkey programs, designed by teachers and learning specialists, are available to help your child.

Monkeymath

Monkeymath™ uses colorful graphics and three levels of challenges to give a better understanding of number sequences, addition, subtraction, multiplication, and division.

Monkeynews™ uses a newspaper setting to increase your child's ability to read and understand by enabling him to actively participate in the story, answer questions, check facts and type his own headlines.


Monkeynews

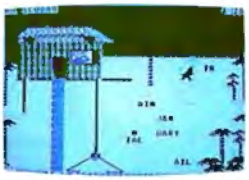
For help with spelling and vocabulary, choose *Monkeybuilder*™. It encourages the child to combine word pieces correctly to form building blocks, and make a tree house for Marc.

All three programs are more fun than a barrel of you-know-whats!

All are available for the Commodore 64, Atari, and Apple computers. All include a FREE Marc the Monkey story and coloring book. And they're all the stuff top bananas are made of!

See them in action wherever software is sold. To find out more facts, send $1.00 for a colorful catalog to: Artworx Software Co., Inc. 150 North Main Street, Fairport, NY 14450.

Or call: 800-828-6573.
(In New York call: 716-425-2833).


Monkeybuilder

Monkeymath
by Dennis Zander
$24.95

Monkeynews
by Dennis Zander
$29.95

Monkeybuilder
by Dean Kindig and Rob Fitter
$29.95

Artworx®
Educational Software with a smile.

The bunny is about to fall into the brier patch (64 version of "Bunny Hop").

```
    {DOWN}{2 LEFT}{Y}{5 DOWN}{2 Y}{DOWN}
    {4 LEFT}{2 Y}{5 DOWN}{2 Y}{DOWN}
    {4 LEFT}{2 Y}"                  :rem 199
117 PRINT"{HOME}{RVS}{7 DOWN}{11 RIGHT}
    {3 Y}{12 RIGHT}{3 Y}{6 DOWN}{2 LEFT}
    {Y}{16 LEFT}{Y}"                :rem 201
120 FORX=9TO21STEP6:FORY=1TOL:POKE1024+X*
    40+RND(1)*40,35:NEXTY,X         :rem 225
125 FORX=11TO17STEP6:FORY=1TOL:POKE1024+X
    *40+RND(1)*40,38:NEXTY,X        :rem 23
130 C=CT:FORX=1TOC                  :rem 120
135 Y=INT(RND(1)*639)+1224:IFPEEK(Y)<>32O
    RY=1505THEN135                  :rem 210
140 IF(Y<1905ANDY>1862)OR(Y<1746ANDY>1702
    )OR(Y<1666ANDY>1622)THEN135     :rem 248
145 IF(Y<1425ANDY>1382)OR(Y<1506ANDY>1462
    )THEN135                        :rem 101
150 POKEY,37:POKEY+K,7:NEXTX:P=1505:D=33:
    CT=CT+1                         :rem 212
155 TI$="000000":POKES-1,130        :rem 215
200 POKEP,32:POKE37154,127:Y=PEEK(56320)A
    NDPEEK(QQ):IF(YAND8)=0THENP=P+1:D=3
                                    :rem 71
205 IFPEEK(P+40)=38AND(NOT(B)ORZ)THENP=P+
    1                               :rem 46
210 IF(YAND4)=0THENP=P-1:D=34       :rem 204
215 POKES+4,32:IFPEEK(P+40)=35AND(NOT(B)O
    RZ)THENP=P-1                    :rem 218
220 B=(B=0):IFBORZTHENSYS12156:POKES+4,33
                                    :rem 243
225 IFA=0ANDPEEK(P+40)<>32THENIF(YAND16)=
    0OR(YAND1)=0ORVTHENA=7-W*2:POKES+11,3
    3                               :rem 62
230 IFA<4-WANDPEEK(P+40)=32ORPEEK(P+40)=3
    7THENP=P+40                     :rem 91
233 POKES+1,3:IFZTHENPOKES+1,15     :rem 151
235 IFA<>0THENA=A-1:IFA>3-WTHENP=P-40:POK
    ES+8,PEEK(S+8)+2                :rem 208
240 IFPEEK(P)=37THEN700             :rem 88
245 IFPEEK(P+40)=36THEN900          :rem 237
250 IFPEEK(P)<>32THENP=P+40:GOTO260:POKES
    ,PEEK(S)-20                     :rem 177
255 POKES,70                        :rem 177
260 IFPEEK(P)=37THEN700             :rem 90
265 POKEP,D:POKEP+K,1:FORX=1TOI:NEXT
                                    :rem 100
270 PRINT"{HOME}{3 DOWN}{8 LEFT}{RVS}
    {WHT}";SC                       :rem 128
275 PRINT"{HOME}{2 DOWN}{RVS}{2 RIGHT}";:
    T=60-VAL(TI$):IFT<1THEN400      :rem 203
280 PRINTT;"{LEFT} ":GOTO200        :rem 156
400 PRINT"{HOME}{2 DOWN}{4 SPACES}{WHT}
    {RVS}0{2 SPACES}":Z=-1:IFTI$<"000130"
    THENGOTO200                     :rem 56
405 Y=1224+INT(RND(1)*719):IFPEEK(Y)=37OR
    (PEEK(Y)=32ANDRND(1)>.2)THEN405
                                    :rem 209
410 POKEY,32:GOTO200                :rem 179
500 PRINT"{12 DOWN}{16 RIGHT}{RVS}{CYN}AG
    AIN ?"                          :rem 82
505 GETA$:IFA$=""THEN505            :rem 87
510 IFA$="Y"THENRUN                 :rem 136
515 IFA$="N"THENEND                 :rem 100
520 GOTO505                         :rem 106
600 IFLI<0THEN500                   :rem 234
605 PRINT"{CLR}{2 DOWN}{18 RIGHT}{WHT}";:
    IFLI>0THENFORX=1TOLI:PRINT"!";:NEXT
                                    :rem 43
610 PRINT" ":LE=LE+1:Z=0:W=0:V=0:IFLE>19T
    HENW=1                          :rem 79
615 IFINT(LE/10)=LE/10THENLI=LI+1:rem 216
620 IFLE/5=INT(LE/5)THENV=-1        :rem 170
625 IFLE/10=INT(LE/10)THENCT=4      :rem 30
630 L=L-1:IFL<3THENL=10             :rem 105
635 RETURN                          :rem 126
700 FORX=1TO20:POKES+4,129:SC=SC+1:PRINT"
    {HOME}{3 DOWN}{8 LEFT}{RVS}";SC:POKES
    +4,128:NEXT                     :rem 162
705 C=C-1:ON-(C=0)+1GOTO265,800     :rem 184
800 POKES,0:FORC=1TOLE:Y=0:X=0:V=0
                                    :rem 120
805 Z=1904+X+Y:X=X+1:POKES+4,33:IFX=39THE
    N825                            :rem 209
810 POKES+1,ABS(Y/20)+10:IFV=0THENY=Y-40:
    IFY=-160THENV=1                 :rem 145
815 IFV=1THENY=Y+40:IFY=0THENV=0     :rem 31
820 POKEZ,32:POKE1904+X+Y,33:POKE1904+X+Y
    +K,1:POKES+4,32:GOTO805         :rem 81
825 POKEZ,32:NEXTC:POKES+1,3        :rem 236
830 IFT>0THENFORX=TTO0STEP-1:POKES+4,129:
    PRINT"{HOME}{2 DOWN}{RVS}{2 RIGHT}";X
    :SC=SC+T                        :rem 124
835 IFT>0THENPRINT"{HOME}{3 DOWN}{7 LEFT}
    {RVS}";SC:POKES+4,128:NEXTX     :rem 24
840 POKES,0:GOSUB600:GOTO100        :rem 206
900 FORY=1TOLE:POKES+1,10:FORX=1TO150STEP
    5:POKES+4,33:POKES+1,X/10       :rem 138
905 POKES+4,32:NEXTX,Y:POKES+1,3:LI=LI-1:
    GOSUB600:GOTO100                :rem 48
```

Program 2: VIC Bunny Hop

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

```
5 DATA0,0,0,0,0,0,0,0,8,12,6,189,127,124,
  56,70,16,48,96,189,254,62,28,34:rem 241
10 DATA0,102,255,255,102,0,0,0,66,40,134,
   51,136,102,102,102            :rem 78
15 DATA0,48,188,255,255,188,48,0,0,102,25
   5,255,102,0,0,0               :rem 228
20 FORX=7424TO7479:READY:POKEX,Y:NEXT:S=3
   6876:I=100                    :rem 196
25 POKE36869,255:POKES+3,109:K=30720:L=10
   :CT=4:LI=3:GOSUB600:POKES+2,15:rem 172
30 DATA160,22,177,253,200,145,253,136,136
   ,192,255,208,245,160,22,177,253,160,0,
   145                           :rem 241
```

```
35 DATA253,160,22,169,32,145,253,96,160,2
   3,177,253,136,145,253,200,200,192,45,2
   08                                :rem 201
40 DATA245,160,22,177,253,160,44,145,253,
   160,22,169,32,145,253,96,169,30,133,25
   4,169                             :rem 104
45 DATA153,133,253,32,184,28,169,220,133,
   253,32,156,28,169,29,133,253,169,31,13
   3                                 :rem 164
50 DATA254,32,184,28,169,96,133,253,32,15
   6,28,169,161,133,253,32,184,28,96
                                     :rem 141
55 FORX=7324TO7423:READY:POKEX,Y:NEXT
                                     :rem 151
100 PRINT"{HOME}{6 SPACES}{YEL}{RVS}BUNNY
    {2 SPACES}HOP":PRINT"{WHT}{RVS}TIME":
    PRINT"{5 LEFT}{RVS}SCORE"        :rem 147
105 FORX=8164TO8185:POKEX,36:POKEX+K,5:NE
    XT                               :rem 197
110 FORX=38576TO38861:POKEX,1:NEXT:I=I-3
                                     :rem 200
115 PRINT"{HOME}{6 DOWN}{YEL}{RVS}{Y}
    {9 RIGHT}{2 Y}{9 RIGHT}{Y}{5 DOWN}
    {2 Y}{4 LEFT}{DOWN}{2 Y}{5 DOWN}{2 Y}
    {4 LEFT}{DOWN}{2 Y}"             :rem 227
120 FORX=8TO20STEP6:FORY=1TOL:POKE7680+X*
    22+RND(1)*22,35:NEXTY,X          :rem 237
125 FORX=10TO16STEP6:FORY=1TOL:POKE7680+X
    *22+RND(1)*22,38:NEXTY,X          :rem 35
130 C=CT:FORX=1TOC                   :rem 120
135 Y=INT(RND(1)*351)+7768:IFPEEK(Y)<>320
    RY=8055THEN135                   :rem 227
140 IF(Y<8143ANDY>8118)OR(Y<8056ANDY>8030
    )OR(Y<8011ANDY>7986)THEN135       :rem 6
145 IF(Y<7879ANDY>7854)OR(Y<7924ANDY>7898
    )THEN135                         :rem 159
150 POKEY,37:POKEY+K,7:NEXTX:P=8055:D=33:
    CT=CT+1                          :rem 219
155 TI$="000000":POKES-1,130         :rem 215
200 POKEP,32:POKE37154,127:Y=PEEK(37152):
    IF(YAND128)=0THENP=P+1:D=33      :rem 244
205 IFPEEK(P+22)=38AND(NOT(B)ORZ)THENP=P+
    1                                 :rem 46
210 POKE37154,255:Y=PEEK(37137):IF(YAND16
    )=0THENP=P-1:D=34                :rem 127
215 IFPEEK(P+22)=35AND(NOT(B)ORZ)THENP=P-
    1                                 :rem 46
220 B=(B=0):IFBORZTHENSYS7380:POKES-1,PEE
    K(S-1)+20:IFPEEK(S-1)=170THENPOKES-1,
    130                              :rem 167
225 IFA=0ANDPEEK(P+22)<>32THENIF(YAND32)=
    0OR(YAND4)=0ORVTHENA=7-W*2:POKES,195
                                     :rem 235
230 IFA<4-WANDPEEK(P+22)=32ORPEEK(P+22)=3
    7THENP=P+22                      :rem 91
235 IFA<>0THENA=A-1:IFA>3-WTHENP=P-22:POK
    ES,PEEK(S)+20                    :rem 58
240 IFPEEK(P)=37THEN700              :rem 88
245 IFPEEK(P+22)=36THEN900          :rem 237
250 IFPEEK(P)<>32THENP=P+22:POKES,PEEK(S)
    -20:GOTO260                      :rem 177
255 POKES,70                         :rem 177
260 IFPEEK(P)=37THEN700               :rem 90
265 POKEP,D:POKEP+K,1:FORX=1TOI:NEXT
                                     :rem 100
270 PRINT"{HOME}{3 DOWN}{6 LEFT}{RVS}";SC
                                      :rem 65
275 PRINT"{HOME}{2 DOWN}{RVS}";:T=60-VAL(
    TI$):IFT<1THEN400                :rem 145
280 PRINTT;"{LEFT} ":GOTO200         :rem 156
400 PRINT"{HOME}{2 DOWN}{2 SPACES}{CYN}
    {RVS}0{2 SPACES}":Z=-1:IFTI$<"000130"
    THENGOTO200                      :rem 210
405 Y=7746+INT(RND(1)*417):IFPEEK(Y)=37OR
    (PEEK(Y)=32ANDRND(1)>.5)THEN405
                                     :rem 222
410 POKEY,32:GOTO200                 :rem 179
500 PRINT"{10 DOWN}{7 RIGHT}{RVS}{GRN}AGA
    IN ?"                            :rem 170
505 GETA$:IFA$=""THEN505              :rem 87
510 IFA$="Y"THENRUN                  :rem 136
515 IFA$="N"THENEND                  :rem 100
520 GOTO505                          :rem 106
600 IFLI<0THEN500                    :rem 234
605 PRINT"{CLR}{2 DOWN}{9 RIGHT}{WHT}";:I
    FLI>0THENFORX=1TOLI:PRINT"!";:NEXT
                                     :rem 38
610 PRINT" ":LE=LE+1:Z=0:W=0:V=0:IFLE>19T
    HENW=1                           :rem 79
615 IFINT(LE/10)=LE/10THENLI=LI+1:rem 216
620 IFLE/5=INT(LE/5)THENV=-1         :rem 170
625 IFLE/10=INT(LE/10)THENCT=4        :rem 30
630 L=L-1:IFL<3THENL=10              :rem 105
635 RETURN                           :rem 126
700 FORX=1TO20:POKES+1,126+(20-X)*3:SC=SC
    +1:PRINT"{HOME}{3 DOWN}{6 LEFT}{RVS}"
    ;SC:NEXT                         :rem 64
705 C=C-1:ON-(C=0)+1GOTO265,800      :rem 184
800 POKES,0:FORC=1TOLE:Y=0:X=0:V=0
                                     :rem 120
805 Z=8142+X+Y:X=X+1:POKES-1,188-Y:IFX=21
    THEN825                          :rem 137
810 IFV=0THENY=Y-22:IFY=-66THENV=1
                                     :rem 133
815 IFV=1THENY=Y+22:IFY=0THENV=0      :rem 31
820 POKEZ,32:POKE8142+X+Y,33:POKE8142+X+Y
    +K,1:GOTO805                     :rem 167
825 POKEZ,32:NEXTC:POKES-1,0         :rem 235
830 IFT>0THENFORX=TT0STEP-1:PRINT"{HOME}
    {2 DOWN}{RVS}";X:SC=SC+T:PRINT"{HOME}
    {3 DOWN}{5 LEFT}{RVS}";SC        :rem 164
835 IFT>0THENPOKES,0:POKES,170+T:NEXTX
                                     :rem 206
840 POKES,0:GOSUB600:GOTO100         :rem 206
900 FORY=1TOLE:POKES-1,0:FORX=255TO200STE
    P-1:POKES,X:NEXTX,Y:POKES,0      :rem 192
905 LI=LI-1:GOSUB600:GOTO100         :rem 183
```

Time has run out for the bunny (VIC version).

Program 3: Atari Bunny Hop

```
CE 0 POKE 752,1:PRINT "{CLEAR}":POSI
     TION 15,10:PRINT "BUNNY HOP":PO
     SITION 10,12:PRINT "USE JOYSTIC
     K PORT 1":POSITION 14,14
AF 1 FOR A=1536 TO 1536+126:READ B:P
     OKE A,B:NEXT A
CP 2 PRINT "PLEASE WAIT":FOR T=1 TO
     1000:NEXT T:GRAPHICS 17:SETCOLO
     R 0,0,14:SETCOLOR 2,12,8:SETCOL
     OR 3,2,8:DIM T$(3)
FM 3 SND=150:OPEN #1,4,0,"K:"
MG 11 CHBAS=57344:CHSET=(PEEK(106)-8
      )*256:FOR I=0 TO 1023:POKE CHS
      ET+I,PEEK(CHBAS+I):NEXT I:POKE
       756,CHSET/256
CF 12 READ NCHR:FOR I=1 TO NCHR:READ
       RPLC:FOR J=0 TO 7:READ A:POKE
       CHSET+8*RPLC+J,A:NEXT J:NEXT
      I
FL 25 LE=1:L=10:CT=4:LI=3:GOSUB 600
MN 100 POSITION 5,0:PRINT #6;"BUNNY
        HOP":PRINT #6;"TIME";:PRINT
       #6;"{5 SPACES}SCORE"
PG 105 COLOR 199:PLOT 0,22:DRAWTO 19
       ,22
LO 115 POSITION 0,12:PRINT #6;"{2 M}
       ":POSITION 18,12:PRINT #6;"
       {2 M}":POSITION 0,18:PRINT #6
       ;"{2 M}":POSITION 18,18
BH 116 PRINT #6;"{2 M}"
EL 120 FOR Y=8 TO 20 STEP 6:FOR X=1
       TO L:POSITION RND(1)*19,Y:PRI
       NT #6;"F":NEXT X:NEXT Y
ID 125 FOR Y=10 TO 16 STEP 6:FOR X=1
        TO L:POSITION RND(1)*19,Y:PR
       INT #6;"K":NEXT X:NEXT Y
HC 130 C=CT:FOR R=1 TO C
IG 135 Y=INT(RND(1)*10)+6:X=RND(1)*19
BF 136 LOCATE X,Y,PEEK:IF PEEK<>32 T
       HEN 135
CL 137 IF Y=8 OR Y=10 OR Y=14 OR Y=1
       6 OR Y=20 THEN 135
CO 150 POSITION X,Y:PRINT #6;CHR$(234)
FC 152 NEXT R:D=76:CT=CT+1
OF 155 POKE 18,0:POKE 19,0:POKE 20,0
JB 190 X=0:Y=11
IN 200 COLOR 32:PLOT X,Y:SND=50
CJ 205 IF STICK(0)=7 THEN X=X+1:D=76
       :IF X>19 THEN X=0:POKE 77,0
OE 206 LOCATE X,Y+1,Z:IF Z=75 AND (
       NOT (B) OR Z1) THEN X=X+1:IF
       X>19 THEN X=0
FB 210 IF STICK(0)=11 THEN X=X-1:D=6
       8:IF X<0 THEN X=19:POKE 77,0
NP 215 LOCATE X,Y+1,Z:IF Z=70 AND (
       NOT (B) OR Z1) THEN X=X-1:IF
       X<0 THEN X=19
HC 220 B=(B=0):IF B OR Z1 THEN QQ=US
       R(1536)
JO 225 LOCATE X,Y+1,Z:IF A=0 AND Z<>
       32 THEN IF STICK(0)=14 OR STR
       IG(0)=0 OR V THEN A=7-W*2:SOU
       ND 3,150,12,15
HD 230 IF A<4-W AND Z=32 OR Z=234 TH
       EN Y=Y+1
KB 235 IF A<>0 THEN A=A-1:IF A>3-W T
       HEN Y=Y-1:SND=SND-2:SOUND 3,S
       ND,10,15
CE 240 LOCATE X,Y,ZZ:IF ZZ=234 THEN
       700
CM 245 IF Z=199 THEN 900
DC 250 IF ZZ<>32 THEN Y=Y+1:SND=SND+
       20:SOUND 3,SND,10,15:GOTO 260
AD 255 SOUND 3,150,10,0
HH 260 IF ZZ=234 THEN 700
DI 265 COLOR D:PLOT X,Y:FOR G=1 TO I
       :NEXT G
MD 270 POSITION 16,1:PRINT #6;SC
NL 275 T=INT(60-(PEEK(19)*256+PEEK(2
       0))/60):IF T<1 THEN 400
IE 277 T$=STR$(T):IF LEN(T$)=1 THEN
       POSITION 6,1:PRINT #6;T$;" ":
       GOTO 200
FG 280 POSITION 6,1:PRINT #6;T:GOTO
       200
EI 400 POSITION 4,1:PRINT #6;"  0  "
       :Z1=1:IF T>-30 THEN 200
LL 405 XX=RND(1)*16:YY=RND(1)*16+4:L
       OCATE XX,YY,Z:IF Z=234 OR (Z=
       32 AND RND(1)>0.5) THEN 405
IF 410 COLOR 32:PLOT XX,YY:GOTO 200
ME 500 SOUND 3,0,0,0:POSITION 4,11:P
       RINT #6;"ANOTHER TRY?"
FJ 503 DIM A$(1)
DL 505 GET #1,A
FE 510 IF A=ASC("Y") THEN CLR :DIM T
       $(3):SND=150:GOTO 25
GI 515 IF A=ASC("N") THEN END
GK 520 GOTO 505
IH 599 GOTO 599
OK 600 IF LI<0 THEN 500
ME 605 PRINT #6;"{CLEAR}":POSITION 8
       ,23:IF LI>0 THEN FOR X=1 TO L
       I:PRINT #6;"D";:NEXT X
HF 610 LE=LE+1:Z1=0:W=0:V=0:IF LE>19
        THEN W=1
NI 615 IF INT(LE/10)=LE/10 THEN LI=L
       I+1
KK 620 IF LE/5=INT(LE/5) THEN V=-1
BO 625 IF LE/10=INT(LE/10) THEN CT=4
GJ 630 L=L-1:IF L<3 THEN L=10
HO 635 RETURN
MC 700 FOR S=20 TO 1 STEP -1:SOUND 3
       ,126+S*3,10,15:SC=SC+1:POSITI
       ON 16,1:PRINT #6;SC:NEXT S
LI 705 C=C-1:ON -(C=0)+1 GOTO 265,800
LO 800 SOUND 3,130,10,0:FOR C=1 TO L
       E:Y=0:X=0:NEXT C
AE 840 SOUND 0,0,0,0:GOSUB 600:GOSUB
        100
```

The bunny prepares to eat a carrot (Atari version).

```
ND 900  FOR Y=1 TO LE:FOR X=205 TO 20
        0 STEP -1:SOUND 3,X,10,15:NEX
        T X:NEXT Y
MB 905  SOUND 3,0,0,0:LI=LI-1:GOSUB 6
        00:GOTO 100
PJ 1536  DATA 104,24,165,88,105,160
MB 1542  DATA 133,203,165,89,105,0
GA 1548  DATA 133,204,32,82,6,24
CF 1554  DATA 165,88,105,200,133,203
MC 1560  DATA 165,89,105,0,133,204
GL 1566  DATA 32,105,6,24,165,88
PK 1572  DATA 105,24,133,203,165,89
IO 1578  DATA 105,1,133,204,32,82
HA 1584  DATA 6,24,165,88,105,64
MF 1590  DATA 133,203,165,89,105,1
IP 1596  DATA 133,204,32,105,6,24
CG 1602  DATA 165,88,105,144,133,203
MG 1608  DATA 165,89,105,1,133,204
NN 1613  DATA 32,82,6,96
NL 1615  DATA 160,0,177,203,170,160,1
         ,177,203,136,145,203,200,200
         ,192,20,208,245,136,138,145,
         203,96,160,19,177,203,170
KE 1616  DATA 136,177,203,200,145,203
         ,136,136,192,255,208,245,200
         ,138,145,203,96
OC 2500  DATA 7,32,0,0,0,0,0,0,0,0,0,44
         ,8,12,6,189,127,124,56,70,36
         ,16,48,96,189,254,62,28,34,3
         8,0,102,255,255
KG 2505  DATA 102,0,0,0,39,66,40,134,
         51,136,102,102,102
FP 2506  DATA 42,0,48,188,255,255,188
         ,48,0
MN 2507  DATA 43,0,102,255,255,102,0,
         0,0
```

©

Blueberries

Bill Root

Don't be fooled by the name. "Blueberries" is a fast-paced strategic game for one or two players. Originally written for Atari computers with at least 16K memory, versions are included for Commodore 64, IBM PC, and PCjr. Two joysticks required.

Picking blueberries might seem to be a pleasant task for children on a hot dusty summer day. But not in this game. You won't have time for a nap in the shade.

First you must plant the seeds for the blueberries. And once they grow into blueberries, you must pick them before they grow into redberries or rot altogether.

When you run the Atari version of "Blueberries," you will first see a title screen which says: GTIA/CTIA (G/C)? Push either G or C depending on which chip you have. (Computers purchased after January 1982 probably have the GTIA chip.) If you don't know which you have, just choose one—you can stop the program later and rerun it if the colors are wrong.

Dividing Up The Farm

Then, in the middle of the screen you'll see the actual playing field, which is split up into two planting fields. Player 1 plays on the upper field; player 2 plays on the lower field.

The very bottom of the screen displays the current options. Pressing the OPTION and SELECT keys will change these. Select the options you want for the game. HANDS means the number of farmhands you'll have to help with the picking. The various LIMITs mean that the game will end once one player reaches that LIMIT. A NO LIMIT game continues until one player loses all of his farmhands.

Once the options are chosen and each player has a joystick (player 1 uses port 1, player 2 uses port 2), the game can be started by pressing START. (One-player games can be played from either joystick port, depending on which field you want to play.)

Meet The Farmhands

Player 1 controls the small farmhand standing in the lower left corner of the top field (that's Farmer Jack). Player 2 controls the farmhand standing in the upper right corner of the bottom field (that's Farmer Bob). Moving the joysticks in the four compass directions moves the farmers similarly.

Try moving the farmers around their fields. You will notice that you can't go through the bushes separating the fields. If you try to do so you hear a noise.

In the upper right or lower left corner of each field, you'll see a small shed where the blueberry seeds are stored. In order to get the seeds, you must maneuver your farmer into the shed. Once you go into the shed, you will be placed outside of it automatically, and you will hear a short, razzy sound.

It's Planting Time

Now you can plant the seeds by moving your farmer while holding down the fire button of your joystick. The seeds are small, long, and light green.

You will have to return to the shed periodically to get more seeds, as your farmer can get only a limited number of seeds each time.

Once the seeds are planted, they should soon start growing into blueberries. The growth of the blueberries is random, however; the seeds that have been planted the longest will not necessarily grow into blueberries first.

Harvesting

Picking the blueberries is even easier than planting the seeds—simply move your farmer over the blueberries.

You may notice, while picking blueberries, that when you run over the seeds they disappear. You are not picking the seeds back up when you do this. What you are doing is destroying them, and they can no longer grow into blueberries.

You may also notice that some of the blueberries turn red after a while. This is actually the second stage of the berry metamorphosis: redberries. The redberry stage represents the degradation (due primarily to age) of the blueberries. Redberries, while they can be picked, aren't worth as many points as the blueberries.

Redberries are less desirable not only because of their lower point value, however. Because blueberries have already aged by the time they turn into redberries, redberries are susceptible to rotting.

Once a redberry has rotted it turns white (although it may appear a very light green on your TV). Whiteberries are not to be picked; in fact, any farmhand attempting to do so will be forced to retreat to the farmhouse to recover from the ill effects.

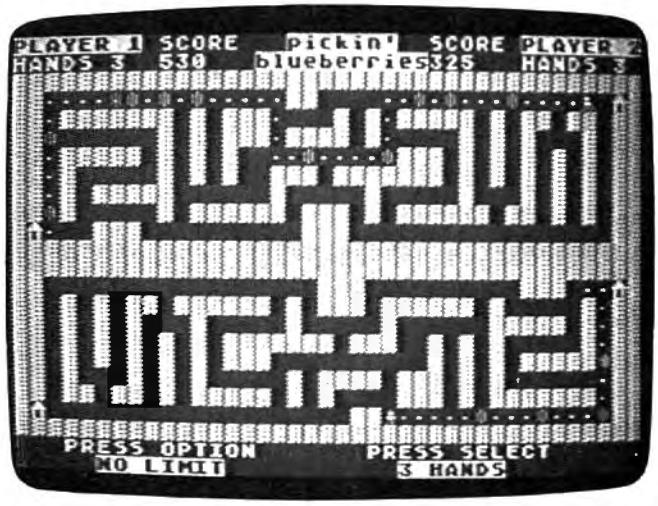
The Payoff

Each player gains 10 points for each seed planted, 50 points for each blueberry picked, and 25 points for each redberry picked. In turn, each player loses 5 points for each seed run over and 200 points for running into a whiteberry.

The game will end once one player loses all of his farmhands or when one player reaches the set LIMIT. At this point PLAYER 1 or PLAYER 2 at the top of the screen will flash to show who won the game.

Blueberries can be restarted at any time while the program is running by pressing START (on the Atari). This, however, puts you in the option-selecting mode. Press START again to begin game play.

Blueberries can also be played with just one player, and since the speed increases as the game progresses, it can be just as challenging as the two-player game. The player can play on either field by plugging the joystick into port 1 or port 2. The object of a one-player game is simply to beat a high score.



One berry has turned overripe (Atari version).

Program 1: Atari Blueberries

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

```
FP 10 GOTO 5000
IM 100 S1=PEEK(632):S=PEEK(633):ST1=
       PEEK(644):ST=PEEK(645):POKE 7
       7,0:IF S=15 AND S1=15 THEN 39
       0
OL 110 DXY1=(S=7)-(S=11)+40*((S=13)-
       (S=14)):DXY2=(S1=7)-(S1=11)+4
       0*((S1=13)-(S1=14))
BO 120 Z=(S1=7)-(S1=11):IF Z THEN FB
       =68+Z
NP 130 IF ST OR (ST=0 AND SD=0) THEN
        POKE XY1,0:XY1=XY1+DXY1
IB 140 IF ST=0 AND SD>0 THEN POKE XY
       1,70:SOUND 0,25,10,8:SD=SD-1:
       XY1=XY1+DXY1:SC1=SC1+10:SOUND
        0,0,0,0
GK 170 IF ST1 OR (ST1=0 AND SD1=0) T
       HEN POKE XY2,0:XY2=XY2+DXY2
EL 180 IF ST1=0 AND SD1>0 THEN POKE
       XY2,70:SOUND 1,10,10,8:SD1=SD
       1-1:XY2=XY2+DXY2:SC2=SC2+10:S
       OUND 1,0,0,0
KK 200 A=PEEK(XY1):A1=PEEK(XY2)
MN 210 IF A=65 THEN SOUND 0,250,8,14
       :XY1=XY1-DXY1:FOR X=1 TO 2:NE
       XT X:SOUND 0,0,0,0
PP 220 IF A1=65 THEN SOUND 1,200,8,1
       4:XY2=XY2-DXY2:FOR X=1 TO 2:N
       EXT X:SOUND 1,0,0,0
DO 250 IF A=66 THEN SOUND 0,200,6,10
       :FOR Q=1 TO 50:NEXT Q:GOSUB 8
       00:XY1=XY1-DXY1:SOUND 0,0,0,0
```

WE LOVE COMMODORE
and
We Love Our Customers

That's why we only sell and support Commodore 64 and Vic 20 computers!! We have • the best prices • over 1000 programs • 500 accessories • absolutely the best service • one day express mail delivery • immediate replacement warranty • 15 day free trial • programming knowledge • technical knowledge • we are the only one in the U.S.A. with complete support for Commodore 64 and Vic 20 computers!!

PROTECTO ENTERPRIZES Box 550, Barrington, IL 60010

SEND IN THIS COUPON TODAY FOR A FREE 64-PAGE "EXCLUSIVE COMMODORE-64 AND VIC-20 CATALOG" — PLUS OUR SPECIAL "BUY MORE—SAVE MORE COUPON"

(Save up to $500 on software and accessories) .

Name _____

Address _____

City _____ State _____ Zip Code _____

No One! But No One! Can Compare

TO

PROTECTO ENTERPRIZES

TO ORDER WRITE OR CALL: PROTECTO ENTERPRIZES, BOX 550, BARRINGTON, IL 60010

Call 312/ 382-5244

8 to 5 Weekdays 9-12 Saturdays

(See Next 10 Pages)

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	$18.95	$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 — COD

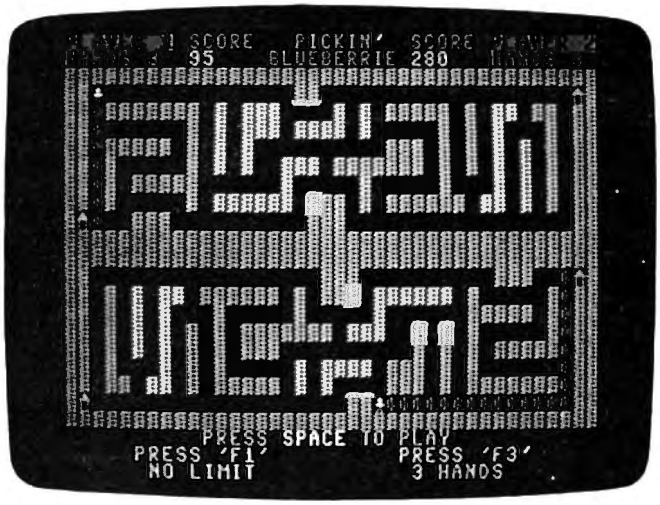
PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)

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

```
MC 260 IF A1=66 THEN SOUND 1,200,6,1
       0:FOR Q=1 TO 50:NEXT Q:GOSUB
       850:XY2=XY2-DXY2
EG 300 POKE XY1,68:POKE XY2,FB:SOUND
        1,0,0,0
HE 310 IF A=71 THEN SC1=SC1+50:U=USR
       (1630)
KN 320 IF A1=71 THEN SC2=SC2+50:U=US
       R(1653)
HJ 330 IF A=72 THEN SC1=SC1+25:U=USR
       (1630)
LC 340 IF A1=72 THEN SC2=SC2+25:U=US
       R(1653)
FO 350 IF A=73 THEN C=38:GOTO 2000
FN 360 IF A1=73 THEN C=6:GOTO 2020
BC 370 IF A=70 THEN SOUND 0,100,10,8
       :SC1=SC1-5:SOUND 0,0,0,0
CD 380 IF A1=70 THEN SOUND 1,75,10,8
       :SC2=SC2-5:SOUND 1,0,0,0
IF 390 POSITION 26,1:? SC1;" ":POSIT
       ION 9,1:? SC2;" ":POKE SAVMSC
       +55,226:POKE SAVMSC+72,40

AP 400 IF PEEK(53279)=6 THEN 2170
AA 410 IF SC1>=LIM OR SC2>=LIM THEN
       2100
JB 420 LV=210+(SC1+SC2)*1.7E-03+0.5
EM 430 IF LV>255 THEN LV=255
DB 440 POKE 255,LV
GD 450 GOTO 100
HA 800 SD=INT(50*RND(0)+51)-INT(SC1/
       5000)+INT(SC2/1000)
CG 810 IF SD<25 THEN SD=INT(50*RND(0
       )+51)
HK 820 RETURN

KG 850 SD1=INT(50*RND(0)+51)-INT(SC2
       /5000)+INT(SC1/1000)
IN 860 IF SD1<25 THEN SD1=INT(50*RND
       (0)+51)
HP 870 RETURN
EB 1000 POKE 559,0:? "PLAYER 1 SCORE
        {3 SPACES}pickin' SCORE PLA
        YER 2";
DG 1010 ? "HANDS 3  0{5 SPACES}blueb
        erries0{5 SPACES}HANDS 3 ";
KK 1030 ? "{40 A}";
IK 1040 ? "{2 A}{15 SPACES}{2 A}
        {19 SPACES}{B}{A}";
JP 1050 ? "{2 A} {7 A} {A} {3 A}
        {4 SPACES}{A} {A} {5 A} {A}
        {2 A} {2 A} {2 A}";
JH 1060 ? "{2 A}{7 SPACES}{A} {A}
        {3 A} {4 A} {A}{5 SPACES}
        {A} {A} {A}{3 SPACES}{A}
        {2 A}";
JK 1070 ? "{2 A} {5 A} {A} {A} {A}
        {10 SPACES}{3 A} {A} {A} {A}
        {A} {A} {2 A}";
JO 1080 ? "{2 A} {A}{5 SPACES}{A}
        {A} {A}  {3 A} {7 A} {A}
        {A} {A} {A} {A} {2 A}";
JI 1090 ? "{2 A} {A} {5 A}
        {6 SPACES}{A}{5 SPACES}{A}
        {5 SPACES}{A}{3 SPACES}{A}
        {A} {A} {2 A}";
JK 1100 ? "{2 A} {A}{5 SPACES}{A}
        {6 A} {3 A} {A} {6 A} {2 A}
        {A} {A} {2 A}";
IM 1110 ? "{A}{B}{3 SPACES}{3 A}
        {10 SPACES}{3 A}{17 SPACES}
        {2 A}";
KK 1120 ? "{40 A}";
KL 1130 ? "{40 A}";
IP 1140 ? "{2 A}{17 SPACES}{3 A}
        {10 SPACES}{3 A}{3 SPACES}
        {B}{A}";
JP 1150 ? "{2 A} {A} {A} {2 A} {6 A}
         {A} {3 A} {6 A} {A}
        {5 SPACES}{A} {2 A}";
JG 1160 ? "{2 A} {A} {A} {A}
        {3 SPACES}{A}{5 SPACES}{A}
        {5 SPACES}{A}{6 SPACES}{5 A}
         {A} {2 A}";
JO 1170 ? "{2 A} {A} {A} {A} {A}
        {A} {7 A} {3 A}  {A} {A}
        {A}{5 SPACES}{A} {2 A}";
JM 1180 ? "{2 A} {A} {A} {A} {A}
        {A} {3 A}{10 SPACES}{A} {A}
        {A} {5 A} {2 A}";
JL 1190 ? "{2 A} {A}{3 SPACES}{A}
        {A} {A}{5 SPACES}{A} {4 A}
        {3 A} {A} {A}{7 SPACES}{2 A}
        ";
JM 1200 ? "{2 A} {2 A} {2 A} {A}
        {5 A} {A} {A}{4 SPACES}{3 A}
         {A} {7 A} {2 A}";
IJ 1210 ? "{A}{B}{19 SPACES}{2 A}
        {15 SPACES}{2 A}";
KL 1220 ? "{40 A}";
LM 1250 SAVMSC=PEEK(88)+256*PEEK(89)
        :XY1=SAVMSC+557:XY2=SAVMSC+4
        02:FB=67:POKE XY1,68:POKE XY
        2,FB
BB 1260 U=USR(1676):SC1=0:SC2=0:GOTO
         6
PG 1300 POKE 84,22:? "{3 SPACES}PRES
        S OPTION";:POKE 85,20:? "  P
        RESS SELECT";
ME 1305 A$="{3 SPACES}NO LIMIT":LIM=
        1.0E+95:A1$=" 3 HANDS":L=3:L
        1=3:POKE 84,23:POKE 85,2:? A
        $;:POKE 85,23:? A1$;
AB 1310 POKE 559,34:POKE 53279,8
BO 1320 PK=PEEK(53279)
OK 1330 IF PK=6 THEN POKE 559,0:FOR
        Q=1 TO 125:NEXT Q:POKE 559,3
        4:QID=L:GOTO 100
KB 1340 IF PK=3 THEN SOUND 0,200,10,
        10:GOSUB 1400
KE 1350 IF PK=5 THEN SOUND 0,100,10,
        10:GOSUB 1500
MI 1360 GOTO 1310
EH 1400 IF A$="{3 SPACES}NO LIMIT" T
        HEN A$="  5000 LIMIT":LIM=50
        00:GOTO 1600
EI 1410 IF A$="  5000 LIMIT" THEN A$
        =" 10000 LIMIT":LIM=10000:GO
        TO 1600
PN 1420 IF A$=" 10000 LIMIT" THEN A$
        =" 50000 LIMIT":LIM=50000:GO
        TO 1600

NK 1430 IF A$=" 50000 LIMIT" THEN A$
        =" 100000 LIMIT":LIM=100000:
        GOTO 1600
IP 1440 IF A$=" 100000 LIMIT" THEN A
        $=" 500000 LIMIT":LIM=500000
        :GOTO 1600
FE 1450 IF A$=" 500000 LIMIT" THEN A
        $="{3 SPACES}NO LIMIT":LIM=1
        .0E+95:GOTO 1600
```

```
HB 1500 IF A1$=" 3 HANDS" THEN A1$="
        5 HANDS":L=5:L1=5:POKE SAVM
        SC+46,L+16:POKE SAVMSC+78,L1
        +16:GOTO 1600
GO 1510 IF A1$=" 5 HANDS" THEN A1$="
        3 HANDS":L=3:L1=3:POKE SAVM
        SC+46,L+16:POKE SAVMSC+78,L1
        +16:GOTO 1600
ED 1600 POKE 84,23:POKE 85,2:? A$;"
        {4 SPACES}";:POKE 85,23:? A1
        $;:SOUND 0,0,0,0:FOR Q=1 TO
        50:NEXT Q:RETURN
GG 2000 FOR Q=74 TO 79:POKE XY1,Q:SO
        UND 0,2*Q,8,7:SOUND 1,250-2*
        Q,6,6:FOR T=1 TO 15:NEXT T:N
        EXT Q:GOSUB 2200
NK 2010 SOUND 0,0,0,0:SOUND 1,0,0,0:
        POKE XY1,CO:GOTO 2040
GO 2020 FOR Q=74 TO 79:POKE XY2,Q:SO
        UND 0,2*Q,8,7:SOUND 1,250-2*
        Q,6,6:FOR T=1 TO 15:NEXT T:N
        EXT Q:GOSUB 2250
KE 2030 SOUND 0,0,0,0:SOUND 1,0,0,0:
        POKE XY2,CO
NK 2040 IF C=6 THEN L1=L1-1:SC2=SC2-
        200:POSITION C,1:? L1
HL 2050 IF C=38 THEN L=L-1:SC1=SC1-2
        00:POSITION C,1:? L
AA 2060 IF L1=0 OR L=0 THEN POSITION
         9,1:? SC2:POSITION 26,1:? S
        C1:GOTO 2100
JE 2080 GOTO 100
PM 2100 IF SC1>SC2 THEN B$="PLAYER 2
        ":B1$="PLAYER 2":C=32:GOTO 2
        120
AD 2110 B$="PLAYER 1":B1$="PLAYER 1"
        :C=0
EO 2120 POSITION C,0:? B$;:FOR X=1 T
        O 25:NEXT X:POSITION C,0:? B
        1$;
GD 2130 POKE 53279,8
BP 2140 PK=PEEK(53279)
JM 2150 IF PK<>7 THEN 2170
OE 2160 COL=COL+1:IF COL>15 THEN COL
        =1
BB 2165 SETCOLOR 4,COL,6:GOTO 2120
LG 2170 GRAPHICS 0:POKE 752,1:POKE 7
        56,CHBAS:POKE 16,64:POKE 537
        74,64
PF 2180 SETCOLOR 2,15,6:SETCOLOR 1,1
        5,14:SETCOLOR 4,10,4:POSITIO
        N 0,0:L=QID:L1=L:G=2190:GOTO
         1000
GC 2190 POKE 84,22:? "{3 SPACES}PRES
        S OPTION";:POKE 85,20:? "  P
        RESS SELECT";:POKE 84,23:POK
        E 85,2:? A$;:POKE 85,23:? A1
        $;
KP 2195 POKE SAVMSC+46,L+16:POKE SAV
        MSC+78,L1+16:GOTO 1310
GO 2200 IF PEEK(53770)>127 THEN POKE
         XY1,0:XY1=SAVMSC+557:GOTO 2
        220
BN 2210 POKE XY1,0:XY1=SAVMSC+802
JH 2220 IF L=1 THEN CO=79:RETURN
IH 2230 FOR Q=79 TO 74 STEP -1:SOUND
         0,2*Q,8,6:SOUND 1,250-2*Q,6
        ,7:POKE XY1,Q:FOR T=1 TO 15:
        NEXT T:NEXT Q
CO 2240 SOUND 0,0,0,0:SOUND 1,0,0,0:
        CO=68:RETURN
GP 2250 IF PEEK(53770)>127 THEN POKE
         XY2,0:XY2=SAVMSC+402:GOTO 2
        270
CH 2260 POKE XY2,0:XY2=SAVMSC+157
MN 2270 IF L1=1 THEN CO=79:RETURN
IN 2280 FOR Q=79 TO 74 STEP -1:SOUND
         0,2*Q,8,6:SOUND 1,250-2*Q,6
        ,7:POKE XY2,Q:FOR T=1 TO 15:
        NEXT T:NEXT Q
EN 2290 SOUND 0,0,0,0:SOUND 1,0,0,0:
        CO=FB:RETURN
BO 5000 CHBAS=PEEK(742)-4:D=(PEEK(74
        2)-4)*256
KL 5010 RESTORE 5040:FOR X=1536 TO 1
        591
IC 5020 READ Y:POKE X,Y
EF 5030 NEXT X:U=USR(1536)
JG 5040 DATA 104,173,244,2,133,204,1
        69,0,133,203,133,205,173,230
        ,2,56,233,4,133,206,133,207,
        162,0,160,0,177,203
AI 5050 DATA 145,205,192,255,240,4,2
        00,24,144,244,224,3,240,8,23
        2,230,206,230,204,24,144,230
        ,165,207,141,244,2,96
DD 5100 GRAPHICS 17:POKE 16,64:POKE
        53774,64
LJ 5110 ? #6:? #6:? #6:? #6;"
        {7 SPACES}PICKIN'"
GF 5120 ? #6:? #6;"{5 SPACES}BLUEBER
        RIES"
FM 5130 ? #6:? #6:? #6:? #6:? #6:? #
        6:? #6:? #6;"  GTIA/CTIA (G/
        C)?"
GC 5140 POKE 764,255
OA 5150 IF PEEK(764)=18 THEN C1=71:C
        2=72:? #6;"{4 SPACES}GTIA":G
        OTO 5200
OD 5160 IF PEEK(764)=61 THEN C1=72:C
        2=71:? #6;"{4 SPACES}CTIA":G
        OTO 5200
NA 5170 GOTO 5140
PL 5200 ? #6:? #6:? #6;"  PLEASE WAI
        T . . ."
HL 5300 DATA 104,169,255,141,0,210,1
        69,164
KO 5310 DATA 141,1,210,32,79,6,169,2
        00
AE 5320 DATA 141,0,210,169,162,141,1
        ,210
LD 5330 DATA 32,79,6,169,150,141,0,2
        10
LI 5340 DATA 169,162,141,1,210,32,79
        ,6
EP 5350 DATA 32,79,6,169,0,141,0,210
HM 5360 DATA 162,0,232,224,150,208,2
        51,169
AP 5370 DATA 125,141,0,210,169,172,1
        41,1
FC 5380 DATA 210,32,79,6,169,0,141,0
DE 5390 DATA 210,169,0,141,1,210,96,
        162
AJ 5400 DATA 0,232,160,0,200,192,255
        ,208
MG 5410 DATA 251,224,3,208,244,96
BJ 5420 DATA 169,155,141,2,6,169,100
        ,141
JB 5430 DATA 15,6,169,50,141,28,6,16
        9
```

```
GF 5440 DATA 75,141,56,6,76,0,6
BD 5450 DATA 169,255,141,2,6,169,200
        ,141
MF 5460 DATA 15,6,169,150,141,28,6,1
        69
JE 5470 DATA 125,141,56,6,76,0,6
CI 5480 DATA 104,169,255,133,254,160
        ,155
GJ 5490 DATA 162,6,169,7,32,92,228,9
        6,165
FD 5500 DATA 255,197,254,240,5,198,2
        54,24
FH 5510 DATA 144,59,169,255,133,254,
        162,0
IK 5520 DATA 134,253,165,88,133,252,
        24,165
HH 5530 DATA 89,101,253,133,253,172,
        10,210
CH 5540 DATA 177,252,201,70,208,7,16
        9,71
ED 5550 DATA 145,252,24,144,19,201,7
        1,208
MI 5560 DATA 7,169,72,145,252,24,144
        ,8
CG 5570 DATA 201,72,208,4,169,73,145
        ,252
CM 5580 DATA 232,224,4,208,203,76,98
        ,228
LI 5590 RESTORE 5300:FOR X=1536 TO 1
        763
FH 5600 READ Y:POKE X,Y:NEXT X
NJ 6000 FOR X=1 TO 15:READ A:IF A=-1
         THEN A=C1
FF 6010 IF A=-2 THEN A=C2
AC 6020 Z=A*8
CJ 6030 FOR Y=0 TO 7:READ N1:POKE Z+
        D+Y,N1:NEXT Y:NEXT X
LD 6040 POKE 82,0:POKE 83,40:DIM A$(
        15),A1$(10),B$(8),B1$(8):COL
        =0
LH 6050 GRAPHICS 0:POKE 752,1:POKE 7
        56,CHBAS:POKE 16,64:POKE 537
        74,64
AN 6060 SETCOLOR 2,15,6:SETCOLOR 1,1
        5,14:SETCOLOR 4,10,4:POSITIO
        N 0,0:G=1300:GOTO 1000
IK 6070 DATA 65,127,246,127,246,127,
        246,127,246
AB 6080 DATA 66,24,60,126,255,102,10
        2,102,102
FC 6090 DATA 67,0,4,14,4,30,36,74,17
CE 6100 DATA 68,0,8,28,8,62,8,20,36
DP 6110 DATA 69,0,32,112,32,120,36,8
        2,136
GG 6120 DATA 70,0,0,0,0,0,48,0,0,0
AK 6130 DATA -1,0,8,42,42,42,42,8,0
IC 6140 DATA -2,0,16,84,84,84,84,16,
        0
EA 6150 DATA 73,0,24,126,126,126,126
        ,24,0
IB 6160 DATA 74,0,16,56,16,56,16,40,
        0
IH 6170 DATA 75,0,0,16,56,16,56,56,4
        0
HO 6180 DATA 76,0,0,0,16,56,124,56,4
        0
IC 6190 DATA 77,0,0,0,0,16,56,254,23
        8
ED 6200 DATA 78,0,0,0,0,0,16,186,255
KF 6210 DATA 79,0,0,0,0,0,0,16,56
```



The blueberries have just ripened in this 64 version of "Blueberries."

Program 2: Blueberries, 64 Version

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

```
10 GOTO5000                                    :rem 95
100 S1=PEEK(J1):S=PEEK(J0):SV=(S1AND16)/1
    6:SU=(SAND16)/16:S1=S1AND15:S=SAND15
                                               :rem 99
105 IFS=15ANDS1=15THEN380                      :rem 223
110 D1=(S=11)-(S=7)+40*((S=14)-(S=13)):D2
    =(S1=11)-(S1=7)+40*((S1=14)-(S1=13))
                                               :rem 137
130 IFSUOR(SU=0ANDSD=0)THENPOKEX1,32:X1=X
    1+D1                                       :rem 74
140 IFSU=0ANDSD>0THENPOKEX1,70:SD=SD-1:X1
    =X1+D1:SA=SA+10:F=25:GOSUB 2500
                                               :rem 135
170 IFSVOR(SV=0ANDSE=0)THENPOKEX2,32:X2=X
    2+D2                                       :rem 85
180 IFSV=0ANDSE>0THENPOKEX2,70:SE=SE-1:X2
    =X2+D2:SB=SB+10:F=10:GOSUB 2600
                                               :rem 144
200 A=PEEK(X1):A1=PEEK(X2)                     :rem 248
210 IFA=65THENX1=X1-D1:F=250:GOSUB 2500
                                                :rem 0
220 IFA1=65THENX2=X2-D2:F=200:GOSUB 2600
                                               :rem 49
250 IFA=66THENGOSUB800:X1=X1-D1:F=200:GOS
    UB 2500                                    :rem 82
260 IFA1=66THENGOSUB850:X2=X2-D2:F=200:GO
    SUB 2600                                   :rem 141
300 POKEX1,68:POKEX1+CO,7:POKEX2,68:POKEX
    2+CO,7                                     :rem 151
310 IFA<70ORA>73THEN350                        :rem 88
320 ON(A-69)GOTO330,331,332,333                :rem 123
330 SA=SA-5:F=100:GOSUB2500:GOTO350
                                               :rem 55
331 SA=SA+25:GOTO350                           :rem 153
332 SA=SA+50:GOTO350                           :rem 152
333 C=0:GOTO2000                               :rem 126
350 IFA1<70ORA1>73THEN380                      :rem 193
360 ON(A1-69)GOTO370,371,372,373               :rem 192
370 SB=SB-5:F=75:GOSUB2600:GOTO380:rem 28
371 SB=SB+25:GOTO380                           :rem 162
372 SB=SB+50:GOTO380                           :rem 161
```

Notes On The Commodore 64 And IBM PC/PCjr Versions Of Blueberries

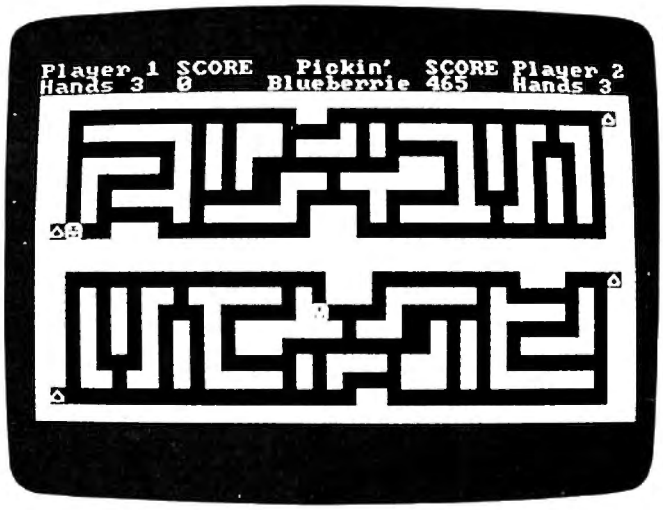
The 64 and PC/PCjr versions are the same as the Atari version except for the scoring routine and the berry development routine. The blueberries in these versions have four stages of development: the seed, the undeveloped berry, the mature berry, and the overripe berry.

If you pick the berry before it has had a chance to sprout, you have five points deducted from your score. If you pick an undeveloped berry, you only get 25 points. If you pick the berry when it is ripe, you receive the full 50 points. If you pick an overripe berry, your farmhand becomes sick and you lose 200 points.

In the Atari version, berries ripen at random times, but in the 64 and PC/PCjr versions all the berries on the screen ripen at the same time. However, the amount of time required for the berries to ripen is determined randomly. Both versions require two joysticks to play, and the Color/Graphics Monitor Adapter board is required to use Program 3 on an IBM PC.

```
373 C=1:GOTO2000                          :rem 131
380 PRINT"{HOME}{DOWN}{25 RIGHT}"SB"
    {LEFT} "                              :rem 219
385 PRINT"{HOME}{DOWN}{8 RIGHT}"SA"{LEFT}
     "                                    :rem 242
390 IFSA>=LIORSB>=LITHEN2100               :rem 7
395 GOSUB 2700                            :rem 234
400 CN=CN+1:IFCN<30+RND(1)*20THEN100
                                          :rem 37
410 CN=0:SYS49152:GOTO100                 :rem 213
800 SD=INT(50*RND(1)+51):RETURN           :rem 72
850 SE=INT(50*RND(1)+51):RETURN           :rem 78
1000 POKE53280,0:POKE53281,0              :rem 22
1005 PRINT"{CLR}{7}{RVS}PLAYER 1{OFF} SCO
     RE{3 SPACES}PICKIN'{2 SPACES}SCORE
     {RVS}PLAYER 2{OFF}";                 :rem 33
1010 PRINT"HANDS 3{2 SPACES}0{5 SPACES}BL
     UEBERRIES0{5 SPACES}HANDS 3 ";
                                          :rem 164
1020 PRINT"{2}AAAAAAAAAAAAAAAAAAAAAAAAAAA
     AAAAAAAAAAAA";                       :rem 140
1030 PRINT"AA{15 SPACES}AA{19 SPACES}BA";
                                          :rem 87
1040 PRINT"AA AAAAAAA A AAA{4 SPACES}A A
     {SPACE}AAAAA A AA AA AA";            :rem 44
1060 PRINT"AA{7 SPACES}A A AAA AAAA A
     {5 SPACES}A A A{3 SPACES}A AA";
                                          :rem 101
1070 PRINT"AA AAAAA A A A{10 SPACES}AAA A
      A A A A AA";                        :rem 232
1080 PRINT"AA A{5 SPACES}A A A{2 SPACES}A
     AA AAAAAAA A A A A A AA";            :rem 44
1090 PRINT"AA A AAAAA{6 SPACES}A
     {5 SPACES}A{5 SPACES}A{3 SPACES}A A
     {SPACE}A AA";                        :rem 230
1100 PRINT"AA A{5 SPACES}A AAAAAA AAA A A
     AAAAA AA A A AA";                    :rem 104
1110 PRINT"AB{3 SPACES}AAA{10 SPACES}AAA
     {17 SPACES}AA";                      :rem 90
1120 PRINT"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     AAAAAAAAAA";                         :rem 248
1130 PRINT"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     AAAAAAAAAA";                         :rem 249
1140 PRINT"AA{17 SPACES}AAA{10 SPACES}AAA
     {3 SPACES}BA";                       :rem 93
1150 PRINT"AA A A AA AAAAAA A AAA AAAAAA
     {SPACE}A{5 SPACES}A AA";             :rem 109
1160 PRINT"AA A A A{3 SPACES}A{5 SPACES}A
     {5 SPACES}A{6 SPACES}AAAAA A AA";
                                          :rem 228
1170 PRINT"AA A A A A A AAAAAAA AAA
     {2 SPACES}A A A{5 SPACES}A AA";
                                          :rem 44
1180 PRINT"AA A A A A A AAA{10 SPACES}A A
      A AAAAA AA";                        :rem 234
1190 PRINT"AA A{3 SPACES}A A A{5 SPACES}A
      AAAA AAA A A{7 SPACES}AA";          :rem 105
1200 PRINT"AA AA AA A AAAAA A A{4 SPACES}
     AAA A AAAAAAA AA";                   :rem 42
1210 PRINT"AB{19 SPACES}AA{15 SPACES}AA";
                                          :rem 87
1220 PRINT"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     AAAAAAAAAA";                         :rem 249
1250 X1=1426:X2=1581:CO=54272:POKEX1,68:P
     OKEX1+CO,7:POKEX2,68:POKEX2+CO,7
                                          :rem 118
1260 SA=0:SB=0:J0=56321:J1=J0-1           :rem 194
1270 FORI=COTOCO+24:POKEI,0:NEXT          :rem 22
1280 POKECO+24,15:POKECO+5,16:POKECO+6,24
     0:POKECO+12,16:POKECO+13,17          :rem 245
1300 PRINT"{GRN}{HOME}{22 DOWN}
     {10 SPACES}PRESS {6}SPACE{GRN} TO PL
     AY"                                  :rem 197
1301 PRINT"{CYN}{5 SPACES}PRESS 'F1'
     {10 SPACES}PRESS 'F3'"               :rem 219
1305 A$="{3 SPACES}NO LIMIT":LI=1E38:A1$=
     " 3 HANDS":L=3:L1=3                  :rem 199
1306 PRINT"{3 SPACES}"A$"{3 SPACES}","
     {5 SPACES}"A1$"{UP}"                 :rem 31
1310 GETI$:IFI$=""THEN1310                :rem 189
1330 IFI$=" "THEN100                       :rem 4
1340 IFI$="{F1}"THEN1400                  :rem 190
1350 IFI$="{F3}"THEN1500                  :rem 193
1360 GOTO1310                             :rem 200
1400 IFA$="{3 SPACES}NO LIMIT"THENA$="
     {2 SPACES}5000 LIMIT";LI=5000:GOTO13
     06                                   :rem 189
1410 IFA$="{2 SPACES}5000 LIMIT"THENA$="
     {2 SPACES}10000 LIMIT":LI=10000:GOTO
     1306                                 :rem 62
1420 IFA$="{2 SPACES}10000 LIMIT"THENA$="
     {2 SPACES}50000 LIMIT":LI=50000:GOTO
     1306                                 :rem 115
1430 IFA$="{2 SPACES}50000 LIMIT"THENA$="
      100000 LIMIT":LI=100000:GOTO1306
                                          :rem 208
1440 IFA$=" 100000 LIMIT"THENA$=" 500000
     {SPACE}LIMIT":LI=500000:GOTO1306
                                           :rem 5
1450 IFA$=" 500000 LIMIT"THENA$="
     {3 SPACES}NO LIMIT":LI=1E38:GOTO1306
                                          :rem 62
```

```
1500 IFA1$=" 3 HANDS"THENA1$=" 5 HANDS":L
     =5:L1=5:GOTO1600                  :rem 81
1510 IFA1$=" 5 HANDS"THENA1$=" 3 HANDS":L
     =3:L1=3:GOTO1600                  :rem 78
1600 POKE1070,L+48:POKE1102,L1+48:GOTO130
     6                                 :rem 119
2000 IFCTHENL1=L1-1:SB=SB-200:POKEX2,32:X
     2=1581:GOTO2020                   :rem 67
2010 L=L-1:SA=SA-200:POKEX1,32:X1=1426
                                       :rem 164
2020 POKE1070,L+48:POKE1102,L1+48      :rem 55
2021 POKEX1,68:POKEX1+CO,7:POKEX2,68:POKE
     X2+CO,7                           :rem 201
2025 GOSUB2700                         :rem 18
2026 POKECO+4,33:FORI=200TO10STEP-5:POKEC
     O+1,I:FORJ=1TO5:NEXTJ,I:GOSUB2700
                                       :rem 143
2030 IFL<>0ANDL1<>0THEN100             :rem 4
2040 PRINT"{HOME}{DOWN}{25 RIGHT}"SB"
     {LEFT} "                          :rem 6
2050 PRINT"{HOME}{DOWN}{8 RIGHT}"SA"
     {LEFT} "                          :rem 25
2060 FORTD=1TO3000:NEXT                :rem 148
2100 B$="1":IFSA<SBTHENB$="2"          :rem 81
2110 PRINT"{CLR}{10 DOWN}{13 RIGHT}PLAYER
      "B$" WINS"                       :rem 3
2130 PRINT"{4 DOWN}{8 SPACES}PRESS SPACE
     {SPACE}TO CONTINUE"               :rem 220
2140 GETA$:IFA$<>" "THEN2140           :rem 238
2150 GOTO1000                          :rem 194
2500 POKECO+1,F:POKECO,0:POKECO+4,33:FORI
     =1TO20:NEXT:RETURN                :rem 9
2600 POKECO+8,F:POKECO+7,0:POKECO+11,33:F
     ORI=1TO20:NEXT:RETURN             :rem 161
2700 POKECO+4,0:POKECO+11,0:RETURN:rem 67
5000 PRINT"{CLR}{13 DOWN}{9 SPACES}REDEFI
     NING CHARACTERS"                  :rem 193
5010 POKE55,0:POKE56,48:CLR            :rem 66
5020 POKE56334,PEEK(56334)AND254:POKE1,PE
     EK(1)AND251                       :rem 232
5030 FORI=0TO2047:POKEI+12288,PEEK(I+5324
     8):NEXT                           :rem 76
5040 POKE1,PEEK(1)OR4:POKE56334,PEEK(5633
     4)OR1                             :rem 184
5050 FORI=0TO71:READA:POKE12808+I,A:NEXT
                                       :rem 207
5060 POKE53272,(PEEK(53272)AND240)OR12
                                       :rem 96
5070 FORI=0TO40:READA:POKE49152+I,A:NEXT:
     GOTO1000                          :rem 3
6000 DATA 127,246,127,246,127,246,127,246
                                       :rem 236
6010 DATA 24,60,126,255,102,102,102,102
                                       :rem 98
6020 DATA 0,0,0,0,0,0,0,0              :rem 150
6030 DATA 0,8,28,8,62,8,20,36          :rem 140
6040 DATA 0,0,0,0,0,0,0,0              :rem 152
6050 DATA 0,0,0,0,48,0,0,0             :rem 213
6060 DATA 0,0,8,42,42,8,0,0            :rem 22
6070 DATA 0,16,84,84,84,84,16,0        :rem 249
6080 DATA 0,24,126,126,126,126,24,0
                                       :rem 172
7000 DATA 160,0,169,0,133,251,169,4,133,2
     52,177,251,201,70,240,8,201,71
                                       :rem 145
7010 DATA 240,4,201,72,208,5,170,232,138
                                       :rem 162
7015 DATA 145,251,200,208,234,230,252,165
                                       :rem 220
7020 DATA 252,201,8,208,226,96         :rem 198
```



Player 2 has got a good start (IBM version).

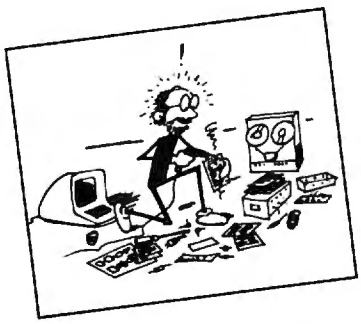
Program 3: PC/PCjr Blueberries

```
2 DEF SEG=0:SCREEN 0,1
3 LOCATE 1,1,0
4 WIDTH 40:KEY OFF
5 GOSUB 5000
6 DEF SEG=&HB800
7 STRIG ON
10 GOTO 1000
100 J1=STICK(0):J2=STICK(1):S=-1*(J2>80)
-2*(J1<30)-3*(J1>80)-4*(J2<30):ST=NOT(ST
RIG(1))
105 J1=STICK(2):J2=STICK(3):S1=-1*(J2>80
)-2*(J1<30)-3*(J1>80)-4*(J2<30):ST1=NOT(
STRIG(3))
110 DXY1=(S=2)-(S=3)+40*((S=4)-(S=1))
115 DXY2=(S1=2)-(S1=3)+40*((S1=4)-(S1=1)
)
120 DXY1=DXY1*2:DXY2=DXY2*2
130 IF ST OR (ST=0 AND SD=0) THEN POKE X
Y1,32:XY1=XY1+DXY1
140 IF ST=0 AND SD>0 THEN POKE XY1,176:P
OKE XY1+1,1:SOUND 200,1:SD=SD-1:XY1=XY1+
DXY1:SC1=SC1+10
170 IF ST1 OR (ST1=0 AND SD1=0) THEN POK
E XY2,32:XY2=XY2+DXY2
180 IF ST1=0 AND SD1>0 THEN POKE XY2,176
:POKE XY2+1,1:SOUND 210,1:SD1=SD1-1:XY2=
XY2+DXY2:SC2=SC2+10
200 A=PEEK(XY1):A1=PEEK(XY2)
210 IF A=219 THEN SOUND 500,1:XY1=XY1-DX
Y1
220 IF A1=219 THEN SOUND 510,1:XY2=XY2-D
XY2
250 IF A=127 THEN SOUND 300,1:GOSUB 800:
XY1=XY1-DXY1
260 IF A1=127 THEN SOUND 310,1:GOSUB 850
:XY2=XY2-DXY2
300 POKE XY1,1:POKE XY2,1
305 POKE XY1+1,14:POKE XY2+1,14
310 IF A=178 THEN SC1=SC1+50
320 IF A1=178 THEN SC2=SC2+50
330 IF A=177 THEN SC1=SC1+25
340 IF A1=177 THEN SC2=SC2+25
350 IF A=15 THEN C=38:GOTO 2000
```

```
360 IF A1=15 THEN C=6:GOTO 2000
370 IF A=176 THEN SOUND 100,1:SC1=SC1-5
380 IF A1=176 THEN SOUND 110,1:SC2=SC2-5
390 LOCATE 2,26:PRINT SC1" ":LOCATE 2,9:
PRINT SC2" "
410 IF SC1>=LIM OR SC2>=LIM THEN 2100
450 CNT=CNT+1:IF CNT>30+RND*20 THEN CNT=
0 ELSE 100
455 DEF SEG:CALL ZZ:DEF SEG=&HB800
456 GOTO 100
800 SD=INT(50*RND+51)-INT(SC1/5000)+INT(
SC2/1000)
810 IF SD<25 THEN SD=INT(50*RND+51)
820 RETURN
850 SD1=INT(50*RND+51)-INT(SC2/5000)+INT
(SC1/1000)
860 IF SD1<25 THEN SD1=INT(50*RND+51)
870 RETURN
1000 CLS:COLOR 7,0,0:PRINT"Player 1 SCOR
E   Pickin'  SCORE Player 2";
1010 PRINT"Hands 3  0      Blueberries0
   Hands 3 ";
1020 W$=CHR$(219):B$=CHR$(32)
1030 COLOR 6,0,0:PRINT STRING$(40,219);
1040 PRINT W$W$STRING$(15,32)W$W$STRING$
(19,32)CHR$(127)W$;
1050 PRINT W$W$B$STRING$(7,219)B$W$B$W$W
$W$STRING$(4,32)W$B$W$B$STRING$(5,219)B$
W$B$W$W$B$W$W$B$W$W$;
1060 PRINT W$W$STRING$(7,32)W$B$W$B$W$W$
W$B$STRING$(4,219)B$W$STRING$(5,32)W$B$W
$B$W$B$B$B$W$B$W$W$;
1070 PRINT W$W$B$STRING$(5,219)B$W$B$W$B
$W$STRING$(10,32)W$W$W$B$W$B$W$B$W$B$W$B
$W$B$W$W$;
1080 PRINT W$W$B$W$STRING$(5,32)W$B$W$B$
W$B$B$W$W$W$B$STRING$(7,219)B$W$B$W$B$W$
B$W$B$W$B$W$W$;
1090 PRINT W$W$B$W$B$STRING$(5,219)STRIN
G$(6,32)W$STRING$(5,32)W$STRING$(5,32)W$
B$B$B$W$B$W$B$W$B$W$W$;
1100 PRINT W$W$B$W$STRING$(5,32)W$B$STRI
NG$(6,219)B$W$W$W$B$W$B$STRING$(6,219)B$
W$W$B$W$B$W$B$W$W$;
1110 PRINT W$CHR$(127)B$B$B$W$W$W$STRING
$(10,32)W$W$W$STRING$(17,32)W$W$;
1120 PRINT STRING$(40,219);
1130 PRINT STRING$(40,219);
1140 PRINT W$W$STRING$(17,32)W$W$W$STRIN
G$(10,32)W$W$W$B$B$B$CHR$(127)W$;
1150 PRINT W$W$B$W$B$W$B$W$W$B$STRING$(6
,219)B$W$B$W$W$W$B$STRING$(6,219)B$W$STR
ING$(5,32)W$B$W$W$;
1160 PRINT W$W$B$W$B$W$B$W$B$B$B$W$STRIN
G$(5,32)W$STRING$(5,32)W$STRING$(6,32)ST
RING$(5,219)B$W$B$W$W$;
1170 PRINT W$W$B$W$B$W$B$W$B$W$B$W$B$STR
ING$(7,219)B$W$W$W$B$B$W$B$W$B$W$STRING$
(5,32)W$B$W$W$;
1180 PRINT W$W$B$W$B$W$B$W$B$W$B$W$B$W$W
$W$STRING$(10,32)W$B$W$B$W$B$STRING$(5,2
19)B$W$W$;
1190 PRINT W$W$B$W$B$B$B$W$B$W$B$W$STRIN
G$(5,32)W$B$STRING$(4,219)B$W$W$W$B$W$B$
W$STRING$(7,32)W$W$;
1200 PRINT W$W$B$W$W$B$W$W$B$W$B$STRING$
(5,219)B$W$B$W$STRING$(4,32)W$W$W$B$W$B$
STRING$(7,219)B$W$W$;
1210 PRINT W$CHR$(127)STRING$(19,32)W$W$
STRING$(15,32)W$W$;
1220 PRINT STRING$(40,219);
1250 XY1=1114:XY2=804:POKE XY1,1:POKE XY
2,1:POKE XY1+1,14:POKE XY2+1,14
1260 SC1=0:SC2=0
1290 LOCATE 23,7:COLOR 2,0,0:PRINT"Press
 SPACE to start play"
1300 LOCATE 24,1:COLOR 7,0,0:PRINT"   Pr
ess '1'          Press '2'";
1305 A$="   No Limit":LIM=9.999999E+37:A
1$=" 3 Hands":L=3:L1=3:LOCATE 25,1:PRINT
 A$,"       "A1$;
1310 I$=INKEY$:IF I$="" THEN 1310
1320 IF I$=" " THEN LOCATE 23,1:PRINT SP
ACE$(79);:LOCATE 25,1:PRINT SPACE$(39);:
GOTO 100
1330 IF I$="1" THEN GOSUB 1400
1340 IF I$="2" THEN GOSUB 1500
1350 GOTO 1310
1400 IF A$="   No Limit" THEN A$="  5000
 Limit":LIM=5000:GOTO 1600
1410 IF A$="  5000 Limit" THEN A$=" 1000
0 Limit":LIM=10000:GOTO 1600
1420 IF A$=" 10000 Limit" THEN A$=" 5000
0 Limit":LIM=50000!:GOTO 1600
1430 IF A$=" 50000 Limit" THEN A$=" 1000
00 Limit":LIM=100000!:GOTO 1600
1440 IF A$=" 100000 Limit" THEN A$=" 500
000 Limit":LIM=500000!:GOTO 1600
1450 IF A$=" 500000 Limit" THEN A$="   N
o Limit":LIM=9.999999E+37:GOTO 1600
1500 IF A1$=" 3 Hands" THEN A1$=" 5 Hand
s":L=5:L1=5:POKE 92,L+48:POKE 156,L1+48:
GOTO 1600
1510 IF A1$=" 5 Hands" THEN A1$=" 3 Hand
s":L=3:L1=3:POKE 92,L+48:POKE 156,L1+48:
GOTO 1600
1600 LOCATE 25,1:PRINT A$,"       "A1$;:
RETURN
2000 FOR I=270 TO 250 STEP -2:SOUND I,1:
NEXT
2040 IF C=6 THEN L1=L1-1:SC2=SC2-200:LOC
ATE 2,C:PRINT L1:POKE XY2,32:XY2=804:POK
E XY2,1:POKE XY2+1,14
2050 IF C=38 THEN L=L-1:SC1=SC1-200:LOCA
TE 2,C:PRINT L:POKE XY1,32:XY1=1114:POKE
 XY1,1:POKE XY1+1,14
2060 IF L1=0 OR L=0 THEN LOCATE 2,9:PRIN
T SC2:LOCATE 2,26:PRINT SC1:GOTO 2100
2080 GOTO 100
2100 IF SC1<SC2 THEN C=1:B$="1" ELSE C=3
3:B$="2"
2200 COLOR 23,0:LOCATE 1,C:PRINT"PLAYER
"B$:COLOR 3,0:LOCATE 23,1:PRINT"
Press SPACE to continue"
2210 A$=INKEY$:IF A$=" " THEN 1000 ELSE
2210
5000 DEF SEG:ML$=SPACE$(48):V=VARPTR(ML$
):ZZ=PEEK(V)+256*PEEK(V+1)
5010 FOR I=0 TO 47:READ A:POKE ZZ+I,A:NE
XT:RETURN
10000 DATA &H55, &H1E, &HBE, &H00, &H00,
 &HBB, &H00, &HB8, &H8E, &HDB
10010 DATA &H8A, &H04, &H3C, &HB0, &H75,
 &H04, &HFE, &H04, &HEB, &H0F
10020 DATA &H3C, &HB1, &H75, &H04, &HFE,
 &H04, &HEB, &H07, &H3C, &HB2
10030 DATA &H75, &H03, &HC6, &H04, &H0F,
 &H46, &H46, &HB1, &HFE, &HE0
10040 DATA &H0E, &H72, &HDF, &H1F, &H5D,
 &HCA, &H00, &H00
```

Computing Together

Fred D'Ignazio, Associate Editor



New research suggests that infants are much brighter than we once thought. This research has prompted anxious parents who are worried about their children's ability to cope with a high-tech future, to enroll their infants in computer courses before they are even out of diapers. After class, the parents bring the kids home and drill them using flash cards. On the cards are written words like RAM, ROM, BITS, and BYTES. The parents think that early familiarity with computer technology and jargon will be the youngsters' ticket to a good college and a successful career.

Unfortunately, these parents are teaching their kids skills that may soon be obsolete. After all, it will be the twenty-first century before today's infants enter college or the job market. Between now and then, computers are going to change drastically.

Instead of concentrating on bits and bytes, parents of young children should concentrate on more general skills. They should strive to build a relaxed, comfortable relationship between their children and computers—a constructive relationship that enhances the child's self-image and self-confidence. As the child gets older, this sort of relationship will be more enduring and more valuable than specific skills which may quickly go out of date.

Toddler Burnout

Understandably, parents want their children to do something productive on the computer. For example, they may buy drill-and-practice software that will help give the child a boost in a school subject with which he is struggling.

At first, this approach works well. The child diligently works at the computer and seems to be making progress. But then boredom sets in, the software's novelty fades, and the child loses interest in the computer. The parents' natural reaction is to make the child sit at the computer and continue drilling.

Unfortunately, this approach may lead to *toddler burnout*. For many kids, the joy of computing is replaced by the drudgery of computing. Computers are no longer fun, they are pure work. If kids are "strapped to their computer" every afternoon (as I was once strapped to my piano), they could develop a lifelong negative attitude toward computers and a mental block about using them.

The Computer Playground

We have so many computers around our house that people think we must be a futuristic family. They think that our computers are plugged into everything, including the coffee maker, the thermostat, the bathroom scales, and the toaster oven. They think we live computerized lives.

Fred D'Ignazio is a computer enthusiast and author of several books on computers for young people. His books include Katie and the Computer *(Creative Computing),* Chip Mitchell: The Case of the Stolen Computer Brains *(Dutton/Lodestar),* The Star Wars Question and Answer Book About Computers *(Random House), and* How To Get Intimate With Your Computer (A 10-Step Plan To Conquer Computer Anxiety) *(McGraw-Hill).*

As the father of two young children, Fred has become concerned with introducing the computer to children as a wonderful tool rather than as a forbidding electronic device. His column appears monthly in COMPUTE!.

Such A Deal Inc.

NEW LOW PRICES

Gemini 10X	$267
Legend 80 CPS	$239
Legend 100 CPS	$259
12 In. Amber Monitor	$89
Concord Disk Drive	$297

SUCH-A-STEAL ON SOFTWARE!

Epyx Summer Games	$25
SubLogic Flight Simulator II	$37
Screenplay Pogo Joe	$19
Access Beachhead	$23
Infocom Sorcerer	$33
Continental Home Acct.	$47
Timeworks Word Writer	$39
Timeworks Data Manager II	$39
Commodore Magic Desk	$55
Microware Clone Machine	$39
Blue Sky Super Copy	$29
Handic CalcResult Adv'd	$75

CALL FOR OTHER SUCH-A-STEAL PRICES ON SOFTWARE AND HARDWARE FOR YOUR COMMODORE

CALL TOLL FREE
1-800-431-8697
For Customer Service
Call: 602-957-3619

ORDERING & TERMS: Send cashier check, money order, personal/company checks allow 3 weeks bank clearance VISA/MasterCard accepted. Provide phone number with order. SHIPPING: Software add $4.00 for first three pieces, add $1.00 each additional piece. Hardware add $10.00. Returns must have authorization number (call 602-957-3619 for authorization number). All returned merchandise subject to restocking fee and must come with all original packaging. No returns allowed after 30 days from shipping date. Prices are for cash, VISA and MasterCard add 3%. Prices subject to change without notice. All products subject to availability from manufacturers and/or suppliers. All prices in U.S. dollars.

SAVE MORE THAN EVER ON...

3M Scotch® DISKETTES

AND OTHER COMPUTER NEEDS!

NEW! Lower Prices

LIFETIME WARRANTY!

3M BULK DISKETTES AT TREMENDOUS SAVINGS!

These are *genuine* 3M diskettes with a lifetime warranty. But they are bulk packed in cartons of 50 with separate white Tyvec envelopes. **No identification labels, write protect tabs or cartons are provided!** A great buy for volume users.

$160 ea. Qty. 50 ← 5¼" SSDD

5¼" DSDD → **$202** ea. Qty. 50

5¼" SSDD-96TPI $2.20

5¼" DSDD-96TPI $2.75

All have reinforced hub.
SOFT SECTOR ONLY!
(for IBM, APPLE, KAYPRO, DEC and about 99% of all computers.)
Must be ordered in multiples of 50!

BOXED 3M DISKETTES WITH ALL THE TRIMMINGS!

Factory-fresh 3M packaging with envelopes, 3M logo labels, ID labels and write-protect tabs.

$185 ea. Qty. 20 ← 5¼" SSDD

5¼" DSDD → **$235** ea. Qty. 20

5¼" SSDD-96TPI	$2.60	8" SSSD	$2.05
5¼" DSDD-96TPI	$3.25	8" SSDD	$2.50
		8" DSDD	$3.10

Minimum order of 20 diskettes. Additional diskettes in multiples of 10.

3M HEADCLEANING KITS

Stop swearing and start cleaning. This non-abrasive cleaning kit has everything you need for thirty applications.

$23.00 + $1.50 Shpng.

SAVE MONEY WITH A CLEAN COMPUTER!
INTRODUCING MINI-VAC

Most computer malfunctions are caused by dust. MINI-VAC is ideal for cleaning keyboards, screens, drives and printers. (Great for photo equipment, too!) Equipped with an easy-empty bag, two directional wands and two fine-brush nozzles. Don't compute without it. (Requires 9-volt battery which is not included.)

$21.95 + $3.00 Shpng.

AT LAST: A DISK DRIVE DIAGNOSTICS SYSTEM THAT WORKS!

The Dymek Recording Interchange Diagnostic (RID) is a **professional**, but easy to use, drive diagnostic disk. It tests drive speed, radial position, hysteresis, write function, erase crosstalk, signal-to-noise and clamping. In short, it's a professional's system that will help you keep your machine in prime condition...and avoid the evils of data loss.

$29.95 + $1.50 Shpng.

MEDIA-MATE 50: A REVOLUTION IN DISKETTE STORAGE

Every once in a while, someone takes the simple...and makes it elegant. This unit holds 50 5¼" diskettes, has grooves for easy stacking, nipples to keep diskettes from slipping in the case and several other features. We like it.

$10.95 + $2.00 Shpng.

DISKETTE 70 STORAGE: STILL A GREAT BUY

Dust-free storage for 70 5¼" diskettes. Six dividers included. An excellent value.

$14.95 + $3.00 Shpng.

PRINTER RIBBONS AT BARGAIN PRICES.

EPSON MX-70/80	$3.58 + .25 Shpng.
EPSON MX-100	$6.99 + .25 Shpng.
Okidata Micro 84	$3.66 + .25 Shpng.
Diablo 630 Mylar	$2.60 + .25 Shpng.
Diablo 630 Nylon	$2.93 + .25 Shpng.

THE END TO RS-232 CABLE PROBLEMS: SMARTCABLE

Now interfacing almost any two RS-232 devices is simple and quick. Just plug in SMARTCABLE and flip two switches. The logic of both devices is figured out immediately and you can get to work.

$79.95 + $1.50 Shpng.

Shipping: 5¼" DISKETTES—Add $3.00 per 100 or fewer diskettes. 8" DISKETTES—Add $4.00 per 100 or fewer diskettes. OTHER ITEMS: Add shipping charges as shown in addition to diskette shipping charges. Payment: VISA and Mastercard accepted. COD orders only, add $3.00 handling charges. Taxes: Illinois residents, please add 8% sales tax.

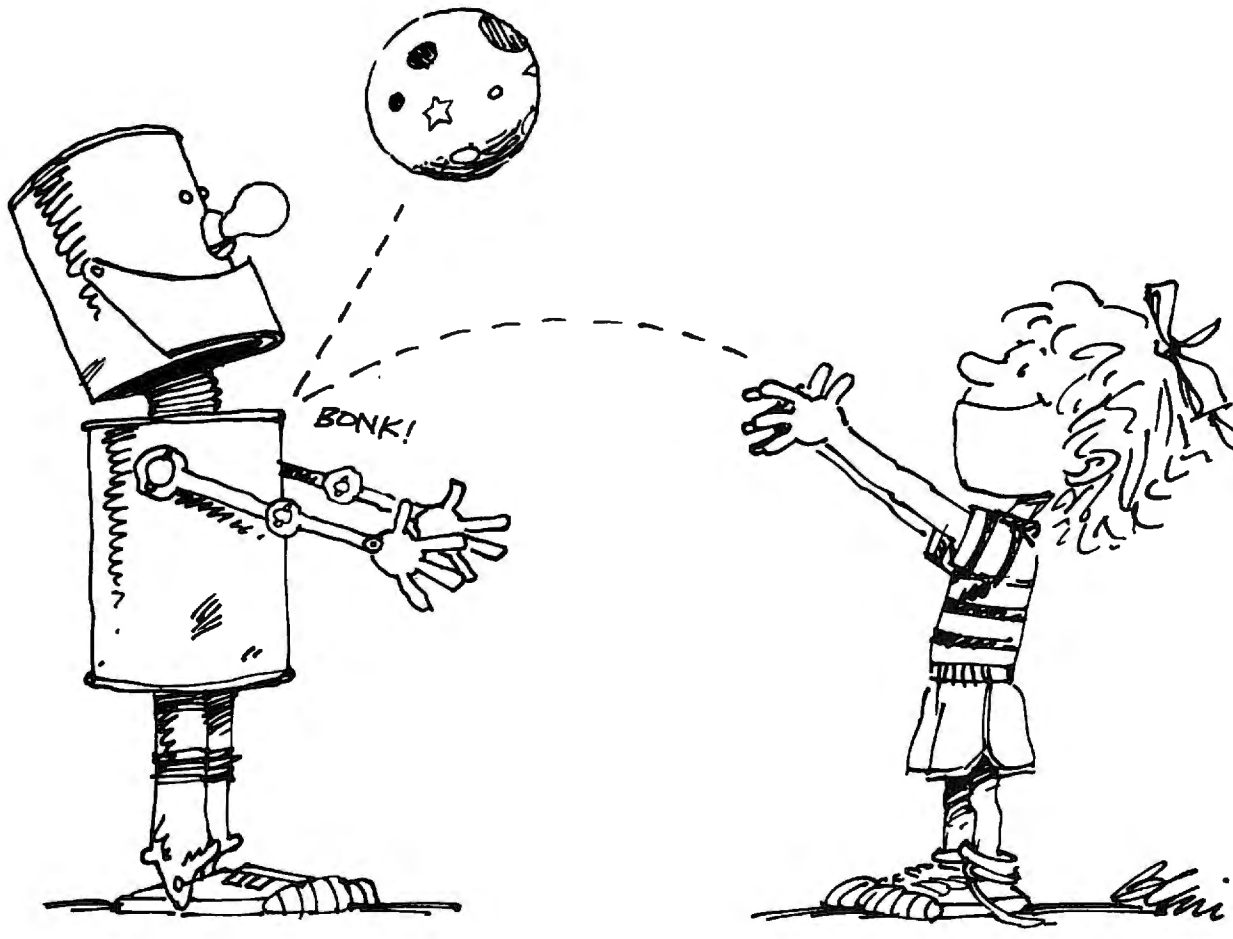
WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

Nationwide: 1-800-621-6827
Illinois: 1-312-944-2788
Hours: 9AM - 5PM Central Time
Minimum Order: $35.00

DISK WORLD!, Inc.
SUITE 4806
30 EAST HURON STREET
CHICAGO, ILLINOIS 60611

Authorized Distributor
Information Processing Products

3M



Nothing could be further from the truth. When I get the chance to tell people what really goes on, I say that we have an Erma Bombeck household. Sure we use computers, but not to make our lives more rigid, organized, and mechanical. Instead, we use them as an electronic *playground*—and not just for Catie and Eric, but for me and my wife, Jan, too.

When people ask me what kind of software we buy for the kids, I say that we buy the software that turns *us* on. Then when the kids see us using the computer and having a good time, it gets them excited, too.

At my house we don't think of play as trivial. To us, play is a product of love. If we love to do something, it isn't work, it is play.

I would like my children to love to use computers, to use them playfully and creatively. I never want my kids to feel that computers are chains tying them to a hateful task. Instead, I want them to see computers as wings that enable them to swoop, dive, and have fun, and take them to new heights and soar to the limits of their abilities and imaginations.

The Computer As A Babysitter

The computer makes a great babysitter—even better than TV (unless you have cable, a VCR, and lots of tapes). It will soon be a big temptation for parents to turn on the computer to get their little kids out of their hair.

The computer can make a healthful babysitter—to a point. It can provide a much-needed break for a harried parent. And it can become a child's companion and a patient teacher. Also, *flying solo* on a computer can be a very positive experience for a child. It can give them a sense of control, mastery, and responsibility that they seldom experience at such a young age.

However, it is easy for little kids to get too much of a good thing. More than computing they need time to play with other children, get lots of exercise, fresh air, and experience the joy of swinging, digging in sand, and getting elbow-deep in finger paints.

Most important of all, they need to spend time with their parents. Computers make great toys, but they cannot replace parents. Parents are children's first and most important toys. Computers make a very poor substitute.

New games are starting to appear (including many programs from Children's Television Workshop, Spinnaker, Sunburst, and Counterpoint Software) that encourage parents and children to play on the computer *together*. Then the computer changes from being a babysitter that isolates the child to an *electronic hearth* that brings the whole family together actively and

happily. In fact, studies at New York University suggest that computers *encourage* families to spend more time together.

Computer Elevator Shoes

Computers are like booster shoes. They can give handicapped people a boost so they can go about their lives on par with the rest of the world. Computers can also play this role with young children.

My children are always at the bottom of the family totem pole, *except when they use the computer.* I encourage five-year-old Eric and eight-year-old Catie to do things on the computer that enhance their abilities, that increase their self-respect and self-confidence, and give them a leg up on the rest of us. Here are some of the things our kids do on their computers:

• Gobbledygook Processing. Five-year-old Eric bangs on the keys of the computer and gets it to print out page after page of gobbledygook. Eric is learning how to type, he thinks he is doing work, and he takes his gobbledygook to school and sends it to both his grandmothers. Remarkably, the gobbledygook is gradually starting to make sense. Real words, phrases, and sentences are starting to appear. Most important of all, Eric is developing the habit of using the computer as a *tool* to help him think better and not as a crutch to do his thinking for him.

•Training The Family Pet. Catie and Eric treat our computers like pets. Sometimes they pull their tails, but mostly they are learning "computer manners"—how to treat the computers kindly and responsibly. They can turn on all the computers, use the floppy disks and cartridges, and call up all their favorite programs. Eric, for example, is so good that when I hired a housekeeper and a secretary, he taught them how to use the family computers.

• Computer Scribbling. Catie and Eric have a skill that Janet and I have lost: They can scribble! When we turned Catie and Eric loose on a computer touch tablet—like the PowerPad from Chalk Board and the KoalaPad from Koala Technologies—it was incredible. The tablets enhanced the kids' motor skills, allowed them to make fine, detailed changes to their drawings and pictures, and gave them the freedom to creatively scribble. We now have a slideshow of the children's computer pictures and a door full of their drawings on the new Macintosh computer.

• The Computer Sandbox. The children play games on the computer that give them the most control. *They* control the computer, rather than the other way around. One of the children's most popular games is to play on the keyboard, pushing buttons just to see what happens. They call this "Flying the Cursor." Doing this they have discovered how to get the computer to make moving rainbows, colorful letters, upside-down letters, pictures, and sounds—all without writing or buying a single program.

• The Electronic Picturebook. The kids have both learned how to read by using computer adventure games for young children like Sierra On-Line's *Troll's Tale* and *Dragon's Keep.* They enter the microworlds inside the computer and instantly become the heroes at center stage. To journey through the world they have to remember where they are, and read the signs in the pictures and the messages at the bottom of the screens. In these games, words gain real meaning and power. They are the keys Catie and Eric use to outwit an ugly troll or rescue small animals from a mean dragon.

Robots: Bag Ladies And Alarm Clocks

We have lots of computers around the house, but we also have robots. In fact, we run a flophouse for robots. We never know when a robot will come to our door looking for a home. Then I write an article about the robot, and, pretty soon, we have to send the robot along to another writer so they can write about the robot, too.

My children love robots—not as servants, but as pets. When TOPO the robot came to visit us, for example, my children noticed that TOPO was naked and dressed it in various costumes. My son tied his blanket to TOPO and turned it into a superhero. My daughter dressed TOPO as a New York bag lady, as a little girl, and as a witch, complete with a long pointed hat, a black cape, and vampire teeth.

TOPO never washed any dishes, made any beds, or took out the trash, but it was still useful. Every school morning, I turned TOPO on and sent it into the children's bedrooms to wake them up. When Jan and I wake up the kids they growl, whine, and complain. But when TOPO appeared, did a silly jig, and said, "Wake up, sleepyheads. Time to get out of bed," the kids got up smiling and gave the robot a hug.

Robots may never be good as maids or butlers, but they make great pets and alarm clocks. C

Richard Mansfield, Senior Editor

Trapping Bugs

It was a moth, according to legend, that caused a program to crash in the early days when computers were built of vacuum tubes and tons of copper wire. The critter had flown into the machine. From this we get the term *bug*, meaning that there is an error, a problem in a computer program. And tracking down bugs is called *debugging*.

As all programmers soon learn, there is no permanent cure for bugs—they are always hiding inside a freshly written program of any complexity. Some bugs are obvious and will show up the first time a program is tried out. Some are hidden away and permit most of the program to run without error. A complex program might run well for weeks or months and then a particular sequence of events will trigger a well-hidden bug.

Program Sketches

For many, programming is similar to painting or sculpting. First you jump in and roughly create the outlines, the main ideas. At this point you've essentially made a sketch of the final program. Then you start testing the program by RUNning it, refining it until it performs as it should.

What are the best ways to look for bugs? Luckily, the most common bugs, typos, are reported to you by BASIC itself. On the Atari, if you try to enter a line like this: PRINF X, you will get an immediate SYNTAX ERROR report. Other versions of BASIC wait to report typos until after you RUN the program, but the effect is the same. Your computer tells you what's wrong and which line to fix.

Many other bugs show up quickly when you first try out the program: Nothing appears onscreen; things appear, but in the wrong places; or the numbers are all wrong. In other words, the program isn't even coming close to your expectations. These are often easy bugs to work with because they aren't usually caused by the interaction of two parts of your program. There's some gross failure somewhere. You've simply got to look at your formatting routine or your mathematical definitions to see where the problem is.

Between The Cracks

Some of the hardest bugs to find are hidden in the cracks. They are usually the result of a clash between two otherwise perfectly functional subroutines. For example, if your program uses the variable *T* to stand for the total of an addition problem and then you use a subroutine with a loop that also uses T:

```
10 T = BOLTS + WASHERS
.
.
.
800 FOR T = 1 TO 500
```

As you can see, no matter what your total of bolts and washers is, it will be left at 500 anytime you use the subroutine at line 800.

A similar interaction between variables can be even more subtle. In many versions of BASIC, only the first two letters of a variable name have any significance. So, if you name one thing BOLTS and another thing BOWLING, these two things will appear to the computer as a single variable called *BO*. And, as in the example above, the most recent number assigned to BO will be the *only* value that variable can have.

The Worst Bugs

But the worst bugs are not in the computer at all. They're in the programmer's mind. And since you must use your brain to ferret out the errors

caused by that brain—you can see the paradox. These errors tend to be of two types: incorrect setups and bad logic.

An example of an incorrect setup would be thinking you've defined a variable when, in fact, you haven't, or using > when you mean <. The variations on this theme are endless and you can look at > dozens of times and not even stop to think about it as a possible source of error.

Bad logic would include such things as subroutines which exit via GOTO instead of RETURN; INPUT at the wrong time; or forgetting about the first or last item in a sequence like a DATA list.

Sometimes there's only one way to find a deeply hidden bug: stepping through the program. There are two levels of step testing. You can insert STOP in various places, then check to see that the variables are what they should be at these stopping points. Then CONT to the next STOP and ask to have the variables printed again (type: ? X,Y,Z$). This rough test is often enough to pinpoint the place where the program has gone wrong.

Alternatively, you can use the single-stepping TRACE function found in many programmer's aid programs. These aids add commands to BASIC like RENUMBER, DELETE, and usually have a single-stepping function as well.

When you activate a TRACE command, your program executes step by step, one command at a time. After each command, the status of all active variables is displayed on screen along with the program line so you can locate where things begin to come unglued. Often, a TRACE function permits you to define how fast it will execute and even allows you to turn it on or off from within the program. TRACEing is a slow, but nearly always successful way to trap the most devious bugs.

If all else fails, it's sometimes advisable to ask for help from a friend. His brain won't have been implicated in the original error, and he can therefore often spot the > you keep ignoring. ©

Computers And Society

David D. Thornburg, Associate Editor

Technostress

Craig Brod is a psychotherapist who is seeing a disturbing trend in his patients. He is seeing a new malaise that he calls *technostress.* His concern over this ailment was apparently so severe that he felt obliged to write a book about it (*Technostress: The Human Cost of the Computer Revolution,* Addison-Wesley, $16.95). Whether technostress is a serious malady is not for me to ponder, but there is little question in my mind that Dr. Brod's book will induce stress in many of its computer-literate readers.

According to Dr. Brod, our societal fabric is being reshaped as a result of our headlong push into the information age. It is astounding to see the nature of the ailments that appear to be caused by the mere use of computers in society:

> The wife of a director of computer services for a large bank reports that when she first met her husband, he was a warm and sensitive man. Today he has no close friends and his only recreational activity is watching television. He no longer has patience for the easy exchange of informal conversation. One night, she asked him to slow down as they walked home.
>
> "Walk faster," he replied.
>
> "I *can't* walk faster. My legs are shorter than yours."
>
> "That's no excuse," he said. "You have to learn to walk more *efficiently.*"

But Is It Pervasive?

Now, seriously, folks, this man has problems—but I can't believe that the computer is the cause of them. The fact is that, as a therapist, Dr. Brod is more likely than the rest of us to encounter people who are having difficulty adapting to change—and we are definitely going through a period of intense change. Nonetheless, rather than suggesting that he is observing the aberrant behavior of a minority, Dr. Brod goes so far as to suggest that we are *all* potential victims of the onslaught of computer technology.

As one example of this, he suggests that the reason we as a nation have purchased so many computers is because we fear them:

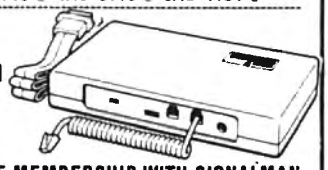
> Ironically, we are motivated by fear to accept what is supposed to bring security and hope. Workers and managers fear obsolescence if they are not at the technological forefront. Parents, concerned about the demands of future educators and employers, feel compelled to make sure their children are computer-literate at an early age. Those who do not join the revolution will, we are told, become relics of a backward culture.

I seriously question whether any readers of this magazine felt that society compelled them to buy a computer. If it is considered a "negative sell" to encourage people to become facile with the tools of their future, then so be it.

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. His recent series Computer Art and Animation *(Addison-Wesley) includes four books on Logo for the Atari, Commodore, Radio Shack, and TI computers.* Discovering Apple Logo *(Addison-Wesley) shows how Logo can be used as a tool for exploring the art and pattern of nature. He has been called "an enthusiastic advocate for a humanistic computer revolution," and his editorial opinions have appeared in* COMPUTE! *since its inception.*

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 @

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 Model 1200/300	279
DC HAYES Smartmodem	**219**
DC Hayes Smartmodem 1200/300	519

PROM QUEEN for C64 or VIC	**130**
SM-KIT 64 program & disk utilities	**45**
STAT Statistics Package for C64	**95**
Solid Oak 2 Level Stand for C64 or VIC	**29**
C64/VIC Switch (networking)	129
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	79
CARD PRINT B Printer Interface—C64/VIC	48
CARDBOARD/3s Motherboard - VIC	22
CARDCO C64/VIC Calculator Keypad	32
CARDRAM/16 RAM Expansion - VIC	49
Complete CARDCO Line in stock	
CIE and VIE IEEE Interfaces in stock	
MSD Dual SuperDisk for C64 or IEEE	**570**
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	19
COMAL Language for C64	**14**
with sprites, color graphics, sound, turtle graphics.	
BusCard II from Batteries Included	**159**
ULTRA BASIC – 64 with Turtle Graphics	37
Super Disk Utility – C64 – includes backup	19
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	**40**
Kraft Apple Paddle Pair	30
Koala Pad Touch Tablet-Apple/Franklin	**90**
SPINNAKER Software in stock	
Broderbund Software in stock	
16K RAM Card for Apple	59
Multiplan—Microsoft	**179**
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
AB 80 Column Card with Softswitch	**95**
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

MSD Dual Super Disk for IEEE or C64	**570**
replaces 4040 drive	
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**

VICTORY Software for VIC and C64

Metamorphosis	16	Creator's Revenge	16
Labyrinth of Creator	16	Galactic Conquest	16
Kongo Kong	16	Annihilator	16
Chomper Man	16	Grave Robbers	13
Bounty Hunter	16	Adventure Pack I or II	16

PAPER CLIP Word Processor – CBM/C64	**60**
ORACLE Data Base from Batteries Included	**89**
SPINNAKER Software C64, Apple, IBM, Atari	
Compute!'s First Book of PET/CBM	11
POWER ROM Utilities for PET/CBM	78
WordPro 4+ - 8032, disk, printer	285
VISICALC for PET, ATARI, or Apple	189
Compute's First Book of 64 Sound & Graphics	**11**
SM-KIT enhanced PET/CBM ROM Utilities	**40**
PET Spacemaker II ROM Switch	36
Compute's First Book of Games	11
Dust Cover for PET, CBM, 4040, or 8050	8
CmC Interfaces (ADA1800, ADA1450, SADI in stock)	
Compute!'s Reference Guide to 64 Graphics	**11**
Compute!'s Machine Language for Beginners	11
HES Software and Hardware 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.	
Metacompiler for FORTH for independent object code	**30**
Floating Point for FORTH	**20**
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	**59**
SuperGraphics – BASIC Language Extensions	**45**
Fast graphics, sound, turtle graphics routines for PET/CBM.	
RAM/ROM for PET/CBM 4K $75 8K $90	
COMAL Language for C64, CBM, PET	**14**
Smart Terminal Software for C64/VIC	**10**
CBM Public Domain Software – C64 27 disks	**75**

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	**229**
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.

I, for one, see things in a much more positive light.

The Best Tool Available

The fact of the matter is that I use computers for several reasons. I use a word processor for all my correspondence, books, and articles because it is, quite simply, the best tool available for the job. The fact that some authors can whip out books in front of an old Underwood is fine with me—but I'm not one of those authors. Dr. Brod suggests that authors who use word processors do not produce as finely crafted works as those who have to completely retype their early drafts.

I think he is confusing technology with writing style. Yes, it is easier to change a line or paragraph with a word processor—one doesn't have to retype an entire page. But if I have written something that just doesn't hang together, I rewrite it from scratch, and so do many other authors with whom I have talked. In fact, I have found that college students often hand in essays that they would really like to change, simply because the time associated with retyping the complete document is prohibitively long. The presence of a word processor may, in fact, make better writers of all of us.

Parents are right to be interested in their children's education, and they are right to realize that the computer can be an important educational tool. But to suggest that the success of Apple and Commodore and IBM has arisen out of a fear in the buying public is (and this is *not* a medical opinion on my part) pure hogwash.

Yes, guilt has been used to sell computers, especially in the late 1970s ("Make an investment in your child's future—buy a computer"), but this isn't any different from the time-honored approach for selling encyclopedias.

True, It's Not Perfect

Now I know that some of you must be saying that the computer revolution isn't all it's cracked up to be, that computers are frustrating to use, that they cause disruptions in offices when they are installed, and that computers and robots are likely to change the very nature of our workforce—especially in the blue and pink collar areas.

You are right, of course. Many of us use computers in spite of their poor user interfaces (even though these are improving all the time), and, yes, a lot of jobs are going to disappear in the near future. But, after all, we have been down this road before in our history. Many buggy whip manufacturers must have either changed their business or gone bankrupt when the automobile replaced the horse.

The computer will be no more or less traumatic in its impact on society. Dr. Brod is quick to point out, of course, that the convenience of automobiles has brought with it 50,000 annual deaths on our nation's highways, and some severe smog problems as well.

I would be the last to suggest that the automobile has been an unqualified blessing, but I can't imagine our culture surviving, let alone growing, if we went back to the horse and buggy.

A Heavier Workload

One of Dr. Brod's points is that many of the labor-saving aspects of computers haven't been realized by the people who are using them. Secretaries who, according to the word processor ads, should now have time for "that extra cup of coffee" are finding that their ability to generate letter-perfect documents is increasing their workload as their managers ask for work to be redone until it is perfect. Organizations that functioned adequately when financial statements were generated quarterly are now using electronic spreadsheets to do financial statements on a weekly basis, thus increasing the workload for that department.

Dr. Brod is correct in assuming that more isn't necessarily better, but one must ask how much the computer contributed to the increased flow of information. From my own experiences inside Fortune 500 companies, the information backlog has been there all the time—all the computer is doing is helping to handle a preexisting problem.

As I mentioned a few months ago, John Naisbitt's concept of high-tech/high-touch (as expressed in his book *Megatrends*) showed that as we became more involved with the use of technology in our work and play, we have also become more interested in those things that make us uniquely human. Dr. Brod claims that the reverse is true—that the computer world is symbol-intensive, not sensual, and that this constricts us in our ability to interpret and create in a human way.

If this is the case, then how was Michelangelo able to create such a sensitive work of art as the *Pietà* using such cold and inhuman tools as the hammer and chisel? Once again, I feel that Dr. Brod has confused the technology with its use. The two are quite different from each other.

The Computer As Scapegoat

And yet, in chapter after chapter, we find that computer technology is the purported cause of much that is "wrong" with our society. His chapter on computers and kids, for example, is filled with the typical hand-wringing about videogames that has appeared in all the tabloids. This is surprising, considering that many of his col-

leagues who have actually researched the matter find that video arcades do *not* cause perversion, or even acne.

After devoting several chapters to his observations of the purported ills foisted upon us by computers, Dr. Brod does give some careful thought to ways to make us capable of handling this technology, even though many of us never knew we had any problems.

Technostress is a book filled with quotable material, and it will probably be heralded as an important book by technophobes everywhere. It will probably receive a lot of press, and its author will probably be in great demand as a speaker, as he carries his message to the world. After all, as a society, we always seem to favor the bad news over the good, and seem to devote our energies to looking for only the real or imagined wrongs in our world.

If Dr. Brod wanted to perform a service to mankind, he might have devoted his energies to solving *this* problem, rather than extrapolating the quirks of his patients to the rest of society. ©

Glenn M. Kleiman

The Computer Speaks, But Will It Listen?

Computer-generated speech, already used in some software, will be incorporated into many educational programs in the next few years. Spoken instructions and responses will be used in programs designed for prereading children and for students who have reading difficulties. Speech will be an integral component of programs which help students learn reading, spelling, and foreign languages, and will make many other types of educational programs more interesting and enjoyable.

Computerized speech can open new worlds for handicapped people. Special programs enable blind users to direct a speech synthesizer to read aloud the words on the computer screen. This makes computerized information bases, word processing, programming languages, and many other computer tools available to the blind. Computerized speech can also help provide communication aids for people with speech impairments.

Computerized speech *recognition* devices are also becoming less expensive and more readily available. These enable computers to recognize words people say, and can make programs easier to use and more appealing. More importantly, speech recognition devices make computers accessible to many people who have physical handicaps which prohibit them from using keyboards.

Two Types Of Computer Speech

There are two general types of computer-generated speech: *stored vocabulary* and *unlimited vocabulary*.

Dr. Glenn M. Kleiman is an educational psychologist and software developer. He is the author of Brave New Schools: How Computers Can Change Education *(Reston/Prentice-Hall) and the designer of* Square Pairs, *an educational game program (Scholastic, Inc.).*

Stored vocabulary speech is created by a person saying the words. Special devices and programs measure characteristics of the sound waveform (for example, intensity, pitch) as the person pronounces each word. Numbers representing the waveform at each fraction of a second are stored in the computer. That is, the speech waveform (an example of what is called *analogue* information) is converted to a sequence of numbers (*digitized* information). The numbers are then used to recreate the sound of the word whenever it is needed.

Stored vocabulary speech can sound very human when individual words are produced. However, it usually sounds choppy and somewhat artificial when the words are combined into sentences. With this technique, the computer is limited to the words previously stored in its memory.

Each digitized word requires a large amount of memory—many numbers must be stored for the computer to recreate the spoken words clearly—so the vocabulary of a personal computer with digitized speech is limited. However, the possibilities for digitized speech will expand as larger-capacity computer memories become less expensive, and as more efficient techniques are developed for representing speech waveforms within the computer's memory.

Unlimited Vocabulary

With unlimited vocabulary speech, programs for generating the individual speech sound (phonemes) are stored in the computer, along with the rules for combining them into words, phrases, and sentences. This technique of speech synthesis enables the computer to produce any word from its component sounds. Synthesized speech does not sound as natural as digitized speech, but it has been greatly improved in recent years.

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
Intec 48K (400) $89
Mosaic 48K (400) $98
Intec 64K (400) $105
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) $15
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) $65
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) $23
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
Wico Trackball $36
Compuserve $26
Covers $7
Maxells $25
Memorex $24
Elephants $18
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 $398
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

HAYES
Smartmodem $209
Smartmodem 1200 .. $498
Micromodem II $259
Micromodem IIe $239
NOVATION
J-Cat $99
Apple Cat II $259
D-Cat $149

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 $138
Video Pak 80 ..... $129
Z80 Video Pak $209

RECORDERS
Cardco Recorder $48
1530 Commodore ... Call
Cassette Interface ... $29

INTERFACES
The Connection $85
Bus Card $138
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

DIRECT MODEMS
Hesmodem $53
1650 Automodem $99
1600 Modem ..... Call

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) $13
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) $17
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) ... $39
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) $20
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

Phoneme synthesis techniques have been combined with *text-to-speech conversion* programs. These programs contain a set of rules which tell the computer how to change any sequence of letters into speech. Creating a program of this sort for English is difficult, since many letters and letter patterns are pronounced in various ways, depending on the context of their use. For example, the word *read* is pronounced differently depending upon whether it refers to the past or future (for example, *John read the book* versus *John will read the book*). The same aspects of English which cause difficulties for people in learning to read also cause difficulties in programming computers to convert written English to spoken English.

While text-to-speech programs do not produce human-sounding speech, most people understand it easily after a short time—much the way we can understand someone who has a foreign accent and mispronounces some words. Text-to-speech is valuable for people with impaired vision. However, it is not suitable for educational applications in which clear speech is essential.

A Talking Apple

The Echo II speech synthesizer, for Apple II computers, makes use of both stored and unlimited vocabulary techniques. The Echo II is a board that plugs into a slot in the Apple. A speaker or headphone then plugs into the board. The board has volume and pitch controls, but these can also be controlled from software. The basis of the Echo II is a speech synthesis chip made by Texas Instruments. This chip, an advanced version of the one used in the original Speak and Spell toy, is used in most of the available speech synthesizers.

The Echo II comes with a text-to-speech program. It also allows you to enter speech more directly by using symbols to represent each sound (for example, there are different symbols for the long *e* of *Pete* and the short *e* of *bet*). In addition, a disk containing 700 digitized words is available. These provide a good demonstration of the superior quality of digitized speech.

With the Echo II, it is easy to add speech to your own program. You can change the volume, pitch, and rate of speech, all under the control of your program. Produced by Street Electronics, the Echo II sells for about $150. Street Electronics also produces speech synthesizers for the IBM PC and for other personal computers. Other speech synthesizers are available, including Type-'N-Talk from Votrax, Mockingboard from Sweet Micro Systems, and *S.A.M.* from Don't Ask Computer Software.

Computers That Listen

A great deal of research has been devoted to getting computers to recognize people's speech. This research has shown that speech is very complex and that we do not fully understand how people are able to recognize spoken words so easily. It is much more difficult to make computers recognize spoken words than it is to make them pronounce words. However, advances have been made and some usable, although limited, devices are now available.

Current systems for personal computers require the user to program the computer to distinguish among a number of spoken words. The technique is related to stored vocabulary speech. The individual selects a vocabulary to be used. He says each word, then the computer digitizes the sound patterns and stores a set of numbers representing the waveform of the word.

Once trained, the computer recognizes a spoken word by digitizing it and comparing the resulting pattern of numbers to the patterns stored in its memory. Since the pronunciation changes slightly each time an individual says a word, exact matches are not expected, but the computer is programmed to find the closest match. Since people differ widely in their speech patterns, these systems are reliable only in recognizing the words spoken by the person who spoke the original training set.

The digitized representation of each word uses up a lot of computer memory, and the matching process becomes progressively slower and less reliable as more words are added. Therefore, speech recognition systems work well only with limited vocabularies.

It Takes Dictation

One speech recognition device is the Voice Entry Terminal (VET-2), produced by Scott Instruments for Apple II computers. The VET-2 can be programmed for sets of up to 40 words. The Apple II can hold only one set in memory at a time, but others can be loaded from disk as needed.

One important characteristic of the VET-2 is that it functions as a keyboard emulator. It plugs into the computer in parallel with the keyboard, so both can be used together. Each spoken word is associated with a string of printed characters.

When the spoken word is recognized, the VET-2 sends the same signals to the computer that the keyboard sends when the associated keys are pressed. Therefore, you can have the VET-2 recognize a spoken name for each key and then "type" by saying the names of letters, numbers, and special characters. You can then use standard software with voice input replacing the keyboard.

What About Language?

Current technology for personal computers enables us to have computers speak and recognize individual words. But what about sentences and paragraphs? For speech production, we can have the computer string words together, but replicating the intonation and stress patterns of human voices is another, much more difficult, matter.

For speech recognition, anything more complex than the simplest sentence creates inordinate difficulties. Try listening to fluent speakers of a language you do not understand. Can you even tell where one word ends and the next begins? Recognizing the words in spoken sentences generally depends upon being able to understand meanings, something we have not yet learned to program personal computers to do.

Getting computers to produce and understand language is the focus of much of the effort of researchers in artificial intelligence. They have had only limited success, with very powerful computers. For the present, we will have to be content with personal computers which are at the single-word state of language development.

Street Electronics (Echo II)
1140 Mark Ave.
Carpinteria, CA 93013

Sweet Micro Systems (Mockingboard)
Cranston, RI 02910

Votrax (Type-'N-Talk)
500 Stephenson Highway
Troy, MI 48084

Don't Ask Computer Software (Software Automatic Mouth)
2265 Westwood Blvd
Los Angeles, CA 90064

Scott Instruments (Voice-Entry-Terminal)
1111 Willow Springs Drive
Denton, TX 76201

INSIGHT: Atari

Bill Wilkinson

This month we'll conclude our exploration of the source code of the program to load a binary file starting with the GET routine presented last month.

Lines 600–619. GET is a special routine for two reasons. First, it assumes a buffer length of zero, thus forcing a single byte transfer into the A register (Atari I/O spec). Second, if the GET fails, it pops a level off the subroutine stack and goes directly to the end-of-file code at line 4000 (BASIC's line 400). This is a crude but effective simulation of the TRAP 400 code in the BASIC version.

For GET to be a general-purpose subroutine, it would have to simply return the status and let the caller do the error trapping.

The Main Section

This routine begins the real work. All object code is reasonably close to its BASIC parallel.

Lines 1200–1204, 1300–1304. Remember the calling requirements for the I/O routines? Channel in X, address in A and Y. Looks easy once you have built the subroutines.

Lines 1400–1407. Same as above. The only extra here is the need to specify a mode for OPEN. Here, we use mode 4 (just as in BASIC) to indicate we will only read the file.

Lines 2400–2405. Since we just stored the A register in HIGH, we test HIGH first by comparing A with 255. If HIGH is equal to 255, then A contains 255 and we can compare it to LOW. A tiny bit sneaky. Did you ever realize that BASIC has to implement THEN this way? By branching around the following code?

Lines 2600–2701. We used LOW and HIGH to get the START address, but we have already moved their contents to START. Now, they won't be used again until we are through with QUIT, so why not share memory between QUIT, LOW, and HIGH? Again, a little bit sneaky, but not inordinately so.

We could have saved more memory (and code) by doing GETs into the low and high bytes of START directly, but I wanted to keep the code as close as reasonable to the original BASIC.

Lines 3100–3106. See the comments above about START and ADDR.

The FOR Loop

Lines 3300–3302. Remember, if a zero page location points to a desired memory location, use an offset of zero in the Y register to store, load, add, etc., to or from that location.

Lines 3403–3408. Since we are STEPping by one, we need check only for equality.

Lines 3411–3417. If the FOR loop had used a STEP, we would have had to add it on here. Since the step is implied to be one, we can use this simple two-byte increment.

Line 4103. If this routine is called from DOS or from BASIC, the RTS is all that is needed, thanks to the POP in the GET routine.

As I said, one could write this routine in better ways. The most obvious thought that comes to mind is to replace the FOR loop with a block get of the requisite bytes. Since that would produce significantly faster runtime (for large files, at least), we will make these changes next month.

To do so, though, we will also change the BASIC program to enable it to make a call to do block I/O. So, even if you are not into machine language, watch next month for a method of doing fast memory reads and writes to and from disk.

Load A Binary Object File—Program Completed

```
0660                0990 BEGINWORK
                    0991 ;
                    1000 ; BASIC: REM binary object file loader
                    1001 ; ---- just a comment ----
                    1100 ; BASIC: DIM NAME$(30)
                    1101 ;  (the NAME buffer is defined above)
                    1200 ; BASIC: PRINT "WHAT FILE TO LOAD ";
```

ATARI

ATARI is a trademark of ATARI, INC.

ATARI 800XL . Call
Atari 1050 Disk Drive . . . . . Call
Atari 1010 Recorder 77.00
Atari 1027 Printer . Call
Atari 850 Interface . . . . . Call

MOSAIC
64K Ram/400 . . . . . 149.00
64K Ram + Cable Kit 400/800 . . . . . 169.00
48K Ram Kit 94.00
16/32 Expander 64.95
32K Ram 77.95
Mosaic Adaptor 49.95

Replacement Keyboard for Atari 400 . 54.95

3rd Party Printer Interfaces
Interfast I 139.00
Apeface w/cable included 69.95

PRINTERS

Axiom AT-100 . . . . 229.00	Gemini 10X . . . . . Call
Axiom AT-550 . . . . 329.00	Alphacom . . . . . Call
Epson . . . . . Call	Okidata 92A . . . . . Call

Call for assorted Printer Packages

DISK DRIVES
Rana 1000 . . . . . Call
Trak . . . . . For
Concorde . . . . . Prices
Indus GT . . . . .

TOUCH TABLETS
Koala Touch Tablet-D . . . 69.95
Koala Touch Tablet-Cart 74.95

RS232 Modem Adaptor
(Through serial port) 39.95

MONITORS
USI . . . . . Call
AMDEK . . . . . Call

STIMUTECH SUBLIMINAL SOFTWARE FOR ATARI AND CBM 64
Expando-Vision Interface w/one FREE Cart . . . 99.00
Weight Control
Study Habits
Stress Control
Drinking Control
Smoking Control
Career Success
Sexual Confidence
Addit'l Rom Carts 29.95 ea.

ATARI SOFTWARE

ATARI
Conversational Lang-T . . 44.95
My First Alphabet-D 26.95
Home Filing Mgr.-D 37.95
Star Raiders-Cart 32.95
Assembly Editor-Cart . . . 46.95
Macroassembler-D 67.95
Invitation to Program I-T . 19.95
Basketball-Cart 26.95
Graph-It-T 15.95
Dig Dug-Cart 32.95
Atariwriter-Cart 74.95
Donkey Kong-Cart 37.95
Ms. Pac-Man-Cart 39.95
Eastern Front-Cart 32.95
Donkey Kong Jr.-Cart . . . 39.95
Pengo-Cart 35.95
Logo-Cart 79.95
Robitron-Cart 35.95
Pole Position-Cart 39.95
Microsoft Basic II-Cart . . 67.95
Paint-D 33.95
Caverns of Mars-Cart . . . 32.95
Joust-Cart 39.95
Jungle Hunt-Cart 35.95
Mario Bros.-Cart 35.95
Syncalc-D 74.95
Atari Music-D 29.95

SIERRA ON-LINE
Homeword-D 49.95
Dark Crystal-D 27.95
Frogger-D/T 23.95
Quest For Tires-D 23.95
Ultima I-D 23.95
Ultima II-D 41.95
Wizard/Princess-D 22.95

INFOCOM
Deadline-D 34.95
Witness-D 34.95
Zork I, II, III-D 27.95
Starcross-D 27.95
Planetfall-D 34.95
Suspended-D 34.95
Enchanter-D 34.95
Infidel-D 34.95
Sorcerer-D 34.95

LJK
Letter Perfect-D 74.95
Data Perfect-D 74.95
Spell Perfect-D 64.95

BRODERBUND
Arcade Machine-D 41.95
Bank St. Writer-D 49.95
Lode Runner-D 23.95
Drol-D 23.95
Spare Change-D 23.95
Choplifter-D 23.95
Mask of the Sun-D 27.95
Operation Whirlwind-D . . 27.95

EPYX
Gateway To Apshai-Cart 27.95
Pitstop-Cart 27.95
Fun With Music-Cart 27.95
Jumpman Jr.-Cart 27.95
Dragonriders of Pern-D/T 27.95
Temple of Apshai-D/T . . 27.95
Hellfire Warrior-D/T 20.95
Fun With Art-Cart 27.95
Summer Games-D 27.95
Puzzlemania-D 23.95

SSI
Carrier Force-D 41.95
Combat Leader-D/T 27.95
Cosmic Balance-D 27.95
Cosmic Balance II-D 27.95
Fortress-D 23.95
Rails West-D 27.95
Broadsides-D 27.95
War In Russia-D 54.95
Questron-D 34.95

SPINNAKER
Up For Grabs-Cart 27.95
Cosmic Life-Cart 23.95
Facemaker-Cart 23.95
Alphabet Zoo-Cart 23.95
Fraction Fever-Cart 23.95
Kids on Keys-Cart 23.95
Delta Drawing-Cart 27.95
Story Machine-Cart 27.95
Amazing Things-D 27.95
Hey Diddle Diddle-D 20.95
Snooper Troops 1 or 2-D . 30.95
Trains-D 27.95
Aerobics-D 30.95
Adventure Creature-D . . . 27.95

PARKER BROS.
Astrochase-Cart 34.95
Frogger-Cart 34.95
Q-Bert-Cart 34.95
Popeye-Cart 34.95

MISCELLANEOUS FOR ATARI
Castle-Wolfenstein-D . . . 20.95
Home Accountant-D 52.95
Master Type-D/Cart 27.95
Spelunker-D 27.95
Flight Simulator II-D 37.95
Zombies-D/T 23.95
Mr. Robot-D 23.95
River Raid-Cart 31.95
Diskey-D 34.95
Ultima III-D 41.95
Miner 2049'er-Cart 34.95
Scraper Caper-Cart 34.95
Basic XL-Cart 74.95
Monkey Wrench II-Cart . . 49.95
Omnimon 82.95
Starbowl Football-D/T . . 22.95
Hockey-Cart 27.95
Flip Flop-D/T 20.95
Boulder Dash-D/T 20.95
Encounter-D/T 23.95
Quasimoto-D/T 23.95
N.Y.C.-D/T 23.95
Compute Books . . . . . Call
Pitfall-Cart 31.95
Codewriter-D 69.95
The Heist-D 27.95
Mac 65XL-Cart 74.95
Jupiter Mission-D 34.95
Nibbler-D 20.95
Air Support-D/T 23.95
Pogo Joe-D/T 20.95
Analog Compendium -Book 9.95
Movie Maker-D 41.95
Antica 4-D 34.95

Hundreds of items available for the Atari and CBM 64 . . . please call.

D - Disk T - Cassette
Cart - Cartridge

COMMODORE 64

CBM 64 . . . . . Call
1541 Disk Drive 239.00
1526 80 Column Printer 279.00
1530 Datasette 66.00
1702 Color Monitor 249.00
1650 AD/AA Modem 89.00
RS 232 Interface 44.00

COMMODORE 64 SOFTWARE

Program Ref. Guide 19.95
Assembler-D 17.95
Easy Finance I, II, II, IV-D 19.95
Easy Calc-D 64.95
Easy Mail-D 17.95
Easy Script-D 39.95
Easy Spell-D 19.95
Logo-D 49.95
The Manager-D 37.95
General Ledger-D . . 37.95
Accts. Rec.-D 37.95
Accts. Pay.-D 37.95
Magic Desk-D 52.95
Zork I, II or III-D 29.95
Suspended-D 29.95
Starcross-D 29.95
Deadline-D 29.95

COMMODORE 64 SOFTWARE

ACCESS
Neutral Zone-D/T 23.95
Spritemaster-D/T 27.95
Beachhead-D/T 23.95
Music Composer-D 27.95

HES
Multiplan-D 69.95
64 Forth-Cart 41.95
HES Writer-Cart 32.95
HES Mon-Cart 27.95
Turtle Graphics-Cart . . . 41.95
Mutant Camels 20.95
HES Modem 59.95
Super Zaxxon . . . . . Call

SYNAPSE
Blue Max-D/T 23.95
Zaxxon-D/T 27.95
Morgul-D/T 23.95
Shamus II-D/T 23.95
Sentinal-D/T 23.95
Pharoah's Curse-D/T . . 23.95
Slamball-D/T 23.95
Zepplin-D/T 23.95

CARDCO
Cardprint/A 59.95
Cardprint/B 37.95
Cardco + G 64.95
Cardboard/5 . . . . . 54.95
Cardkey 29.95
Cardette/I 29.95
Cardwriter/I 29.95
Cassette Recorder 47.95
Printer Utility-D/T 19.95
Write Now-Cart 34.95
Mail Now-D 27.95

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
The Diary-Cart 23.95
The Tool-Cart 29.95

CBS SOFTWARE
Call for items and prices.

SPINNAKER
Trains-D 27.95
Kidwriter-D 23.95
Alf In The Caves-Cart . . 27.95
Jukebox-Cart 23.95

CYMBAL
Accounts Payable-D . . . 52.95
Accounts Receivable-D 52.95
Inventory Control-D 52.95
Invoice Writer-D 52.95

MISCELLANEOUS FOR CBM 64
Flight Simulator II-D . . . 37.95
Night Mission/ Pinball-D 20.95
Praticalc 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 59.95
Money Manager-D/T . . . 19.95
Bristles-D/T 20.95
Telestar 64-Cart 37.95
Star League Baseball-D/T 20.95
Castle Wolfenstein-D . . 20.95
Sorcerer-D 34.95
Computer Baseball-D . . 27.95
Ringside Seat-D 27.95
Professional Golf-D 27.95
Mastertype-Cart 27.95
Solo Flight-D 23.95
Vic Switch 124.95
Paper Clip-D 64.95
Delphi's Oracle-D 74.95
Super Buscard II . . . . . Call
Instaspeed-D/T 99.95
Codewriter-D 69.95
Cyberia Farm Mgr.-D . . . 37.95
First Class Mail-D 34.95
Aztec-D 27.95
Popeye-Cart 34.95
Ultima III-D 41.95
Ultima II-D 41.95
Miner 2049er-Cart 27.95

ACCESSORIES
WICO Joystick . . . . . Call
Flip 'n' File Trunks 20.95
Flip 'n' File Cart 20.95
Joysensor 24.95
Elephant Disks (Box of 10) 20.00
WICO Trakball 37.95
KRAFT Joystick 15.95

To Order Call Toll Free
800-558-0003

For Technical Info, Order Inquiries, or for Wisc. Orders -
414-351-2007

New Lower Prices

NO SURCHARGE FOR MASTERCARD OR VISA

ORDERING INFORMATION. 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

```
0660 A200       1201     LDX #0*16   ; channel 0 is screen
0662 A906       1202     LDA #MESSAGE/256
0664 A003       1203     LDY #MESSAGE&255 ; address of message to A,Y
0666 202406     1204     JSR PRINT   ; do the actual work
                1300 ; BASIC: INPUT NAME$
0669 A200       1301     LDX #0*16   ; channel 0 is screen
066B A905       1302     LDA #NAME/256
066D A080       1303     LDY #NAME&255 ; address of name to A,Y
066F 202906     1304     JSR INPUT   ; do the actual work
                1400 ; BASIC: OPEN #1,4,0,NAME$
0672 A210       1401     LDX #1*16   ; channel 1
0674 A904       1402     LDA #4
0676 9D4A03     1403     STA ICAUX1,X ; mode for open
0679 A905       1404     LDA #NAME/256
067B A080       1405     LDY #NAME&255 ; address of name
067D 201A06     1406     JSR OPEN    ; again, do the actual work
0680 30DE       1407     BMI BEGINWORK ; if bad file or name, ask again
                2000 ; BASIC: REM get and check header
                2001 ;  (just a comment)
                2002 ;
0682            2003 LINE200
                2004 ;
                2100 ; BASIC: TRAP 400
                2101 ;  implemented in code below
                2200 ; BASIC: GET #1,LOW : GET #1,HIGH
0682 A210       2201     LDX #1*16   ; channel 1
0684 204606     2202     JSR GET     ; get a byte to A reg
0687 8D1806     2203     STA LOW
068A A210       2204     LDX #1*16   ; channel 1
068C 204606     2205     JSR GET     ; get a byte to A
068F 8D1906     2206     STA HIGH
                2300 ; BASIC: TRAP 40000
                2301 ;  really, just turns trap off
                2302 ;  again, implemented in code below
                2400 ; BASIC: IF LOW=255 AND HIGH=255
                2401 ;           GET #1,LOW : GET #1,HIGH
0692 C9FF       2402     CMP #255
0694 D015       2403     BNE EOL240  ; HIGH is not 255
0696 CD1806     2404     CMP LOW
0699 D010       2405     BNE EOL240  ; LOW is not 255
069B A210       2406     LDX #1*16   ; LOW=255 and HIGH=255, so...
069D 204606     2407     JSR GET     ; get a byte to A reg
06A0 8D1806     2408     STA LOW
06A3 A210       2409     LDX #1*16   ; channel 1
06A5 204606     2411     JSR GET     ; get a byte to A
06A8 8D1906     2412     STA HIGH
06AB            2413 EOL240
                2414 ; (the BNE's get here to skip the THEN statements)
                2499     .PAGE ".        BASIC lines 250-410"
                2500 ; BASIC: START=LOW+256*HIGH
06AB AD1806     2501     LDA LOW
06AE 85CE       2502     STA START   ; move LOW byte
06B0 AD1906     2503     LDA HIGH
06B3 85CF       2504     STA START+1 ; same as 256*HIGH
                2600 ; BASIC: GET #1,LOW : GET #1,HIGH
06B5 A210       2601     LDX #1*16   ; channel 1
06B7 204606     2602     JSR GET     ; get a byte to A reg
06BA 8D1806     2603     STA LOW     ; LOW byte of QUIT
06BD A210       2604     LDX #1*16   ; channel 1
06BF 204606     2605     JSR GET     ; get a byte to A
06C2 8D1906     2606     STA HIGH    ; HIGH byte of QUIT
                2700 ; BASIC: QUIT = LOW+256*HIGH
                2701 ;  already done by 2601 thru 2606, see text
                3000 ; BASIC: REM read in a segment
                3001 ;  (just a comment)
```

```
                3100 ; BASIC: FOR ADDR=START TO QUIT
                3101 ;  a small sneaky here:
                3102 ;      START and ADDR are same
                3103 ;      location in ass'y language version
                3104 ;
06C5            3105 FORADDRLOOP
                3106 ;
                3200 ; BASIC: GET #1,BYTE
06C5 A210       3201     LDX #1*16    ; channel 1
06C7 204606     3202     JSR GET      ; get a byte to A-reg
                3203     ;            notice that we do not actually store it
                                      yet
                3300 ; BASIC: POKE ADDR,BYTE
06CA A000       3301     LDY #0       ; needed for indirect addressing
06CC 91CE       3302     STA (ADDR),Y ; an effective poke
                3400 ; BASIC: NEXT ADDR
                3401 ;   strangely, this simple BASIC statement
                3402 ;   causes a lot of work in ass'y language
06CE A5CE       3403     LDA ADDR
06D0 CD1806     3404     CMP QUIT     ; at end of loop yet?
06D3 D007       3405     BNE DONEXT   ; no
06D5 A5CF       3406     LDA ADDR+1
06D7 CD1906     3407     CMP QUIT+1   ; try high bytes also
06DA F008       3408     BEQ ENDOFFOR ; aha! not yet
06DC            3411 DONEXT
06DC E6CE       3412     INC ADDR     ; change low byte of addr
06DE D0E5       3413     BNE FORADDRLOOP
06E0 E6CF       3414     INC ADDR+1   ; and high byte if needed
06E2 D0E1       3415     BNE FORADDRLOOP
                3416 ;
06E4            3417 ENDOFFOR
                3500 ; BASIC: GOTO 200 : REM try another segment
06E4 4C8206     3501     JMP LINE200
                4000 ; BASIC: REM trapped to here, assume end of file
                4001 ;  (just a comment)
06E7            4002 LINE400
                4100 ; BASIC: CLOSE #1
06E7 A210       4101     LDX #1*16    ; channel 1
06E9 201F06     4102     JSR CLOSE    ; do the work
06EC 60         4103     RTS          ; no parallel in BASIC,
                4104 ; we have to RTS to operating system
```

64 EXPLORER

Larry Isaacs

A complete drawing package should allow the user to print characters on the bitmapped display. This month and next I will discuss this topic, and give more examples on the use of the drawing routines presented last month.

There are two methods for printing characters on a bitmapped display. We can POKE the dot patterns of the characters to the bitmapped RAM, or we can draw the characters onto the display.

Let's take a look at the first method, which is faster because no line-drawing routines are required.

POKEing To The Bitmap

The first step in POKEing characters to a bitmapped display is to choose the *cell size*, or dimensions, of our character set. The choice of the cell size can greatly affect the complexity of the routines which print the characters. If a convenient size is chosen, the routines will be simplified; if you are up for a challenge, you can write the routines to accept a variable cell size.

We will use a cell size of 8 dots high by 8 dots wide, for two reasons. First, a width of 8 dots is the number of dots which can be held in a byte. Second, there already exists a set of 8 x 8 characters in the 64's character ROM.

Actually, the 64's normal character display mode is very similar to what we want to accomplish in a bitmapped mode. The process involves character cells and some method of transferring the character dot patterns to the display. However, in the normal character display mode, the format is determined by the character codes found in a character array called screen memory. In screen memory, you can change only whole 8 x 8 characters, so your effective resolution winds up being 40 columns by 25 lines.

Blending Characters And The Bitmap

When using a bitmap display, you can control each dot. This implies that you can place a character at any X,Y position on the screen. This can certainly be done, though it is more difficult than placing a character code in screen memory. What complicates the task somewhat is that the 64's VIC-II chip organizes a bitmap display as groups of 8 x 8 dot cells.

It's possible for the 8 x 8 dot character pattern to span as many as four of the bitmap cells, two horizontally and two vertically. This doesn't create much of a problem vertically, but horizontally the bytes in the character dot pattern may have to be moved or shifted to span two bytes. In addition, when the bytes are added to the bitmap, the routine must not disturb the dots outside the shifted 8 dots of the character pattern.

Next, we must decide how to transfer the dot patterns so they will be visible against the bitmapped background.

Using Conditional Logic

One way of transferring the dot pattern is to add (logical OR) the dots in the pattern to the dots already in the display. Dots which are on in the character dot pattern are also turned on in the display. Dots which are on in the display remain on. This avoids erasing the background as a character is printed to the bitmapped display, but can result in illegible characters if there are too many dots already turned on in the background.

Another way to transfer the dot pattern is to flip (Exclusive-OR) the dots in the pattern into the bitmapped RAM. Dots in the bitmapped RAM which correspond to on dots in the dot pattern are flipped to the opposite state. The advantage of this technique is that it will make characters visible regardless of whether the background is on or off. However, characters can still be illegible if the background is not predominantly either on or off.

Or the transfer could be accomplished by writing the pattern directly into the bitmapped RAM. This type of transfer replaces the background with the character cell. We will use this technique.

A BASIC Example

Let's first demonstrate how the required routines might be implemented in BASIC. Unfortunately,

MICRO-SYS DISTRIBUTORS. THE COMMODORE CONNECTION.

commodore

SOFTWARE FOR C-64

Business

Multiplan	$ 89.00
WordPro 3 + /64 w/SpellRight Plus	$ 79.00
SpellRight Plus	$ 55.00
Calc Result (Advanced)	$ 95.00
Calc Result (Easy)	$ 45.00
Mirage Concepts (data base)	$ 95.00
Mirage Concepts (40 & 80 clm W/P with Dictionary)	$ 95.00
Home Accountant (Continental)	$ 69.00
Tax Advantage (Continental)	$ 49.00
Southern Solutions Accounting G/L, A/R, A/P, P/R, I/M ..... each	$ 69.95

Utilities

Super Basic 64	$ 35.00
Super Copy 64	$ 35.00
Sketch Pad 64	$ 75.00
64 Forth	$ 45.00
MTS Terminal Package (up and down load)	$ 35.00
Simons Basic	$ 19.95
80 Column Expander	$ 55.00

ACCESSORIES

Solo Flight (Simulator)	$ 34.95
Oscar by Databar (Bar Code Reader)	$ 79.95
CBM 1541 Disk Drive	$ 249.00
Concord Parallel Disk Drive	$ 345.00
Concord Slave Drive	Call
MSD Super Disk (Single)	$ 395.00
MSD Super Disk (Dual)	$ 695.00
Vic 1650 Automatic Modem	$ 109.95
Hayes Smart 300 Modem	$ 249.00
Hayes Smart 1200 Modem	$ 629.00
Vic 1530 Datasette	$ 65.00
CBM 1520 Printer Plotter	$ 179.95
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 (Disk)	$ 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	$ 12.00
CBM 8023 Ribbons	$ 14.95
Flip N' File 10, 15, 25, 50	Call
Power Strips w/surge stopper	Call
Computer Care Kit	$ 19.95

INTERFACES

Interpod (full compatibility!!) (Intelligent IEEE & RS232)	Call
The Connection (By Tymac) (Commodore Graphics + 2K Buffer)	$ 95.00
Cardco + G Parallel Interface	$ 79.00
Vic Switch	$ 149.95
ADA 1800 (Parallel-8032 only)	$ 129.00
ADA 1450 (Serial-8032 only)	$ 149.00
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
Custom Computer Cables (we make to your specifications)	Call

MONITORS

CBM 1702 Color Monitor	$ 269.00
Panasonic CT-160 Color	Call
Panasonic TR-120 (Green)	Call
Panasonic TR-120 (Amber)	Call
Panasonic DT-1300 (RGB)	$ 395.00
Monitor Stand (Tilt & Swivel)	$ 29.95
RGB Monitor Cable:	
ET-100C (Apple)	$ 33.80
ET-101C (IBM)	$ 33.80

LETTER QUALITY PRINTERS

Transtar 120 (80 column)	$ 535.00
Transtar 130 (132 column)	$ 725.00
CBM 6400 Printer	$1425.00
NEC Spinwriter	Call
Cardco LQ/1 Printer	$ 565.00

DOT MATRIX PRINTERS

CBM MPS-801 Printer (50cps)	$ 245.00
Okidata 82A	$ 412.50
Okidata 83	$ 635.00
Okidata 84P	$1165.00
Okidata 92P	$ 519.00
Okidata 93P	$ 810.00
Panasonic KX-P1090 Printer	Call
Panasonic KX-P1091 Printer	Call
Panasonic KX-P1092 Printer	Call
Panasonic KX-P1093 Printer	Call
Star Gemini 10X	$ 295.00
Star Gemini 15	$ 499.00
Star Gemini Delta 10	Call

DEALERS INQUIRIES WELCOME

Call to Order

1-800-527-1738

All Others Call

1-214-231-2645

COMMODORE BUSINESS MACHINES

Executive 64 portable (new)	Call
B128-80 128k Bus. Machine (new)	Call
SuperPet (5 languages)	$1059.00
CBM 8032	$ 625.00
CBM 2031 single disk	$ 295.00
CBM 8050 Dual Disk 1 meg.	$ 995.00
CBM 8250 Dual Disk 2 meg.	$1295.00
64K Expansion Board	$ 275.00
SuperPet Upgrade Kit	$ 695.00

BUSINESS SOFTWARE — 8032

WordPro 4 + or 5 +	$ 305.00
Calc Result	$ 199.95
The Manager	$ 199.00
BPI Accounting System (5 separate modules)	$ 325.00
Southern Solutions Accounting System III (Per/Module)	$ 285.00
McTerm Communications Package	$ 175.00

BUSINESS SOFTWARE — B128-80

Superscript II	Call
Superbase	Call
Superoffice	Call
Complete Accounting Systems	Call
Mailing List Manager	Call
Order Entry/Point of Sale	Call
Calc Result	Call

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.

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.)

like the drawing routines presented in earlier columns, the character routines are too slow to be really useful. To enhance their value as an example, we'll try to illustrate modular programming style as well.

One of the main aspects of modular programming is breaking main or primary tasks into smaller, more manageable tasks. Once the tasks have been broken down sufficiently, each may be implemented in a single routine. The more independent each of these separate routines is, the better. This allows you to concentrate on the details involved with the routine as it is written, without being distracted by the details involved with other routines. To show how printing to the bitmapped display might be broken into modules, let's take a look at the logical subdivisions of this task.

Although this program isn't really complex enough to justify a modular approach, I prefer to keep the functions or tasks in separate routines, so long as the routines don't become embarrassingly simple. This helps while debugging, since the symptoms of the bug often eliminate a majority of the routines from consideration. It also helps keep you from accidentally tangling functions together.

When functions get tangled or intertwined, making one change may require making other changes, leading to a snowball effect. And finally, it is good practice to keep functions divided into separate routines when you write a complex program. How well the tasks are divided up can greatly affect how much effort it takes to write and debug the program.

Breaking Down The Task

Putting a character in a bitmapped display will involve transferring bytes into bitmapped RAM, so we will need a routine which does the transferring. We need another routine to calculate the character's position. We also need to know what to write; this will require two routines. We need a routine which will find and read the appropriate bytes in the character ROM. However, the dot patterns are organized based on screen codes, which are different from the Commodore 64 ASCII codes you normally print. This means we need a routine to convert the ASCII code to the corresponding screen code.

Finally, we need a routine to do the horizontal shifting necessary when the character byte needs to span two bytes in the bitmapped RAM. This gives us five routines to be implemented:

1. Convert the character to screen code
2. Read the character's dot pattern
3. Calculate its position in the bitmap and amount of shift
4. Shift a dot-pattern byte
5. Put the dot-pattern byte in the bitmap

By dividing the tasks into well-defined and independent sections, it will be a little easier to implement them than if you tried to throw it all together in one routine. For example, converting the ASCII character code to a screen code can be done without concerning ourselves with where the ASCII code came from, or for what the screen code will be used. The shift section does not need to account for where the shift amount came from or what will be done with the shifted bytes.

Combining The Modules

Once we build the character print routine from these five sections, it is simple to build a string print routine using the character print routine. The result might be a BASIC program like the one that accompanies this article. The program uses the machine language routines discussed in this column in previous issues. Before running this program, you must run the BASIC loader presented in the May 1984 issue. The subroutine at line 100 converts ASCII code CH to the equivalent screen code SC. The subroutine at line 200 uses screen code CH to read the associated dot pattern into the array DP(). This subroutine also uses CP which points to the base of the character dot patterns in ROM.

The subroutine at line 300 uses the coordinates in X and Y to calculate an offset OF into the bitmap, and the shift amount SH. The subroutine at line 400 uses the shift amount SH to right-shift the byte in BY partially into B2. This means shifting dots out of the right end of the byte and into the left end of the other byte. This shift routine also makes the mask bytes, M1 and M2.

Finally, the subroutine at line 500 writes the bytes into the bitmap base of the offset OF, calculated earlier. This routine also uses the mask bytes to keep the necessary old bits from the bitmap bytes, before adding the new dot pattern bits. The subroutine at line 600 prints the character at the current coordinates specified by X and Y, and the subroutine at 700 prints a string at X,Y.

Logical Math

I have used logical operators (OR and AND) rather than division and the INT functions. For example, in line 320, the term (X AND −8) gives the same result as INT(X/8)*8. In the subroutine at line 200, the POKEs are required to turn off interrupts and make the character ROM accessible to the BASIC program.

The main routine uses the string-printing routine at line 700 to label the vertical axis for the plot of a sine wave. As you will see, the

character printing is pretty slow. This part of the program would be much more useful written in machine language. Next month I will discuss the drawing method of putting characters in the bitmapped display, and present machine language routines for both.

Characters On A Bitmapped Display

```
10 REM PRINT CHARACTERS TO BIT-MAP:rem 63
20 JV=49152:REM JUMP TABLE          :rem 6
30 CP=53248:REM LOC. OF CHAR. PATTERNS
                                    :rem 181
40 POKE 785,PEEK(JV+28):REM SETUP USR()
                                    :rem 8
50 POKE 786,PEEK(JV+29)             :rem 17
60 GOTO 1000                        :rem 96
100 REM CONVERT CHAR. TO SCREEN CODE
                                    :rem 96
110 IF CH>31 AND CH<64 THEN SC=CH:RETURN
                                    :rem 249
120 IF CH>63 AND CH<96 THEN SC=CH-64:RETU
    RN                              :rem 155
130 IF CH>95 AND CH<128 THEN SC=CH-32:RET
    URN                             :rem 200
140 IF CH>127 AND CH<192 THEN SC=CH-64:RE
    TURN                            :rem 251
150 SC=CH-128:RETURN                :rem 214
200 REM GET CHARACTER DOT PATTERN:rem 232
210 POKE 56334,PEEK(56344) AND 254
                                    :rem 221
220 POKE 1,PEEK(1) AND 251           :rem 50
230 FOR IX=0 TO 7                   :rem 100
240 DP(IX)=PEEK(CP+SC*8+IX):NEXT    :rem 202
250 POKE 1,PEEK(1) OR 4             :rem 159
260 POKE 56334,PEEK(56334) OR 1      :rem 69
270 RETURN                          :rem 121
300 REM CALC OFFSET AND SHIFT COUNT
                                    :rem 43
310 TY=199-Y:SH=X AND 7              :rem 27
320 OF=(TYAND-8)*40+(XAND-8)+(TYAND7)
                                    :rem 106
330 RETURN                          :rem 118
400 REM SHIFT BYTE TO CORRECT POSITION
                                    :rem 84
410 B2=0:M1=0:M2=255:IF SH=0 THEN RETURN
                                    :rem 13
420 FOR K=1 TO SH:B2=B2/2           :rem 52
430 IF BY AND 1 THEN B2=B2 OR 128   :rem 85
440 BY=BY/2:M1=(M1/2)OR128:M2=M2/2:NEXT
                                    :rem 28
450 RETURN                          :rem 121
500 REM PUT BYTE AT X,Y              :rem 24
510 GOSUB 300:REM CALCULATE OF & SH
                                    :rem 171
520 GOSUB 400:REM SHIFT OVER        :rem 131
530 AD=57344+OF:REM GET ADDRESS FOR BY
                                    :rem 167
540 POKE AD,USR(OF) AND M1 OR BY    :rem 230
550 IF SH=0 THEN RETURN              :rem 64
560 POKE AD+8,USR(OF+8) AND M2 OR B2
                                    :rem 136
570 RETURN                          :rem 124
600 REM PUT CHARACTER AT X,Y        :rem 114
610 GOSUB 100:REM CONVERT CH        :rem 114
620 GOSUB 200:REM READ DOT PATTERN
                                    :rem 233
630 Y=Y+8:REM PUT CHAR. FROM TOP DOWN
                                    :rem 173
640 FOR IX=0 TO 7:Y=Y-1:BY=DP(IX):rem 136
650 GOSUB 500:REM PUT BYTE          :rem 251
660 NEXT:RETURN                     :rem 245
700 REM PUT STRING S$ AT X,Y         :rem 52
710 FOR SP=1 TO LEN(S$)             :rem 218
720 CH=ASC(MID$(S$,SP,1))           :rem 123
730 GOSUB 600:REM PUT THE CHARACTER
                                    :rem 53
740 X=X+8:NEXT                      :rem 100
750 RETURN                          :rem 124
1000 REM MAIN ROUTINE               :rem 240
1010 SYS JV:SYS JV+6,0:SYS JV+9,0,1
                                    :rem 237
1020 FOR I=0 TO 10                  :rem 100
1030 LB=-1+I*.2:S$=STR$(LB)         :rem 213
1050 X=5:Y=46+10*I:GOSUB 700:NEXT:rem 147
1060 SYS JV+12,32,50:SYS JV+18,32,150
                                    :rem 214
1070 SYS JV+12,32,100:SYS JV+18,319,100
                                    :rem 54
1080 FOR I=0 TO 10                  :rem 106
1090 X=30:Y=50+10*I                 :rem 246
1100 SYS JV+12,X,Y:SYS JV+18,X+4,Y
                                    :rem 205
1110 NEXT                            :rem 2
1120 SYS JV+12,32,100:PI=3.1416     :rem 124
1130 SX=256/(2*PI):SY=50             :rem 71
1140 FOR I=0 TO 2*PI STEP 2*PI/100
                                    :rem 236
1150 SYS JV+18,32+I*SX,100+SIN(I)*SY
                                    :rem 22
1160 NEXT                            :rem 7
9000 GET Z$:IF Z$="" THEN 9000      :rem 231
9010 SYS JV+3                       :rem 199
```

MACHINE LANGUAGE

Jim Butterfield, Associate Editor

Decimal Mode
Part 1

The 6502 has an option which affects only the add (ADC) and subtract (SBC) instructions: decimal mode.

Decimal mode is invoked with the Set Decimal (SED) command, and canceled with Clear Decimal (CLD). It may be affected by stack activities that pull the status register—PLP for Pull Processor status, and RTI—but this is unusual. In most computer environments you can assume that decimal mode is not in force when your program is invoked; but if you're not sure, it won't hurt to give a CLD.

Decimal mode is intended to help with certain types of numbers: Binary Coded Decimal (BCD) numbers. You might want to use this type of number system when the values are used mostly for input and output with little calculation involved.

Binary numbers—the computer's usual numeric values—are good for advanced calculations. Multiplication and division are easy to do in binary, and more advanced calculations can readily be developed. The only problem with binary numbers is this: They must be converted to decimal at the time of input or output.

Decimal numbers, or more accurately BCD numbers, are easy to input and output since they are held in the same decimal notation as was entered or will be seen by the user. With decimal mode, we may add or subtract these numbers without converting them to binary. But if we want to do more advanced mathematics, we'll certainly go to binary.

Accounting programs often use decimal mode. Similarly, many games keep scores in decimal format, since the only activities are adding points as they are scored and displaying the results.

What Is BCD?

The easiest way to describe a number held in Binary Coded Decimal is this: When you display it in hexadecimal format, you see the correct decimal value. Let's explain this with a few examples.

A value of 9 is held within a byte as binary 00001001. This is true whether you are using binary or BCD numbering. If we print the contents of this byte in hexadecimal, it is displayed as 09. Now, this not only represents the value nine, it looks like nine.

If we are in binary mode and add one to the above value, we'll get 00001010. The value is ten but the number displays in hex as 0A. This doesn't look like ten to those of us who are not trained to read hex. Worse: If we add six, we'll get a value of 16, which prints as hex value 10. This doesn't look like 16—if we didn't know it was a hexadecimal number, we might think it was ten.

Let's go back to our original value of nine, but switch to decimal mode. If we add one, using the ADC instruction, we'll end up with binary 00010000. We know that the value must represent ten, and when we print the hexadecimal it shows up as 10—which looks like ten. We must ignore the usual binary rules, which would tell us that binary 00010000 is equivalent to decimal 16. In BCD, this binary number has a value of 10. If we add a six in decimal mode, we'll get 00010110 which has a value of 16 and prints out as hexadecimal 16.

We've decided to use the bits in a different way. The four high bits—the high nybble, as it's sometimes called—represent a tens digit; the four low bits, or low nybble, represent units. Each nybble may have a value from 0 to 9, but the six

Why Pay Retail?

Business

Commodore

Easy Script 64/D	$ 39
Easy Spell 64/D	$ 17
Easy Finance I, II ea./D	$ 17
Accts. Receivable/D	$ 39
General Ledger/D	$ 39
Payroll/Checkwriting/D	$ 39
Assembler 64/D	$ 15
Logo/D	$ 49
Simon's Basic/CRT	$ 17
Bank Street Writer/D	$ 49
Home Accountant/D	$ 48
Tax Advantage	$ 39
FCM	$ 39
Paper Clip Word Processor	$ 65
Delphi's Oracle	$ 89
Practicalc 64	$ 40
Calc Result/Easy-CRT	$ 59
Calc Result/Advanced-D	$109
Multiplan/D	$ 65
Data Manager/D	$ 17
Electronic Checkbook	$ 17
Swif tax/D	$ 39
Quick Brown Fox/D	$ 56

Atari

Atariwriter	$ 79
Visicalc	$179
Financial Wizard	$ 45
Letter Perfect	$ 79
Data Perfect	$ 79
Spell Perfect	$ 59
Spell Wizard	$ 45
Text/Spell Wizard Combo	$ 65
Bank Street Writer	$ 49
Bank Street Speller	$ 49
Homeword	$ 49
Syn File +	$ 65
Syn Text	$ 23
Syn Mail	$ 34
Home Accountant	$ 48
Tax Advantage	$ 39
Complete Personal Accountant	$ 65
Atspeller	$ 33
Diskette Mailing List	$ 21
Miles Payroll System	$119
Atari Accounting	$169
Family Finance	$ 36
File Manager 800	$ 65

Education

Atari

Harcourt Brace SAT/D	$59
Chaterbee/D	$29
Match Wits/D	$22
Early Games/D	$22
Early Games Music/D	$22
Spellicopter	$29
Creature Creator	$29
Math Maze	$29
Alien Addition/D	$25
Meteor Multiplication	$25
Compu-Read/D	$22
Compu-Math/Fractions	$29
Compu-Math/Decimals	$29
Spelling Bee Games	$29
Speed Read +/D	$45
Spider Eater/Koala	$22
Bumble Bee/Learning Co.	$29
Snooper Troops 1, 2 ea.	$32
Face Maker/CRT	$25
Kids on Keys/CRT	$25
Alphabet Zoo/CRT	$25
Kid Writer/CRT	$25
Delta Drawing/CRT	$29

Commodore

CBS Addition/Subt.	$17
CBS Multiplication/ Division	$17
Speed Reader II/D	$49
Word Attack/D	$39
Mathblaster/D	$39
Spellakazan/D	$29
Crypto Cube/D	$29
Master Type/D-CRT	$29
Songwriter/D	$29
Alphabet Zoo/CRT	$25
Fraction Fever/CRT	$25
Delta Drawing/CRT	$29
Facemaker/CRT	$25
Trains/D	$29
Kidwriter/CRT	$25
Dungeons of Algebra Dragons	$17
Juggles Rainbow	$22
Bumble Bee	$29
Early Games Piece of Cake	$22
Early Games Match Maker	$22

Specials of the Month

Atari 600XL	$169
Atari 800XL	$299
Rana 1000 Disk Drive	$379
1027 Letter Quality Printer	$299
Commodore SX64 Portable Computer	$899
Commodore Graphics Printer	$269
Koala Touch Tablet	$ 79.00
Flip & File 50	$ 22.50
Elephant Disks S/S	$19.50

Monitors

Leading Edge Green Hi-Res 12"	$ 89
USI Amber	$169
NEC Green Screen	$169
Leading Edge Amber Hi-Res 12"	$ 89
Amdek Color I+	$315

Hardware

Printers

Leading Edge Gorilla	$189
NEC 8023A	$459
Okidata ML 82A	Call
Okidata ML 83A	Call
Okidata ML 84P	Call
Okidata ML 92P	Call
C. ITOH 8510 Prowriter	$399
C. ITOH 8510 SP	$665
Mannesmann Tally 160L	$629
Gemini 10X	$315
Mannesmann Tally Spirit	$345
Alphacom 81	$189

Modems

Atari 1030 Modem	$115
Commodore Automodem	$ 95
Hayes 1200	$529
Hayes 300	$229

Recreation

Crisis Mountain	$ 25
Ken Uston's Professional Blackjack	$ 49
Temple of Apshai	$ 29
Lode Runner	$ 25
Aztec	$ 29
Gumball	$ 25
Mask of the Sun	$ 29
Witness	$ 36
Planetfall	$ 36
Deadline	$ 36
Zork I, II, III ea.	$ 29
Enchanter	$ 36
Infidel	$ 36
Blue Max	$ 25
Joust	$ 43
Robotron 2084	$ 36
Pole Position	$ 36
Ms. Pac Man	$ 36
B-1 Nuclear Bomber (C)	$ 12
Midway Campaign (C)	$ 12
Nukewar (C)	$ 12
Telengard (C)	$ 17
Jumpman Jr./CRT	$ 29
Pitstop/CRT	$ 29
Flight Simulator II/D	$ 39
Night Mission Pinball	$ 20
Q Bert/CRT	$ 39
Frogger/CRT	$ 39
Popeye/CRT	$ 39

Computer Outlet

1095 East Twain, Las Vegas, NV 89109 Mon.–Fri. 8 a.m. to 6 p.m., Sat. 9 a.m. to 5 p.m.

1-800-634-6766

Information and Inquiries: (702) 796-0296 Order Status: (702) 369-5523.

IF YOU DON'T SEE IT ... CALL!

ORDERING INFORMATION AND TERMS: For fast delivery send cashier checks, money orders or direct bank wire transfers. Personal and company checks allow 3 weeks to clear. Charges for C.O.D. orders are $3.00 minimum or 1% for orders over $300. School purchase orders welcomed. Prices reflect a cash discount only and are subject to change without notice. Please enclose your phone number with any orders. SHIPPING: — Software: $3.00 minimum. SHIPPING — Hardware: (Please call) SHIPPING — Foreign Orders: APO & FPO orders: $10 minimum and 15% of all orders over $100. Nevada residents add 5¼% sales tax. All goods are new and include factory warranty. Due to our low prices, all sales are final. All returns must be accompanied by a return authorization number. Call 702-369-5523 to obtain one before returning goods for replacement. Catalogs: .50¢ U.S., $1.00 foreign.

VISA

highest combinations corresponding to hex A, B, C, D, E, and F will never be used.

This makes BCD less efficient than binary for storing numbers. The highest BCD number that we can store within a single byte is 99, as compared to 255 for binary. We can use several bytes together to hold larger numbers, but BCD always holds less: A two-byte BCD number can go from 0000 to 9999, compared to a two-byte unsigned binary number which can range from 0 to 65535.

But it's convenient. When we wish to output such a number, we extract each digit, convert it to ASCII with an ORA #$30, and print it. (We get the left digit by using four LSR instructions, and the right digit with AND #$0F.) An equivalent binary number would need a divide-by-ten routine before it could be output.

Similarly, input is a snap. As each ASCII digit arrives, it has its high bits stripped (with AND #$0F) and gets packed together with another digit to generate the two-to-a-byte BCD value.

An Example

Here's a sample program to show the power of BCD numbers and ease of programming with them. We'll have the computer (PET, VIC, or 64) output a table of multiples of the number 142857. This is a favorite peculiar number of mine; you'll see why when we print the table.

```
                        ; set value to zero
033C  A2  00            LDX  #$00
033E  8E  90  03              STX  LOW
0341  8E  91  03              STX  MED
0344  8E  92  03              STX  HIGH
                        ; do the addition
0347  18                LOOP CLC
0348  78                     SEI
0349  F8                     SED
034A  AD  90  03              LDA  LOW
034D  69  57                  ADC  #$57
034F  8D  90  03              STA  LOW
0352  AD  91  03              LDA  MED
0355  69  28                  ADC  #$28
0357  8D  91  03              STA  MED
035A  AD  92  03              LDA  HIGH
035D  69  14                  ADC  #$14
035F  8D  92  03              STA  HIGH
0362  D8                     CLD
0363  58                     CLI
                        ; print the number
0364  A0  02                  LDY  #$02
0366  B9  90  03        LP    LDA  LOW,Y
0369  4A                     LSR  A
036A  4A                     LSR  A
036B  4A                     LSR  A
036C  4A                     LSR  A
036D  09  30                  ORA  #$30
036F  20  D2  FF              JSR  $FFD2
0372  B9  90  03              LDA  LOW,Y
0375  29  0F                  AND  #$0F
0377  09  30                  ORA  #$30
0379  20  D2  FF              JSR  $FFD2
037C  88                     DEY
037D  10  E7                  BPL  LP
                        ; print RETURN and loop
037F  A9  0D                  LDA  #$0D
0381  20  D2  FF              JSR  $FFD2
0384  E8                     INX
0385  E0  07                 CPX  #$07
0387  D0  BE                 BNE  LOOP
0389  60                     RTS
```

Note that we hold the value we are calculating in three bytes; called LOW, MED, and HIGH; we add starting at the low byte and working up. The Carry flag works the same way as is usual for addition. While we're in decimal mode, we lock out the interrupt so that the interrupt routines won't do their arithmetic in the wrong mode. The addition sequences could have been written as a loop; for the sake of clarity, it was done using "straight line" coding.

For printing, we start from the high byte, of course. The output routine for BCD is simple compared to what we would need to do with binary values.

If you'd rather enter the program from BASIC, here's the same program in DATA statements. It will work on all Commodore machines.

```
100 DATA 162,0,142,144,3,142,145,3
110 DATA 142,146,3,24,120,248,173,144,3
120 DATA 105,87,141,144,3,173,145,3
130 DATA 105,40,141,145,3,173,146,3
140 DATA 105,20,141,146,3,216,88,160,2
150 DATA 185,144,3,74,74,74,74,9,48
160 DATA 32,210,255,185,144,3,41,15,9,48
170 DATA 32,210,255,136,16,231,169,13
180 DATA 32,210,255,232,224,7,208,190,96
200 FORJ=828 TO 905
210 READX:T=T+X
220 POKEJ,X
230 NEXT J
240 IF T<>8325 THEN STOP
250 SYS 828
```

You might like to examine the output of the program to see what's so special about the first seven multiples of the number 142857.

Next month, we'll discuss special features and wrinkles of decimal mode. ©

Atari Artist

Andrew S. Katz

With this program, you can place shapes of any size, orientation, or color anywhere on the screen. Use the joystick to create the shape, and change its color with the press of a single key.

In spite of its simplicity, "Atari Artist" can be used to draw complex designs as well as realistic scenes. Draw a circle inside a triangle inside a circle, and so on. Piece together a house in the midst of a forest. Then store your art on disk or tape.

Atari Artist comes in two versions. Version 1, a four-color version, has a blue status window and runs in 16K. Version 2, a 16-color GTIA version, has a gray status window and needs 24K to run.

To use this program, you'll need a joystick plugged into port 1. Be sure to have a cassette recorder or disk drive attached if you wish to LOAD and SAVE copies of your designs. When you first RUN the program, the title screen will appear. It will give you information about the keys' uses. At this point, you may wish to select a version. Version 1 is set up by default. If you wish to use Version 2, press and release the joystick button. The number 2 should replace the 1 after the word version. Press the button again to return to Version 1.

Once you've selected the version, move the joystick. If you have selected Version 2 and the message ERROR 147 ON LINE 1000 is on the screen, your Atari does not have enough memory for Version 2. Type RUN again, and this time use Version 1.

Marking The Shape

After several seconds the play screen will appear. All three markers are on top of one another at the top of the screen. Notice the two-line status window at the bottom of the screen.

To move a marker, push the joystick in the direction you want the marker to go. It should respond instantly. The marker you are moving is called the current marker and is indicated by a pinkish tint. The other two markers are white. The markers may move anywhere on the screen, including the hidden area behind the status window. If you try to move it off the screen, the marker will stop at the screen's boundary.

To control the other markers, release the joystick and press the joystick button. Notice that MARKER # lights up in the status window. This is to show you that you are in the process of picking a new current marker. Release the joystick button. MARKER # is no longer lit up, but the number beneath it has changed. It has increased by one, or cycled back from 2 to 0. Also, a different marker now has a pinkish tint. That's the marker that now responds to the joystick. Very soon, you will find the movement of the markers and the switching between them to be quite simple.

Change The Marker Speed

The speed at which the joystick moves the markers across the screen can be changed. Speeds range from 1 to 9. Speed 1 is normal, Speed 2 is twice as fast, and so on up to Speed 9, which is nine times as fast as Speed 1. The higher speeds do not permit you to stop at every point on the screen. These high speeds are used to get across the screen quickly, or to assist in more advanced drawing. To change speed, press the joystick button and move the joystick.

Notice that the highlighted item in the status window changed from MARKER # to SPEED. Move the joystick toward you to decrease speed or away from you to increase speed. You will see the number under SPEED in the status window change as you move the joystick. When you've

reached the desired speed, release the joystick button. Now when you use the joystick, it will move the current marker at the speed you set.

To change the color, release the joystick and press the OPTION key. Notice that in the status window COLOR has lit up. This is to show you that you are in the process of choosing the next color in the sequence. When you release the OPTION key, the next color is shown beneath COLOR. To step through the color sequence, repeatedly press and release OPTION. When the color sequence reaches the last color, it starts again from the first color (the one in effect when you first started). Each version has its own color sequence listed in the table. The colors you actually see may vary, depending upon your computer and the tint adjustment on your TV.

Drawing Colors

Version 1:

ORANGE
GREEN
BLUE
BLACK or erase

Version 2:

GOLD
ORANGE
REDORG (red orange)
PINK
PURPLE
VIOLET
STBLUE (steel blue)
BLUE
BYBLUE (baby blue)
TURQUO (turquoise)
GRBLUE (green blue)
GREEN
YELGRE (yellow green)
ORGGRE (orange green)
LTGREN (light green)
BLACK or erase

Two Fundamental Shapes

To change the shape, release the joystick and press the SELECT key. Notice that in the status window SHAPE has lit up. This is to show you that you are in the process of selecting the other fundamental shape. The two fundamental shapes are TRIANGL (triangle) and CIRCLE. Now, release the SELECT key. The shape underneath SHAPE has changed from TRIANGL to CIRCLE, or from CIRCLE to TRIANGL.

To draw a shape, just press START. The program will take control and draw the shape. When the drawing has completed, control over the keys and joystick returns to you. The amount of time it takes to draw a shape will depend upon its size. A circle takes more time than a triangle, and Version 2 is slower than Version 1. The shape shown under SHAPE is drawn and given the color shown under COLOR. If the shape is TRIANGL, the three markers are its corners. If the shape is CIRCLE, it is drawn using the markers as points along its circumference. As the shape is drawn, it covers (or erases) anything that was in its position on the screen.

SAVEing The Screen

To store the screen display on cassette or disk, or to reload a previously stored screen, press the OPTION and SELECT keys at the same time. The status window is then replaced with the first level of prompt. If you press RETURN, you'll get the status window back. You must press L for LOAD or S for SAVE. Other keys will be rejected and a buzz will sound. Do not press BREAK or SYSTEM RESET.

When you press S or L, the second level of prompt will be shown. Now, you must type a filename such as C for cassette or D:ANDY.GRT for disk. You cannot type more than 15 letters for a filename. Any additional letters or invalid keystrokes will be ignored. Mistakes can be corrected with the backspace key. After typing a filename, press RETURN. If no filename is shown, you will get the status window back. If the filename is invalid, you will see ERROR DETECTED TRY AGAIN for several seconds before the status window returns.

If the filename was correct and you have the disk or cassette set up, the SAVE or LOAD should proceed as explained in the tape or disk manual. When the SAVE or LOAD is complete or interrupted, the status window returns.

Keep The Versions Separate

During a LOAD, the second prompt will come with a warning to use files saved under the current version. A Version 2 screen loaded into Version 1 will result in some striped colors and height distortion. A Version 1 screen loaded into Version 2 will result in different colors and height distortion. Also, it will attempt to LOAD past the end of the file. During a LOAD you will see the screen fill from top to bottom.

Feel free to interrupt a LOAD by pressing BREAK. This is a way to merge the top of a SAVEd screen with the bottom of the current screen. However, pressing BREAK or SYSTEM RESET may cause the program to crash. If this happens, press SYSTEM RESET and type RUN.

Before drawing the shape, the program calculates the numbers it needs from the positions of the markers. For the purpose of positioning, the screen is treated as an X-Y grid with X,Y pairs for each separate point or pixel on the screen. The X can be thought of as column and the Y as row. The upper left corner of the screen is assigned 0,0 and the lower right corner is assigned 159,79 (79, 159 in V2). Then it uses the numbers to draw the shape one row at a time.

Creating A Triangle

Lines 507–540 contain the triangle predrawing section. Line 510 finds the highest (A), middle (B), and lowest (C) markers by comparing the markers' Y coordinates. Line 1550 has the six possibilities for three markers listed out in advance. Lines 530–536 calculate the slopes of the imaginary lines connecting the markers. Lines 11–30 contain the drawing routine. There are two sections divided by a horizontal line at B. In the first section, horizontal lines are drawn from line CA to line BA. In the second section, horizontal lines are drawn from line CA to line CB. The two special cases where AY = BY or BY = CY are also handled.

The circle predrawing section is lines 600–680. The two crucial factors here are the location of the center of the circle (RX,RY), and the radius of the circle (R). The center of the circle is found by using the bisectors rule from geometry. To apply that rule, connect points C and A and points B and A. Then, make lines which pass through the midpoints of lines CA and BA and are perpendicular to CA and BA. We can use the point-slope method to describe these lines. Finally, find where these lines intersect. That is done by solving simultaneous linear equations.

Plotting A Circle

To find the radius, calculate the distance from the center of the circle to point A. In the program, any of the three markers are used as points A, B, and C. The markers are tried in different orders in line 1550 until a center is found.

Notice line 650. The TRAP is there to test for the case where the slopes of the bisectors are equal. This will occur only when the three markers are in a straight line. You can't draw a circle on a straight line. The actual drawing is performed by lines 2–10. It is done by drawing the upper half and the bottom half simultaneously, starting at the equator and going to the poles. X,Y pairs which are outside the screen range are converted to fit on the screen for partial horizontal lines.

Finally, an FT factor is used to make round circles. If you draw circles without using FT, they come out oval. This is because the height of a screen pixel is not equal to its width.

With careful planning, you can construct interesting designs or detailed scenes that have the quality of a watercolor painting. By combining the two fundamental shapes of nature—the circle and the triangle—you can form many other shapes such as rectangles, stars, diamonds, and crescents. The program teaches children drawing composition and the names of the colors.

Drawing A Rectangle

Let's draw a rectangle.

Step 1: Move the markers together until they are exactly on top of one another. This will be the lower left corner of the rectangle.

Step 2: Increase the speed (9 is OK).

Step 3: Move a marker right by tapping the joystick. Count how many taps you make.

Step 4: Do the same thing with another marker but in the up direction.

Step 5: Press START.

Step 6: Move the third marker right and then up the same number of times you counted in steps 3 and 4.

Step 7: Press START and you'll have a rectangle.

Now that you have the general idea, try drawing some shapes on your own.

Here's some advice about circles. Since the markers form the edge of the circle, lining up the markers in a straight line will form a very large circle. In fact, it may not form a circle at all, because you can't draw a curve on a straight line. The computer will buzz at you if you tell it to draw a straight line circle. Move one marker a little and try again. You will see that very large circle. Sometimes circles are partly off and partly on the screen. If the partly off part is drawn first, you may have to wait a few seconds before you see your circle being drawn. Be patient. Soon you will become familiar with how circles are made, so you will know in advance how one will come out before it's drawn.

The Background Comes First

When you draw a scene, remember to do the background first. It is just like painting: The new shape will cover the old. You may notice that certain colors contrast each other and certain colors blend into each other. This and other visual effects can and should be used to your advantage. Remember also that the same color can look different with different backgrounds.

If you see the colors changing after you have been drawing for a while, your Atari is in *attract mode*. The purpose of attract mode is to protect your TV from permanent burn-in of colors. To get your normal colors back, just press the SPACE bar or a letter key.

There is no specific feature for clearing the screen, but it's easy to start with a clean slate. Just move the markers to three of the corners of the screen and draw a BLACK TRIANGL. Then move a marker to the fourth corner from the corner diagonally opposite and draw again.

You may want to modify the program. One simple modification is to use the 16 shades of the GTIA mode. In this mode, the names of the colors should be reinterpreted as shades of gray. In lines 1525 and 1530, change 623 to 65, 87 to 9, and 712 to 0.

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

```
FL 1 GOTO 10000
IC 2 FOR Y=0 TO R:X=SQR(RS-Y*Y):X1=FT*
     (RX-X):X2=FT*(RX+X):Y1=RY-Y:Y2=RY
     +Y
PO 3 IF Y1>YMAX THEN Y1=YMAX
AB 4 IF Y2>YMAX THEN Y2=YMAX
BH 5 X1=X1*(X1>0):IF X1>XMAX THEN X1=-
     1
PP 6 IF X2>XMAX THEN X2=XMAX
OC 7 IF X1<0 OR X2<0 THEN 10
PE 8 IF Y1>=0 THEN PLOT X1,Y1:DRAWTO X
     2,Y1
PI 9 IF Y2>=0 THEN PLOT X1,Y2:DRAWTO X
     2,Y2
PN 10 NEXT Y:GOTO 100
HB 11 IF AY=BY THEN PLOT AX,AY:DRAWTO
      BX,BY:GOTO 20
IA 15 FOR Y=AY TO BY:PLOT CX-(CY-Y)*KC
      A,Y:DRAWTO BX-(BY-Y)*KBA,Y:NEXT
      Y
HI 20 IF BY=CY THEN PLOT BX,BY:DRAWTO
      CX,CY:GOTO 30
IH 25 FOR Y=BY TO CY:PLOT CX-(CY-Y)*KC
      A,Y:DRAWTO CX-(CY-Y)*KCB,Y:NEXT
      Y
CN 30 GOTO 100
NE 100 REM MAIN LOOP
NI 110 IF PEEK(53279)=6 THEN 500
HG 120 ST=STICK(0)
OB 130 IF ST=15 THEN IF PEEK(53279)=3
       THEN 700
OD 132 IF ST=15 THEN IF PEEK(53279)=1
       THEN 900
OJ 135 IF ST=15 THEN IF PEEK(53279)=5
       THEN 800
GF 140 IF ST=15 THEN IF STRIG(0)=0 THE
       N 300
OD 150 XM(MARKER)=XM(MARKER)+SPEED*X(S
       T):YNEW=YM(MARKER)+SPEED*Y(ST)
HA 160 IF XM(MARKER)>206 THEN XM(MARKE
       R)=206
BH 170 IF XM(MARKER)<48 THEN XM(MARKER
       )=48
CE 180 IF YNEW<16 THEN YNEW=16
HP 190 IF YNEW>111 THEN YNEW=111
JD 200 POKE 53252+MARKER,XM(MARKER)
CA 204 IF YNEW=YM(MARKER) THEN 210
OA 205 POKE PMM+YM(MARKER),PEEK(PMM+YM
       (MARKER))-MK(MARKER):POKE PMM+Y
       NEW,PEEK(PMM+YNEW)+MK(MARKER):Y
       M(MARKER)=YNEW
FN 210 GOTO 100
HM 300 S$(3,10)=HEID$(1,8)
BK 320 IF STICK(0)<>15 THEN S$(3,10)=H
       EAD$(1,8):GOTO 400
GA 330 IF STRIG(0)=0 THEN 320
QE 345 POKE 704+MARKER,14
AO 350 MARKER=MARKER+1:IF MARKER=3 THE
       N MARKER=0
OG 355 S$(3,10)=HEAD$(1,8):S$(46,46)=C
       HR$(MARKER+16):POKE 704+MARKER,
       60
GD 360 GOTO 100
OL 400 S$(11,18)=HEID$(9,16)
BN 405 FOR W=1 TO 50:NEXT W
EI 410 IF STICK(0)=15 THEN S$(11,18)=H
       EAD$(9,16):GOTO 100
DH 420 IF STICK(0)=14 THEN SPEED=SPEED
       +1:IF SPEED>9 THEN SPEED=9
CH 430 IF STICK(0)=13 THEN SPEED=SPEED
       -1:IF SPEED<1 THEN SPEED=1
PG 435 S$(55,55)=CHR$(SPEED+16)
GK 440 GOTO 405
GN 500 COLOR COLR:RESTORE 1550
MN 505 IF SHAPE=1 THEN 600
OP 507 TRAP 699:READ A,B,C
IJ 510 IF YM(A)<=YM(B) AND YM(B)<=YM(C
       ) THEN AY=YM(A):AX=XM(A):BY=YM(
       B):BX=XM(B):CY=YM(C):CX=XM(C):G
       OTO 520
GM 511 GOTO 507
NJ 520 AX=(AX-48)*XT:BX=(BX-48)*XT:CX=
       (CX-48)*XT:AY=(AY-16)*YT:BY=(BY
       -16)*YT:CY=(CY-16)*YT
EE 530 TRAP 532:KCA=(CX-AX)/(CY-AY)
EF 532 TRAP 534:KBA=(BX-AX)/(BY-AY)
EP 534 TRAP 536:KCB=(CX-BX)/(CY-BY)
MJ 536 TRAP 40000
DE 540 GOTO 11
OJ 600 TRAP 699:READ A,B,C
DC 601 AX=(XM(A)-48)*XT/FT:AY=(YM(A)-1
       6)*YT
DG 610 BX=(XM(B)-48)*XT/FT:BY=(YM(B)-1
       6)*YT
DL 620 CX=(XM(C)-48)*XT/FT:CY=(YM(C)-1
       6)*YT
II 625 IF CY=AY OR BY=AY OR AX=BX THEN
        600
DK 630 KCA=(AX-CX)/(CY-AY)
DJ 632 KBA=(AX-BX)/(BY-AY)
PB 640 LCA=(CY+AY)/2-KCA*(CX+AX)/2
OP 642 LBA=(BY+AY)/2-KBA*(BX+AX)/2
PE 650 TRAP 699:RY=(KBA*LCA-KCA*LBA)/(
       KBA-KCA)
NN 660 TRAP 40000:RX=(RY-LBA)/KBA
AG 670 RS=(RY-AY)*(RY-AY)+(RX-AX)*(RX-
       AX)
NP 675 R=SQR(RS):IF R>200 THEN 699
AJ 680 GOTO 2
EF 699 TRAP 40000:FOR I=0 TO 30:POKE 5
       3279,0:NEXT I:GOTO 100
CD 700 S$(19,26)=HEID$(17,24)
OA 720 IF PEEK(53279)=3 THEN 720
GG 730 COLR=COLR+1:IF COLR=NCOLRS THEN
        COLR=0
AJ 740 S$(19,26)=HEAD$(17,24):S$(60,65
       )=COLR$(COLR*6+1,COLR*6+6)
GG 750 GOTO 100
CA 800 S$(27,34)=HEID$(25,32)
OE 820 IF PEEK(53279)=5 THEN 820
BI 830 SHAPE=1-SHAPE
NE 840 S$(27,34)=HEAD$(25,32):S$(68,74
       )=SHAPE$(SHAPE*7+1,SHAPE*7+7)
GH 850 GOTO 100
GA 900 SS$=S$:POKE 764,255:CLOSE #2:OP
       EN #2,4,0,"K:":POKE 702,64:POKE
        694,0
HH 910 S$="  PRESS [■] TO LOAD SCREEN FR
       OM FILE{8 SPACES}PRESS [S] TO SAV
       E SCREEN TO FILE{8 SPACES}"
JF 920 FOR I=1 TO 80:S$(I,I)=CHR$(ASC(
       S$(I,I))-32):NEXT I
JO 925 GOSUB 2000:IF A=155 THEN 999
FN 926 IF A=ASC("L") THEN W=4:B=7:GOTO
        940
JE 927 IF A=ASC("S") THEN W=8:B=11:GOT
       O 940
OI 930 FOR I=1 TO 25:POKE 53279,0:NEXT
        I:GOTO 925
JD 940 S$="  FILE NAME ?{67 SPACES}":FI
       LE$=S$(14,28)
BL 941 IF W=4 THEN S$(42,65)="FILE MUS
       T BE FOR VERSION":S$(67,67)=CHR
       $(ASC("0")+V)
```

```
JK 943 FOR I=1 TO 80:S$(I,I)=CHR$(ASC(
       S$(I,I))-32):NEXT I
FE 950 I=0
CK 955 GOSUB 2000:IF A=155 AND I=0 THE
       N 999
BH 960 IF A=155 THEN 980
KF 965 IF A=126 AND I=0 THEN 955
EF 968 IF A=126 THEN A=ASC(" "):GOSUB
       978:I=I-1:GOTO 955
ON 970 IF I=15 THEN 955
EI 975 I=I+1:GOSUB 978:GOTO 955
CH 978 FILE$(I,I)=CHR$(A):S$(14+I,14+I
       )=CHR$(A-32):RETURN
IJ 980 TRAP 997:POKE 54286,64
PD 981 IF FILE$(1,1)="C" THEN POKE 537
       75,35:POKE 53768,40:POKE 53764,
       0:POKE 53766,0:POKE 53773,255
HI 982 OPEN #1,W,0,FILE$
JN 985 POKE 852,PEEK(88):POKE 853,PEEK
       (89):POKE 856,0:POKE 857,15*V:P
       OKE 850,B
LB 990 B=USR(ADR(CIO$)):GOTO 999
LL 997 TRAP 40000:POKE 54286,192:S$(42
       ,67)="ERROR DETECTED TRY AGAIN
        "
NM 998 FOR I=42 TO 67:S$(I,I)=CHR$(ASC
       (S$(I,I))-32):NEXT I:FOR I=1 TO
        1000:NEXT I
NC 999 TRAP 40000:S$=SS$:CLOSE #2:CLOS
       E #1:GOTO 1520
BN 1000 IF V=2 THEN GRAPHICS 24:PM=PEE
       K(106)-36:NCOLRS=16:XT=0.5:YT=
       2:FT=0.3125:XMAX=79:YMAX=191
BO 1001 IF V=1 THEN GRAPHICS 23:PM=PEE
       K(106)-20:NCOLRS=4:XT=1:YT=1:F
       T=1.25:XMAX=159:YMAX=95
MH 1003 POKE 54279,PM:PMM=PM*256+384:P
       OKE 559,38:POKE 53277,1
JP 1004 POKE 623,1:FOR I=PMM TO PMM+12
       7:POKE I,0:NEXT I
AD 1005 FOR I=0 TO 2:POKE 704+I,14:NEX
       T I
BH 1010 DIM XM(2),YM(2),MK(2),COLR$(96
       ),SHAPE$(14)
LC 1011 FOR I=0 TO 2:XM(I)=125:POKE 53
       252+I,125:YM(I)=16:NEXT I:POKE
        PMM+16,255
PC 1012 MK(0)=3:MK(1)=12:MK(2)=48:REM
       MISSILE MASKS
KP 1013 COLR$="BLACK GOLD  ORANGEREDOR
       GPINK  PURPLEVIOLETSTBLUE BLUE
        BYBLUETURQUOGRBLUEGREEN YELGR
       EORGGRELTGREN"
PG 1014 SHAPE$="TRIANGLCIRCLE ":IF V=1
        THEN COLR$(1,24)="BLACK ORANG
       EGREEN BLUE  "
IC 1015 FOR I=1 TO 96:COLR$(I,I)=CHR$(
       ASC(COLR$(I,I))-32):NEXT I
PL 1016 FOR I=1 TO 14:SHAPE$(I,I)=CHR$
       (ASC(SHAPE$(I,I))-32):NEXT I
JB 1020 SHAPE=0:COLR=1:MARKER=0:SPEED=
       1:POKE 704,60
JM 1021 DIM S$(80):S=ADR(S$):SH=INT(S/
       256):SL=S-SH*256
JA 1022 S$=" ":S$(80)=" ":S$(2)=S$
MA 1023 FOR I=1 TO 80:S$(I,I)=CHR$(ASC
       (S$(I,I))-32):NEXT I
MF 1025 DIM HEAD$(32):HEAD$="MARKER #
        SPEED  COLOR(3 SPACES)SHAPE
       "
BG 1026 DIM HEID$(32):HEID$="MARKER #
        SPEED  COLOR(3 SPACES)SHAPE
       "
DP 1027 FOR I=1 TO 32:HEAD$(I,I)=CHR$(
       ASC(HEAD$(I,I))-32):NEXT I
FA 1028 FOR I=1 TO 32:HEID$(I,I)=CHR$(
       ASC(HEID$(I,I))-32):NEXT I
DA 1029 S$(3,34)=HEAD$:S$(46,46)=CHR$(
       MARKER+16):S$(55,55)=CHR$(SPEE
       D+16):S$(60,65)=COLR$(COLR*6+1
       ,COLR*6+6)
EC 1030 S$(68,74)=SHAPE$(SHAPE*7+1,SHA
       PE*7+7)
OP 1034 DIM X(15),Y(15):FOR I=5 TO 15:
       READ A,B:X(I)=A:Y(I)=B:NEXT I
LJ 1035 DATA 1,1,1,-1,1,0,0,0,-1,1,-1,
       -1,-1,0,0,0,0,1,0,-1,0,0
EE 1100 DIM CIO$(6):FOR I=1 TO 6:READ
       A:CIO$(I,I)=CHR$(A):NEXT I
MJ 1105 DATA 104,162,16,76,86,228
EN 1110 DIM SS$(80),FILE$(15)
KC 1500 DIM DLI$(14):FOR I=1 TO 14:REA
       D A:DLI$(I,I)=CHR$(A):NEXT I
KC 1501 DATA 72,173,111,2,41,3,141,10,
       212,141,27,208,104,64
DB 1502 DL=PEEK(560)+PEEK(561)*256:IF
       V=1 THEN 1514
AD 1503 POKE DL+182,143:POKE DL+183,66
       :POKE DL+186,2:POKE DL+187,PEE
       K(DL+199):POKE DL+188,PEEK(DL+
       200)
JG 1513 POKE DL+189,PEEK(DL+201):POKE
       DL+184,SL:POKE DL+185,SH:GOTO
       1520
LC 1514 POKE DL+93,66:POKE DL+96,2:POK
       E DL+97,PEEK(DL+101):POKE DL+9
       8,PEEK(DL+102)
AJ 1515 POKE DL+99,PEEK(DL+103):POKE D
       L+94,SL:POKE DL+95,SH:GOTO 152
       0
DO 1520 POKE 513,INT(ADR(DLI$)/256):PO
       KE 512,ADR(DLI$)-256*PEEK(513)
MJ 1521 POKE 54286,192
EJ 1525 IF V=2 THEN POKE 623,193:POKE
       87,11
HI 1530 IF V=2 THEN POKE 712,8:POKE 71
       0,8:POKE 709,14
JE 1540 GOTO 100
ED 1550 DATA 0,1,2,2,1,0,0,2,1,1,2,0,1
       ,0,2,2,0,1
DL 2000 A=PEEK(764):IF A=255 OR A=60 O
       R A=39 THEN 2000
BC 2005 GET #2,A:IF A=126 OR A=155 THE
       N RETURN
PB 2010 IF A<32 OR A>=96 THEN 2000
KI 2015 RETURN
AM 10000 GRAPHICS 17:POKE 752,1:V=1
NE 10001 DL=PEEK(560)+256*PEEK(561)
BI 10002 POKE DL+3,71:FOR I=6 TO 11:PO
       KE DL+I,7:NEXT I
FI 10010 POSITION 7,1:? #6;"WELCOME"
FM 10020 POSITION 10,3:? #6;"to"
DG 10030 POSITION 2,5:? #6;"SHAPES and
        COLORs"
IJ 10040 POSITION 0,10:? #6;"OPTION  C
       HANGE COLORS"
GE 10050 POSITION 0,11:? #6;"SELECT  C
       HANGE SHAPES"
FP 10055 POSITION 0,12:? #6;"START
       (3 SPACES)TO DRAW"
HH 10060 POSITION 0,15:? #6;"move stic
       k to begin";
DI 10065 POSITION 0,14:? #6;"button fo
       r version ";V;
FL 10066 IF STICK(0)<>15 THEN 1000
HM 10067 IF STRIG(0)=0 THEN V=3-V
DD 10068 IF STRIG(0)=0 THEN 10068
DF 10069 GOTO 10065
```

C. Regena

Programming Techniques In TI BASIC

This month, by answering some of the common questions I have received from readers, I'm going to give you a variety of programming techniques that you can use in your own programs.

How do you clear part of a screen?

Let's say you have onscreen a nice picture with a description underneath. CALL CLEAR will clear the whole screen; but you want to clear the printing, not the picture. Use CALL HCHAR with the row and column parameters under the picture, and use the number of repetitions that will clear the section you want. For example, to clear the lower half of the screen, CALL HCHAR(13,1,32,32*12). We're starting with row 13, column 1, and clearing with the space (character code 32) for 32*12 squares—32 columns times 12 more rows.

To clear with a different color, redefine a character (in a color set you are not using) as a colored square, then use CALL HCHAR to put that character on the screen:

```
300 CALL COLOR(13,16,16)
310 CALL HCHAR(13,1,128,32*12)
```

To clear a vertical section of the screen, use CALL VCHAR:

```
CALL VCHAR(1,17,32,24*16)
```

To try out this technique, try this sample program:

```
100 CALL HCHAR(1,1,42,32*24)
110 CALL HCHAR(13,1,32,32*12)
900 GOTO 900
```

Change line 110 to the CALL VCHAR statement above and try the program. Next take out line 110 and put in lines 300 and 310 listed above. Experiment with different numbers of repetitions.

How do you get a border around the screen?

CALL SCREEN(*c*), where *c* is a number from 1 to 16, defines the screen color. When you use this statement in a program, the whole screen instantly changes color. CALL COLOR(*s*,*f*,*b*) defines the character colors. The characters are divided into sets of eight characters each. The *s* in the parentheses is the set number and can be from 1 to 16. The *f* is the foreground color of the character, *b* the background color, and they can be one of the 16 color numbers, from 1 to 16.

Now take a look at the characters in set 1. The space is code 32 in set 1. The screen is filled with spaces wherever there isn't any printing or graphics. If you change the color of set 1 to something other than the screen color (background color 1), you'll get color where all the spaces are.

```
100 CALL CLEAR
110 CALL SCREEN(14)
120 CALL COLOR(1,2,16)
900 GOTO 900
```

Press FCTN 4 (CLEAR) to stop the program. You've got a border on the top and on the bottom, but you would like the sides also. When we PRINT messages we have a 28-column line, but when we do graphics we actually have 32 columns—there are two columns on each side of the regular printing section. They currently have spaces in them. To get the screen color in those columns, add

```
115 PRINT ::::::::::::::::::::::::::
```

Or, as you print messages, those extra columns fill with the screen color. (As you PRINT, columns 1, 2, 31, and 32 will contain character 31.) A quicker way to get rid of the spaces in those columns is to fill the columns with a character in the screen color. You may add these lines instead:

```
115 CALL CHAR(152,"")
116 CALL VCHAR(1,1,152,48)
117 CALL VCHAR(1,31,152,48)
```

Now try a few PRINT messages, such as

```
150 PRINT "HELLO"
```

Notice that the letters have little squares of the screen color around them. All the color sets are automatically defined as CALL COLOR(S,2,1), which is black with a transparent background. The color number 1, transparent, will be the screen color. If you want the printing to be black on your inner screen color (the color of the spaces), you need to define the sets with the background color that you used in set 1. Change line 120 above to

```
120 FOR S=1 TO 12
130 CALL COLOR(S,2,16)
140 NEXT S
```

This defines a white background for the first 12 character sets, those sets which have letters and symbols. Now run the program and you will see that the message no longer has the screen color background.

How do you make a simple math drill with graphics?

I have had quite a few requests for an arithmetic drill program. Many readers would like to develop such programs on their own and want to know how to draw a certain number of pictures for the numbers chosen randomly in a simple math problem.

Here is a short program to give you the general idea of using the graphics. I defined character 128 to be the picture. The variables A and B can be numbers from zero to four. Lines 170–200 print the problem on the screen—a simple addition problem. Lines 210 and 220 draw the right number of characters for A and B.

Program 1: Simple Math Drill

```
100 REM  SIMPLE MATH
110 CALL CLEAR
120 CALL CHAR(128,"0024002418")
130 CALL COLOR(13,2,11)
140 RANDOMIZE
150 A=INT(5*RND)
160 B=INT(5*RND)
170 CALL HCHAR(8,10,A+48)
180 CALL HCHAR(10,8,43)
190 CALL HCHAR(10,10,B+48)
200 CALL HCHAR(11,8,95,3)
210 CALL HCHAR(8,12,128,A)
220 CALL HCHAR(10,12,128,B)
230 CALL SOUND(150,1497,4)
240 CALL KEY(0,K,S)
250 IF S<1 THEN 240
260 IF K=32 THEN 400
270 IF K=A+B+48 THEN 310
280 CALL SOUND(100,330,2)
290 CALL SOUND(100,262,2)
300 GOTO 240
310 CALL HCHAR(13,10,K)
320 PRINT "CORRECT!"
330 CALL SOUND(100,262,2)
340 CALL SOUND(100,330,2)
350 CALL SOUND(100,392,2)
360 CALL SOUND(200,532,2)
370 CALL SOUND(1,9999,30)
380 CALL CLEAR
390 GOTO 140
400 CALL CLEAR
410 END
```

If you prefer to have a space between graphics characters, place a character in every other space. You can do this by changing lines 210 and 220 above to the following:

```
210 FOR C=12 TO 12+2*(A-1) STEP2
212 CALL HCHAR(8,C,128)
214 NEXT C
220 FOR C=12 TO 12+2*(B-1) STEP2
222 CALL HCHAR(10,C,128)
224 NEXT C
```

In this sample program, an addition problem is presented and the student answers by pressing a number. If it is incorrect, there is an "uh-oh" sound. If it is correct, an arpeggio is played and the computer goes to the next problem. To stop, press the space bar.

How can you draw a bar graph?

This procedure is similar to the previous sample program. The easiest way to draw a bar graph is to use HCHAR with the appropriate number of repetitions (or VCHAR). You may need to scale the numbers. Take the highest number you'll need to graph, relate it to the greatest number of repetitions you can have in your HCHAR statement, and stay on that row.

Another method is to use PRINT and print the right number of characters for the bar. The following sample program segment demonstrates this method. Character 128 will be a red square. For purposes of illustration, I will use random numbers N up to 90 for the amounts to be graphed. You would probably have specific numbers that have been calculated or read in from DATA.

A is the scaled value (rounded) for N—for every four units one square can be drawn. Line 170 prints the number N then says to start the next printing in the fifth print column. Lines 180–200 print the appropriate number of red squares.

Program 2: Bar Graph Generator

```
100 REM  BAR  GRAPH
110 CALL CLEAR
120 CALL COLOR(13,7,7)
130 FOR I=1 TO 10
140 RANDOMIZE
150 N=INT(90*RND)
160 A=INT(N/4+.5)
170 PRINT N;TAB(5);
180 FOR B=1 TO A
190 PRINT CHR$(128);
```

```
200 NEXT B
210 PRINT ::
220 NEXT I
230 GOTO 230
240 END
```

How do you print a list of items in more than two columns?

As you know, the comma in PRINT statements prints items in two columns—items start either in the first print position or the center position. To get three columns or more, use the TAB function. TAB works like the tab key on a typewriter. You may specify which column to start printing. TAB(7) would start the next print item in the seventh print column. Here's a sample that types three columns of names.

```
100 CALL CLEAR
110 READ L$,M$,N$
120 IF L$="@" THEN 180
130 PRINT L$;TAB(10);M$;TAB(19);N$
140 GOTO 110
150 DATA MIKE,BOB,DICK,RICH
160 DATA JIM,JERRY,MARY,PAULA
170 DATA CHRIS,KEVIN,KATHY,KIRK,@,@
    ,@
180 END
```

How can you print a screen without seeing the scrolling?

Some people don't like to see scrolling as they print. Messages on the TI are always printed on the twenty-fourth row then moved upward. To block this motion, change the screen to black first (because the printing is black), print the messages, then change the screen back to a different color so you can read the printing.

```
100 CALL CLEAR
110 CALL SCREEN(2)
120 PRINT "THIS IS AN EXAMPLE"
130 PRINT ::"TO SEE A SCREEN"
140 PRINT ::"ALL AT ONCE.":::::
150 CALL SCREEN(4)
160 GOTO 160
```

How can you print what is on the screen to the printer?

I'm sorry, but I don't know how to do a *screen dump* of graphics because none of the printers I have right now has the graphics capabilities. You will need to look at your own brand printer manual to see how to use the dot-addressable graphics. If you have a screen of printing, however, with regular printed symbols, you can use the following procedure. The character in each row and column is determined, then that character is printed on the printer. You may need to change the OPEN statement in line 100 to suit your particular printer configuration.

```
100 OPEN #1:"RS232.BA=600"
110 FOR ROW=1 TO 24
120 FOR COL=3 TO 30
130 CALL GCHAR(ROW,COL,G)
140 PRINT #1:CHR$(G);
150 NEXT COL
160 PRINT #1
170 NEXT ROW
180 CLOSE #1
190 END
```

If you want everything you are printing to go both to the screen and to the printer, use both a PRINT statement and a PRINT #1 statement for items printed.

```
100 OPEN #1:"RS232.BA=600"
110 CALL CLEAR
120 PRINT #1:CHR$(12)
130 PRINT "HELLO"
140 PRINT #1:"HELLO"
150 PRINT "ANY MESSAGE"
160 PRINT #1:"ANY MESSAGE"
170 CLOSE #1
180 END
```

Line 120 above goes to the top of a page.

How can you simulate time on the TI?

If you need an exact time, use the CALL SOUND statement in which you can specify an exact duration in milliseconds. If you don't want to hear the sound, use a high frequency and the softest volume.

```
100 PRINT "START"
110 CALL SOUND(1000,9999,30)
120 CALL SOUND(1,9999,30)
130 PRINT "END"
140 END
```

Line 120 is necessary to end the first sound.

If you want to time someone as they are pressing keys to move or are answering a question, use a counter in your CALL KEY loop. You can't relate this counter to an exact time because in each program it will be different—depending on how you do the programming, how long your program is, and how full the memory is. However, once you have your program working, you can print the counter value and use a stopwatch to figure out a formula that relates the actual time to the counter value. ("Type-ette Timer" in my *Programmer's Reference Guide to the TI-99/4A* from COMPUTE! Books uses this technique to time how fast you can type sentences.) Here is a sample:

```
100 T=0
110 CALL KEY(0,K,S)
120 T=T+1
130 IF S<1 THEN 110
140 PRINT T
150 GOTO 100
160 END
```

The faster you press a key, the lower the value for T will be. The longer you wait, the more times the computer will go through the loop and increment T.

Other computers use PRINT AT; how can we do it?

In TI Extended BASIC you can specify a row and column to begin printing an item. However, we don't have that feature in regular console BASIC on the TI. There are several ways to accomplish this, though they're slower than regular printing. First, you can use the regular PRINT statement, perhaps with the TAB function, and then use colons to move the message up to the proper row.

```
PRINT TAB(9);"START PRINTING"::::::
```

The main problem with this method is that it scrolls the screen. If I am labeling graphics, I do all the printing first, then use CALL HCHAR and CALL VCHAR to put up the graphics.

Another method is to treat the letters in the printed message as graphics characters, and use CALL HCHAR to specify the row and column to place the letters on the screen. Here's a general-purpose subroutine that you can use. M$ is the message you want printed, R is the row, and C is the column you want the message to start in.

```
300 FOR L=1 TO LEN(M$)
310 CALL HCHAR(R,C-1+L,ASC(SEG$(M$,
    L,1)))
320 NEXT L
330 RETURN
```

Before you call the subroutine with a GOSUB, specify a row R and a column C and the message M$:

```
900 M$="TEST PRINTING"
910 R=6
920 C=12
930 GOSUB 300
```

How can I put a code in my program?

I have had lots of young people ask me how they can write a program so that whoever runs it must enter a code before the program continues—they don't want their brothers and sisters using their program. The general idea is that you put a code name in the program as a string variable. Next, use INPUT for the user who is running the program to type in the code. Now compare the INPUT value with the code to see whether to continue or not.

```
100 CALL CLEAR
110 CODE$="RANDY"
120 INPUT "ENTER CODE NAME: ":A$
130 IF A$=CODE$ THEN 160
140 PRINT ::"SORRY, INVALID CODE."
150 STOP
160 REM PROGRAM CONTINUES
```

The only problem with specifying the code in line 110 is that anyone can load the program, then LIST it to find out what the code name is. One method I use so people can't read the code name is to hold down the CTRL key (key with the red dot) while you type your code message. Line 110 will now look like this:

```
110 CODE$="    "
```

or you may get some funny-looking graphics characters between the quotes. Now when someone lists your program, they can't tell what the code name is. When *you* run the program, be sure to hold the CTRL key down when you INPUT the code name, and it will match the code in the program.

A Couple Of Warnings

Always use the SHIFT key on the left side of the keyboard to type the plus sign. You don't want to go for the right SHIFT key and accidentally hit the FCTN key—and *quit*!

Do *not* use TI Extended BASIC to run regular TI BASIC programs because they may not run properly. One reason is the double colon used in PRINT statements, and another reason is that I often use graphics in character sets 15 and 16, which are not available in Extended BASIC.

If you have a disk drive attached to your computer, the disk uses up some memory. For any of my published programs, type in CALL FILES(1) and press ENTER, then type NEW and press ENTER, then proceed normally (load a program or start typing a program). This procedure clears about 1000 bytes of memory so a program can fit.

Until Next Time ...

I hope these ideas help you in your programming. Your computer can be a lot of fun. Part of the joy of programming is getting that machine to do what you want it to do. As I continue these columns I hope to present a variety of programs so you can see that this computer is really quite versatile. Your suggestions and letters are always welcome. ©

Programming 64 Sound
Part 2

John Michael Lane

Last month in Part 1, we discussed sound and music in general. This month we examine some techniques for programming more complicated music using the 64's SID chip.

The control register is the most complex register in the chip. *Each* of the eight bits in this register has a different function. Dealing with individual bits within a one-byte register is often a problem for BASIC programmers. One very easy way to approach the problem is to use the following:

```
170 B(0)=1
180 B(1)=0
190 B(2)=1
200 B(3)=0
210 B(4)=0
220 B(5)=0
230 B(6)=0
240 B(7)=1
250 FOR I=0 TO 7
260 Q=Q+B(I)*2↑I
270 NEXT I:POKE S+4,Q
```

This is not efficient programming, but by defining the bits we want (that is, B(I) where I= the bit number) in terms of a 1 and those we don't want in terms of a 0, this segment will work. It will be somewhat slow and cannot be used in a loop that must execute quickly, which is usually the case when doing musical programming.

A quicker method is to think of the bits in terms of their value in an eight-bit binary number. Bit 0 has a value of 1, bit 1=2, bit 2=4, bit 3=8, bit 4=16, bit 5=32, bit 6=64, and bit 7=128. In the case above, we want to set bits 0, 2, and 7 *on*, so we simply add their values: 1+4+128=133. Simply POKE 133 into the register to set those bits. It's much simpler, but requires you to add up the bit values before writing the program, so when you look back on the program one month later you may not have the slightest idea why you chose 133.

The first bit of the control register, bit 0, acts as the gate to turn the sound on and off. Remember that when the gate is opened (when bit 0 is set to 1), the attack phase of the volume envelope begins. When the gate is closed (bit 0 is set to 0), the release phase of the volume envelope is triggered. If the gate is closed prematurely, the sustain, decay, and even a portion of the attack phase may be omitted. Opening and closing the gate is actually very easy. Just remember that POKEing an odd value in register 4 turns the gate on and that POKEing an even value into the register turns the gate off.

Watch The Timing

Be careful of turning the gate off by POKEing zero into the register. That will also clear the waveform bits (which we'll discuss in a second) and will result in your volume envelope having no release phase.

The next bit, bit 1, is the *sync* bit. If this bit is on, the output from voice 1 will be synchronized with the output from voice 3. *Sync* in this case means that the output of voice 1 will be replaced with a logical AND of the output of voice 1 and voice 3. Another way to think of it is that voice 1 is turned on and off with the frequency of voice 3. In order for this bit to have any effect, oscillator three (voice 3) must be set to some frequency less than voice 1. The best way to understand this effect is to listen to it. "Laser" (Program 4) contains a demonstration using the sync bit. When using sync, the lower frequency will predominate. The effect works best when the lower frequency is 1/10 to 1/2 of the higher.

The sync bit has a slightly different effect in the other two voices. In voice 2 it produces a

sync of voice 2 with voice 1, and in voice 3 it produces a sync of voice 3 with voice 2.

The next bit, bit 2, is the *ring modulation* bit. When this bit is set on, it produces nonharmonic overtones that sound like a bell. In order for this effect to take place, the triangular waveform must be selected for voice 1, and voice 3 must have a frequency other than zero.

Ring modulation in the other voices works like the sync bit; that is, for voice 2 to be ring modulated, voice 1 must have a nonzero frequency. For voice 3, voice 2 must be nonzero. In all cases the triangular waveform must be selected for the affected voice.

Bit 3 in the control register is the test bit. Setting the test bit to one will turn off the sound generator. This technique will generally be used only by machine language programmers.

Bits 4–7 are the waveform bits. Turning on bit 4 will select the triangular waveform; bit 5 will select the sawtooth; bit 6, the rectangular pulse; and bit 7, white noise (the hissing sound that you hear between stations on a radio).

At this point you must be asking yourself "What happens if more than one bit is selected?" The answer is that the two (or more) waveforms will be ANDed together (a logical AND will be done on the waveforms). Commodore cautions that selecting more than one waveform while using the white noise waveform could cause the oscillator to go silent, so don't combine waveforms using the white noise waveform. Even while avoiding the white noise waveform, it is still possible to generate four more waveform shapes using combinations of the sawtooth, triangular, and rectangular pulse waveforms. However, the volume declines significantly when combining waveforms.

Register 5 contains the attack and decay values for the sound envelope. The four-bit attack value is held in bits 7–4. The four-bit decay value is held in bits 3–0. The values can be loaded like this:

```
300 A=13:D=5: REM ATTACK=13,DECAY=5
310 POKE S+5,16*A+D
```

Register 6 contains the sustain level and the release value. As above, the sustain level is held in bits 7–4, and the release value in bits 3–0. Program them in the following manner:

```
320 SU=13:R=4: REM SUSTAIN=13,RELEASE=4
330 POKE S+6,16*SU+R
```

Now we've completely covered the seven register groups and shown how to load them. "Twiddle" (Program 1) allows you to explore all possible combinations using these seven registers. The program allows you to set and change any of the values and then listen to an eight-note scale governed by those values. If you can sit down and play with the program for a couple of hours, you'll gain a good understanding of how changing SID parameters affects a sound. The program is also useful for demonstrating how to play a tune within a basic program.

From Sound To Music

To play actual music, you generally write a program which will load all the parameters except the waveform and the frequency. At this point you select the note to be played and POKE the appropriate values into the frequency registers. Then you POKE the waveform value plus one (16+1=17 for triangular, 33 for sawtooth, 65 for the rectangular pulse, and 129 for white noise) into register 4 (the control register). Adding a 1 causes the gate bit (bit 0) to be turned on and the tone begins. The program waits a certain period of time and then POKEs the waveform value (16, 32, 64, or 128) into register 4. By POKEing an even number into the register we turn the gate off, and the note begins its release phase and gradually dies out (according to the release value that you've set).

A simple way to time the note is to use a delay loop. An empty loop (like the one below) will execute 1000 cycles in just about one second.

```
400 FOR I=1 TO 1000:NEXT I
```

Therefore, each cycle is just about 1/1000 second (or a millisecond). To turn the note on and off, the program line will look like this:

```
400 POKE S+4,17:FOR I=1 TO 250:
    NEXT:POKES+4,16
```

The above program line will play a note for about one quarter of a second.

This technique works well for a single voice, but it may not work at all for more than one voice. The problem is that while the computer is timing the duration of one note, it cannot be separately timing voices 2 and 3. We could fill the empty loop with timing routines for voices 2 and 3, but that would change the execution time for the loop and throw the timing off.

A second technique is to use the internal timer of the Commodore 64 through the use of the variable TI. The variable TI is updated automatically on the Commodore 64 and increases by a value of one every 1/60 second. We can use this timer to time the duration of our notes:

```
500 T0=TI: REM INITIALIZE THE VARIABLE "T
    0"
510 T0=T0+D: REM INCREASE "T0" BY DURATIO
    N OF THE FIRST NOTE - D
520 IF T0<=TI THEN GOSUB 1100: REM CHECK
    {SPACE}IF THE TIME IS UP
525 REM IF SO SUBROUTINE 1100 WILL CHANGE
     NOTES
530 GOTO 520: REM IF NOT CHECK TIME AGAIN
```

The key to using this routine is to make sure that the subroutine executes quickly, at least while using multiple voices. "Tune" (Program 2) illustrates this technique using all three voices. But this method isn't problem-free either. We want to reproduce the rhythm of the original tune as accurately as possible. It's physically impossible to change the frequency of all three voices at once. Using BASIC, it's somewhat difficult to change all three voices in less than 1/6 second. For that reason, we split all the frequencies into the higher and lower order bytes before the tune begins. We can then change the frequency of all three voices in about 1/10 second. For most tunes that will be satisfactory. However, for a fast tempo, you might have to omit the second or third voice in order to maintain the rapid changes of the first voice.

Sound Effects

Now, let's briefly explore the sound of a laser firing, or an explosion, siren, or any other sound we need. How can we accomplish it?

There is no direct way. The best approach is trial and error. Listen to the sound carefully. Most sounds in nature cannot be duplicated simply by selecting the right waveform and envelope. Generally, the frequency is also actively changing during the sound's life. While you listen to (or think about) the sound you want, consider what is happening to the frequency. Is it rising or falling? How quickly?

Also consider the volume. Many volume envelopes cannot be duplicated using the attack/decay/sustain/release envelope on the Commodore SID. You will often have to change the volume level through program control, using the volume register (register 24) on the SID.

Programs 3 and 4, "Blast-off" and "Laser," illustrate one approach. In Blast-off, both the frequency and volume are modulated by the program. Laser demonstrates the sync feature and modulates the frequency to produce the laser sound. Both programs were written after much trial and error.

Many authors, when converting programs to the 64, simply drop the sound effects or stop at a sound which is only vaguely like the one they want. Be persistent; the 64 can accurately produce almost any sound. As you gain experience, you will find that the trial and error phase will decrease significantly.

Twiddle illustrates the basic methods of loading the SID registers and lets you experiment by changing the waveform and ADSR envelope while listening to the musical scale.

Tune uses the three voices to play an English folk tune. Don't be discouraged by the long list of DATA statements. Voice 1 repeats the same statements four times, and there is considerable repetition in voices 2 and 3. Once you've typed in the few basic lines, you can simply change the line numbers with the screen editor to produce the remainder of the data statements.

Tune can be used to produce any melody by changing the values in the DATA statements. Each note is represented by a pair of values. The first represents the duration of the note (in sixtieths of a second). A value of 30–40 is appropriate for a quarter note. The second value is the frequency of the note. Appendix E in the *Commodore 64 Programmer's Reference Guide* offers a good, simple frequency table. Below are the values for the 12-semitone scale starting at middle C.

C - 4291	C# - 4547
D - 4817	D# - 5103
E - 5407	
F - 5728	F# - 6069
G - 6430	G# - 6812
A - 7217	A# - 7647
B - 8101	

Notes for other octaves can be calculated by doubling or halving these values, depending upon whether you're going one octave up (doubling) or one octave down (halving).

It is useful to convert one measure of music to one DATA statement if you can. This makes it easier to match the voices.

Voice 1 is the sound of a flute, voice 2 is a mandolin, and voice 3 is a guitar. Blast-off and Laser are supposed to produce the sound of their titles. They are pretty straightforward.

Program 1: Twiddle

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

```
5 S=54272                                :rem 201
7 DIM A(15),D(15)                         :rem 48
10 FORL=STOS+24:POKEL,0:NEXT              :rem 53
15 GOSUB 1000                            :rem 167
17 GOSUB 1100                            :rem 170
18 GOSUB 1200                            :rem 172
20 PRINT"{CLR}";TAB(5);"TOUCH W FOR WAVEF
   ORM"                                    :rem 5
30 PRINT TAB(5)"TOUCH A FOR ATTACK RATE"
                                           :rem 32
40 PRINT TAB(5)"TOUCH S FOR SUSTAIN LEVEL
   "                                      :rem 238
45 PRINT TAB(5)"TOUCH T FOR SUSTAIN TIME"
                                          :rem 171
50 PRINT TAB(5)"TOUCH R FOR RELEASE"
                                           :rem 80
60 PRINT TAB(5)"TOUCH D FOR DECAY"
                                          :rem 168
70 PRINT TAB(5)"TOUCH P FOR PULSE WIDTH"
                                           :rem 88
72 PRINT TAB(5)"TOUCH B TO SET DEAD TIME"
                                           :rem 40
75 PRINT TAB(5)"TOUCH + OR - FOR FREQUENC
   Y CHANGE"                               :rem 85
80 GET A$:IF A$=""THEN80                 :rem 243
82 IF A$="W"THEN 200                     :rem 247
```

```
84 IF A$="A" THEN 250              :rem 232
86 IF A$="S" THEN 300              :rem 248
88 IF A$="R" THEN 350              :rem 254
90 IF A$="D" THEN 400              :rem 229
92 IF A$="P" THEN 450              :rem 248
94 IF A$="T" THEN 500              :rem 250
96 IF A$="+" THEN GOSUB 1400       :rem 131
97 IF A$="B" THEN 550              :rem 240
98 IF A$="-" THEN GOSUB 1450       :rem 140
100 REM                            :rem 117
105 POKE S+24,15                    :rem 59
110 POKE S+5,16*A+D                :rem 225
120 POKE S+6,16*SL+R                :rem 79
130 POKE S+3,INT(P/256)            :rem 248
140 POKE S+2,P-256*INT(P/256)       :rem 60
150 FOR I=1 TO 8                    :rem 15
160 IFINT(F(I))<=65536THENPOKE S+1,INT(F(
    I)/256)                        :rem 229
170 POKE S,F(I)-256*INT(F(I)/256)  :rem 2
180 IFINT(F(I))<=65536THENPOKE S+4,2↑(W+3
    )+1                            :rem 244
185 FORJ=1TOT:NEXT                 :rem 173
187 POKE S+4,2↑(W+3)                :rem 67
188 FORJ=1TOB:NEXT                 :rem 158
190 NEXT I:GOTO 20                 :rem 247
200 PRINT"WAVEFORM IS";" - ";W     :rem 164
202 PRINT"1=TRIANGLE"               :rem 41
204 PRINT"2=SAWTOOTH"               :rem 79
206 PRINT"3=PULSE"                  :rem 98
208 PRINT"4=NOISE"                  :rem 90
210 INPUT"ENTER WAVEFORM (1-4)";W:rem 193
215 IFW<1 ORW>4THEN210              :rem 23
220 GOTO 100                        :rem 94
250 PRINT"ATTACK RATE IS";A        :rem 100
260 INPUT"ENTER ATTACK RATE (0-15)";A
                                   :rem 94
265 IFA<0ORA>15THEN260              :rem 38
270 GOTO 100                        :rem 99
300 PRINT"SUSTAIN LEVEL IS";SL     :rem 121
310 INPUT"ENTER SUSTAIN LEVEL (0-15)";SL
                                   :rem 115
315 IFSL<0ORSL>15THEN310           :rem 218
320 GOTO 100                        :rem 95
350 PRINT"RELEASE RATE IS";R       :rem 191
360 INPUT"ENTER RELEASE RATE (0-15)";R
                                   :rem 185
365 IFR<0ORR>15THEN360              :rem 74
370 GOTO 100                       :rem 100
400 PRINT"DECAY RATE IS";D          :rem 18
410 INPUT"ENTER DECAY RATE (0-15)";D
                                   :rem 12
415 IFD<0ORD>15THEN410              :rem 38
420 GOTO 100                        :rem 96
450 PRINT"PULSE WIDTH IS";100*P/4095
                                   :rem 86
460 INPUT"ENTER PULSE WIDTH (0-100)";P
                                   :rem 191
465 IFP<0ORP>100THEN460            :rem 115
470 P=P*4095/100                    :rem 52
480 GOTO 100                       :rem 102
500 PRINT"SUSTAIN TIME IS";T;"MILLISECOND
    S"                             :rem 236
510 PRINT"MINIMUM TIME FOR ATTACK/DECAY C
    YCLE IS:"                      :rem 44
515 PRINT A(A)+D(D);"MILLISECONDS" :rem 4
520 INPUT"ENTER TIME IN MILLISECONDS";T
                                   :rem 196
530 GOTO 100                        :rem 98
550 PRINT"DEAD TIME IS";B;"MILLISECONDS"
                                   :rem 198
560 INPUT"INPUT DEAD TIME IN MILLISECONDS
    ";B                            :rem 214
570 GOTO 100                       :rem 102
1000 W=1:A=8:D=6:R=9:SL=12:P=2000:T=302
                                   :rem 203
1010 RETURN                        :rem 162
1100 FORI=1TO8:READF(I):NEXT       :rem 234
1110 DATA 4291,4817,5407,5728,6430,7217,8
     101,8538                      :rem 155
1120 RETURN                        :rem 164
1200 FOR I=0TO15:READ A(I):D(I)=3*A(I):NE
     XT                            :rem 160
1210 DATA 2,8,16,24,38,56,68,80,100,250,5
     00,800,1000,3000,5000,7000    :rem 186
1220 RETURN                        :rem 165
1400 FOR I=1TO 8:F(I)=F(I)*2:NEXT:RETURN
                                   :rem 100
1450 FOR I=1TO8:F(I)=F(I)/2:NEXT:RETURN
                                   :rem 110
```

Program 2: Tune

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

```
5 DIM D(3,200),F(3,200),G(3,200)  :rem 254
10 S=54272                        :rem 245
20 FORI=0TO24:POKES+I,0:NEXT       :rem 13
30 FORI=1TO3                      :rem 215
40 J=1                             :rem 28
50 READ D(I,J),F(I,J):REM GET FREQ & DURA
   TION                            :rem 15
55 G(I,J)=INT(F(I,J)/256):F(I,J)=F(I,J)-2
   56*G(I,J)                       :rem 202
60 IF F(I,J)=0 AND D(I,J)=0 THEN 90
                                   :rem 228
70 J=J+1:GOTO 50                  :rem 108
90 PRINT "VOICE";I;" ";J;" NOTES" :rem 64
100 NEXT I                         :rem 25
110 POKES+24,15                    :rem 55
200 REM SET VOICE ONE             :rem 186
210 W1=16:REM TRIANGLE WAVEFORM   :rem 154
220 POKES+5,6*16+0:REM ATTACK=6,DECAY=0
                                   :rem 12
230 POKES+6,10*16+0:REM SUSTAIN=10,RELEAS
    E=0                            :rem 110
300 REM SET VOICE TWO              :rem 211
310 W2=32:REM SAWTOOTH WAVEFORM   :rem 189
320 POKES+12,0*16+9:REM ATTACK=0,DECAY=9
                                   :rem 65
330 POKES+13,00*16+0:REM SUSTAIN=00,RELEA
    SE=00                          :rem 203
400 REM SET VOICE THREE            :rem 82
410 W3=64:REM RECTANGULAR WAVE     :rem 79
420 POKES+17,3:REM DUTY CYCLE 20%:rem 101
430 POKES+19,3*16+10:REM ATTACK=3,DECAY=1
    0                              :rem 160
440 POKES+20,0*16+0:REM SUSTAIN=0:RELEASE
    =0                             :rem 104
500 J=0:K=0:L=0:T1=TI:T2=T1:T3=T1:rem 207
600 IF T1=<TI THEN GOSUB 1100      :rem 49
610 IF T2=<TI THEN GOSUB 1200      :rem 52
620 IF T3=<TI THEN GOSUB 1300      :rem 55
630 GOTO 600                       :rem 104
1000 ON I GOTO 1100,1200,1300      :rem 129
1100 J=J+1:T1=T1+D(1,J)            :rem 215
1115 IFD(1,J)=0 THEN POKES+4,W1:POKES+11,
     W2:POKES+18,W3:END            :rem 217
1117 POKES+4,W1                     :rem 95
1120 POKES,F(1,J):POKES+1,G(1,J)   :rem 51
1140 POKES+4,W1+1:RETURN           :rem 209
1200 K=K+1:T2=T2+D(2,K)            :rem 222
```

```
1210 POKE S+11,W2                    :rem 136
1220 POKE S+7,F(2,K):POKES+8,G(2,K)
                                     :rem 161
1240 POKES+11,W2+1:RETURN             :rem 1
1300 L=L+1:T3=T3+D(3,L)              :rem 229
1310 POKES+18,W3                     :rem 145
1320 POKES+14,F(3,L):POKES+15,G(3,L)
                                       :rem 2
1340 POKES+18,W3+1:RETURN            :rem 10
2000 REM NOTES FOR VOICE ONE         :rem 110
2010 DATA 30,4051                     :rem 54
2020 DATA 30,5407,30,4051,30,6069,30,4051
                                     :rem 215
2030 DATA 30,6430,30,6069,30,5407,30,4050
                                     :rem 218
2040 DATA 30,5407,30,4050,30,6069,30,4050
                                     :rem 215
2050 DATA30,6430,30,7217,30,8101,30,4050
                                     :rem 210
2060 DATA30,5407,30,4050,30,6069,30,4050
                                     :rem 217
2070 DATA30,6430,30,6069,30,5407,30,4050
                                     :rem 222
2080 DATA30,5407,30,4050,30,6069,30,4817
                                     :rem 230
2090 DATA60,5407,30,5407,30,4050      :rem 86
2120 DATA 30,5407,30,4051,30,6069,30,4051
                                     :rem 216
2130 DATA 30,6430,30,6069,30,5407,30,4050
                                     :rem 219
2140 DATA 30,5407,30,4050,30,6069,30,4050
                                     :rem 216
2150 DATA30,6430,30,7217,30,8101,30,4050
                                     :rem 211
2160 DATA 30,5407,30,4050,30,6069,30,4050
                                     :rem 218
2170 DATA30,6430,30,6069,30,5407,30,4050
                                     :rem 223
2180 DATA30,5407,30,4050,30,6069,30,4817
                                     :rem 231
2190 DATA120,5407                    :rem 117
2220 DATA 30,5407,30,4051,30,6069,30,4051
                                     :rem 217
2230 DATA 30,6430,30,6069,30,5407,30,4050
                                     :rem 220
2240 DATA 30,5407,30,4050,30,6069,30,4050
                                     :rem 217
2250 DATA30,6430,30,7217,30,8101,30,4050
                                     :rem 212
2260 DATA30,5407,30,4050,30,6069,30,4050
                                     :rem 219
2270 DATA30,6430,30,6069,30,5407,30,4050
                                     :rem 224
2280 DATA30,5407,30,4050,30,6069,30,4817
                                     :rem 232
2290 DATA120,5407                    :rem 118
2320 DATA 30,5407,30,4051,30,6069,30,4051
                                     :rem 218
2330 DATA 30,6430,30,6069,30,5407,30,4050
                                     :rem 221
2340 DATA 30,5407,30,4050,30,6069,30,4050
                                     :rem 218
2350 DATA30,6430,30,7217,30,8101,30,4050
                                     :rem 213
2360 DATA30,5407,30,4050,30,6069,30,4050
                                     :rem 220
2370 DATA30,6430,30,6069,30,5407,30,4050
                                     :rem 225
2380 DATA30,5407,30,4050,30,6069,30,4817
                                     :rem 233
2390 DATA120,5407                    :rem 119
2900 DATA 0,0                        :rem 113
3000 REM NOTES FOR VOICE TWO         :rem 135
3010 DATA990,0                       :rem 220
3020 DATA60,2703,60,2408             :rem 201
3030 DATA30,2145,30,2025,60,2145      :rem 73
3040 DATA60,2025,60,1804             :rem 199
3050 DATA30,1607,30,1517,60,1351      :rem 80
3060 DATA60,2703,60,2408             :rem 205
3070 DATA30,2145,30,2025,60,2145      :rem 77
3080 DATA60,2025,60,1804             :rem 203
3090 DATA30,1607,30,1517,60,1351      :rem 84
3120 DATA60,2703,60,2408             :rem 202
3130 DATA30,2145,30,2025,60,2145      :rem 74
3140 DATA60,2025,60,1804             :rem 200
3150 DATA30,1607,30,1517,60,1351      :rem 81
3160 DATA60,2703,60,2408             :rem 206
3170 DATA30,2145,30,2025,60,2145      :rem 78
3180 DATA60,2025,60,1804             :rem 204
3190 DATA30,1607,30,1517,60,1351      :rem 85
3220 DATA60,2703,60,2408             :rem 203
3230 DATA30,2145,30,2025,60,2145      :rem 75
3240 DATA60,2025,60,1804             :rem 201
3250 DATA30,1607,30,1517,60,1351      :rem 82
3260 DATA60,2703,60,2408             :rem 207
3270 DATA30,2145,30,2025,60,2145      :rem 79
3280 DATA60,2025,60,1804             :rem 205
3290 DATA30,1607,30,1517,60,1351      :rem 86
3900 DATA 0,0                        :rem 114
4000 REM NOTES FOR VOICE THREE         :rem 6
4010 DATA1950,0                       :rem 10
4020 DATA 60,2703,60,2408            :rem 202
4030 DATA 30,2703,15,2703,15,2703,60,2025
                                     :rem 215
4040 DATA 30,2703,30,2703,30,3034,30,3034
                                     :rem 206
4050 DATA 15,3215,15,3215,15,3215,15,3215
     ,60,3034                         :rem 99
4060 DATA 45,4050,15,3608,45,4050,15,3608
                                     :rem 234
4070 DATA 45,4050,15,3608,15,4050,15,3608
     ,15,3215,15,3034                :rem 249
4080 DATA 60,2703,60,2408            :rem 208
4090 DATA 30,2703,15,2703,15,2703,60,2025
                                     :rem 221
4100 DATA 30,2703,30,2703,30,3034,30,3034
                                     :rem 203
4110 DATA 15,3215,15,3215,15,3215,15,3215
     ,60,3034                         :rem 96
4120 DATA 45,4050,15,3608,45,4050,15,3608
                                     :rem 231
4130 DATA 45,4050,15,3608,15,4050,15,3608
     ,15,3215,15,3034                :rem 246
4140 DATA 60,2703,60,2408            :rem 205
4150 DATA 30,2703,15,2703,15,2703,60,2025
                                     :rem 218
4160 DATA 60,4050,60,4050            :rem 199
4170 DATA 30,4050,15,4050,15,4050,60,4050
                                     :rem 211
4900 DATA 800,0,0,0                  :rem 147
```

Program 3: Blast-off

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

```
10 S=54272                           :rem 245
20 FOR I=STOS+24:POKEI,0:NEXT         :rem 48
30 POKES+24,15                          :rem 8
40 FR=0500                           :rem 254
```

```
50 A=0:D=0:SS=15:R=0                          :rem 122
60 W=128:P=1024                               :rem 35
70 POKES+1,INT(FR/256)                        :rem 17
80 POKES,FR-256*INT(FR/256)                   :rem 66
90 POKES+3,INT(P/256)                         :rem 205
100 POKES+2,P-256*INT(P/256)                  :rem 56
110 POKES+5,16*A+D                            :rem 225
120 POKES+6,16*SS+R                           :rem 86
200 POKES+4,W+1:REM TURN SOUND ON             :rem 223
210 FORI=200TO1 STEP-1                        :rem 0
220 FR=FR+100:REM INCREASE FREQUENCY
                                              :rem 215
222 IF I< 45 THEN POKES+24,I/3:REM NEAR T
    HE END TURN DOWN THE VOLUME               :rem 98
225 F2=INT(FR/256):F1=FR-256*F2               :rem 224
230 POKES,F1:POKES+1,F2                       :rem 118
240 NEXT I                                    :rem 30
250 POKES+4,W:REM TURN SOUND OFF              :rem 198
```

Program 4: Laser

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

```
10 S=54272                                    :rem 245
20 FOR I=STOS+24:POKEI,0:NEXT                 :rem 48
30 POKES+24,143                               :rem 58
40 FR=50000                                   :rem 46
50 A=0:D=8:SS=15:R=08                         :rem 186
60 W=064:P=1024                               :rem 34
70 POKES+1,INT(FR/256)                        :rem 17
80 POKES,FR-256*INT(FR/256)                   :rem 66
90 POKES+3,INT(P/256)                         :rem 205
100 POKES+2,P-256*INT(P/256)                  :rem 56
110 POKES+5,16*A+D                            :rem 225
120 POKES+6,16*SS+R                           :rem 86
130 POKES+1'5,75                              :rem 63
155 POKES+4,W+3:REM USING W+3 TURNS ON
    {2 SPACES}GATE AND SYNC                   :rem 32
160 FORI=1TO25                                :rem 63
170 POKES+15,120-4*I:REM{2 SPACES}DECREAS
    E FREQ VOICE THREE                        :rem 180
180 NEXT I                                    :rem 33
185 POKES+4,W                                 :rem 2
```

Applesoft Lister

David Dobrin

"Applesoft Lister" will give you more readable program listings, along with printer-oriented output, translated control characters, and indention of nested FOR-NEXT loops.

Would you like your Applesoft programs to look like this:

```
10 REM BASIC LISTING WITH APPLESOFT LIST
20 HOME
22 PRINT "ANT SCRAM[G][G][G]"
30 FOR J=0 TO 35
31     VTAB 2
  :    HTAB J+1
40     PRINT " ;=;@"
50     NEXT
60 PRINT "[G][G][G]THAT IS ALL"
```

instead of this:

```
10   REM  BASIC LISTING WITH APPLESOFT LIST
20   HOME
22   PRINT "ANT SCRAM"
30   FOR J = 0 TO 35
31   VTAB 2: HTAB J + 1
40   PRINT " ;=;@"
50   NEXT
60   PRINT "THAT IS ALL"
```

Applesoft programs are usually very difficult to read. The standard LIST function built into Applesoft is unsophisticated, having only the minimum logic necessary to list programs. Here's a program for the Apple that will list Applesoft programs in a nicely formatted fashion. Five major features distinguish "Applesoft Lister" from the standard format:

- There is intelligent spacing between keywords, variables, and operands.
- Multiple statements with a single line number are listed one per line.
- FOR-NEXT constructs are nested.
- Output is oriented for a printer. This listing will not simply "wrap" when it runs out of space on a line.
- Control characters are shown with printable characters.

How Applesoft Lister Works

The program translates the Applesoft intermediate language (IL) into statement numbers and keywords. The keywords are taken from ROM at $D0D0. If this program is to be used with Applesoft in RAM, this value must be changed.

The high byte of the keyword table address is at location $812C. The low byte is at $8130.

When a colon (:) is encountered in the text, the lister starts a new line, indenting appropriately. No action is taken on colons inside double quotes or REM statements. FORs and NEXTs are observed to calculate a nest level.

If you would like to change the indentation of your FOR-NEXT constructs or multiple statements you can change the value at location 32771 with the POKE command. Putting a 0 there will turn indenting off, a 3 will indent three spaces per nest level, a 10 will indent ten spaces per nest level, and so on.

If you want to change the column width, change the value at 32772 with the POKE command. Putting a 39 there will give a screen width. You can also use 80, 132, or whatever your printer width is.

These POKEs can, of course, be made permanent by saving the program to disk or tape after changing.

Control characters are printed inside brackets; for example, CTRL-G appears as [G].

Loading The Program Into Your Apple

The lister program is written entirely in machine language. Program 1 is a BASIC program which READs the machine language from DATA statements and POKEs it into memory.

The program was assembled to load at location $8000. If your machine has less than 48K, the program will have to be relocated.

If you wish to enter the machine language, you can do so by using the Apple monitor (CALL –151). Enter the hex values as shown in Program 2. The *Apple Reference Manual*, Chapter 3, details the use of the resident monitor.

Once the program is entered into the Apple either by the loader or from the monitor, it should be saved to disk or tape before going any farther. This can be done by typing:

```
]BSAVE ALIST,A$8000,L$2F0
```

or

```
*8000.82F0W
```

Running Applesoft Lister

After the program has been stored, it can be utilized by loading the Applesoft program to be listed in the usual manner. The list program can then be loaded with:

```
]BLOAD ALIST
```

or

```
]CALL -155
*8000.82F0R
*(CTRL-C)
```

The listing program can then be run by typing:

```
]PR#x (where x is the slot for your printer interface,
    if you want the output to go to a printer)
]CALL 32768
```

Program 1: BASIC Loader For Applesoft Lister

```
100  FOR I = 32768 TO 33295
110  READ A:CK = CK + A: POKE I,A
120  NEXT
130  IF CK <  > 47880 THEN  PRINT "ERRO
     R IN DATA STATEMENTS": STOP
140  PRINT "LISTER ML LOADED"
150  END
200  DATA  76,5,128,3,80,169,0,133
210  DATA  10,169,1,133,0,169,8,133
220  DATA  1,169,141,32,157,129,32,96
230  DATA  129,133,2,32,96,129,133,3
240  DATA  5,2,208,1,96,32,96,129
250  DATA  133,4,32,96,129,133,5,169
260  DATA  0,133,6,133,7,133,8,162
270  DATA  16,24,248,165,6,101,6,133
280  DATA  6,165,7,101,7,133,7,165
290  DATA  8,101,8,133,8,216,6,4
300  DATA  38,5,144,2,230,6,202,208
310  DATA  224,162,5,160,0,165,8,41
320  DATA  15,208,12,192,0,208,8,224
330  DATA  1,240,4,169,160,208,4,160
340  DATA  1,9,176,32,157,129,152,72
350  DATA  160,4,6,6,38,7,38,8
360  DATA  136,208,247,104,168,202,208,
     213
370  DATA  169,160,32,157,129,32,107,12
     9
380  DATA  169,0,133,9,32,96,129,201
390  DATA  0,208,16,169,141,32,157,129
400  DATA  165,2,133,0,165,3,133,1
410  DATA  76,22,128,166,14,236,4,128
420  DATA  48,18,72,162,0,189,149,129
430  DATA  240,6,32,157,129,232,208,245

440  DATA  32,107,129,104,201,34,208,8
450  DATA  165,9,73,128,133,9,169,34
460  DATA  166,9,208,27,201,58,208,19
470  DATA  162,0,189,141,129,240,6,32
480  DATA  157,129,232,208,245,32,107,1
     29
490  DATA  76,148,128,201,128,16,26,41
500  DATA  127,201,32,16,14,72,169,91
510  DATA  32,157,129,104,9,64,32,157
520  DATA  129,169,93,32,157,129,76,148
530  DATA  128,72,201,129,208,2,230,10
540  DATA  201,130,208,2,198,10,201,178
550  DATA  208,2,230,9,170,188,37,129
560  DATA  132,11,36,11,16,5,169,160
570  DATA  32,157,129,169,208,133,13,16
     9
580  DATA  208,133,12,104,170,160,0,202

590  DATA  16,16,177,12,230,12,208,2
600  DATA  230,13,201,128,16,241,48,242

610  DATA  160,0,177,12,200,170,32,157
620  DATA  129,138,16,246,36,11,80,5
630  DATA  169,160,32,157,129,76,148,12
     8
640  DATA  160,0,177,0,230,0,208,2
650  DATA  230,1,96,162,13,134,14,166
660  DATA  10,16,2,162,0,224,6,48
670  DATA  2,162,6,202,48,14,172,3
680  DATA  128,136,48,247,169,160,32,15
     7
690  DATA  129,76,129,129,96,141,160,16
     0
700  DATA  160,160,160,186,0,141,160,16
     0

710  DATA  160,160,160,160,0,9,128,32
720  DATA  237,253,230,14,96,64,64,64
730  DATA  64,64,64,64,64,64,64,0
740  DATA  0,64,64,64,64,64,64,0
750  DATA  64,64,64,64,64,0,0,64
760  DATA  64,64,64,64,64,0,64,64
770  DATA  0,0,64,64,64,64,0,64
780  DATA  64,64,64,64,0,64,64,0
790  DATA  64,64,64,64,64,64,64,64
800  DATA  64,64,64,64,64,0,192,0
810  DATA  0,192,192,64,192,0,0,0
820  DATA  0,0,192,192,0,0,0,0
830  DATA  0,0,0,0,0,0,0,0
840  DATA  0,0,0,0,0,0,0,0
850  DATA  0,0,0,0,0,0,0,0
```

Program 2: Hex Dump Of Applesoft Lister Machine Language

```
8000- 4C 05 80 03 50 A9 00 85
8008- 0A A9 01 85 00 A9 08 85
8010- 01 A9 8D 20 9D B1 20 60
8018- B1 85 02 20 60 B1 85 03
8020- 05 02 D0 01 60 20 60 B1
8028- 85 04 20 60 B1 85 05 A9
8030- 00 85 06 85 07 85 08 A2
8038- 10 18 F8 A5 06 65 06 85
8040- 06 A5 07 65 07 85 07 A5
8048- 08 65 08 85 08 D8 06 04
8050- 26 05 90 02 E6 06 CA D0
8058- E0 A2 05 A0 00 A5 08 29
8060- 0F D0 0C C0 00 D0 08 E0
8068- 01 F0 04 A9 A0 D0 04 A0
8070- 01 09 B0 20 9D B1 98 48
8078- A0 04 06 06 26 07 26 08
8080- 88 D0 F7 68 A8 CA D0 D5
8088- A9 A0 20 9D B1 20 6B B1
8090- A9 00 85 09 20 60 B1 C9
8098- 00 D0 10 A9 8D 20 9D B1
80A0- A5 02 85 00 A5 03 85 01
80A8- 4C 16 80 A6 0E EC 04 80
80B0- 30 12 48 A2 00 BD 95 B1
80B8- F0 06 20 9D B1 E8 D0 F5
80C0- 20 6B B1 68 C9 22 D0 08
80C8- A5 09 49 80 85 09 A9 22
80D0- A6 09 D0 1B C9 3A D0 13
80D8- A2 00 BD 8D B1 F0 06 20
80E0- 9D B1 E8 D0 F5 20 6B B1
80E8- 4C 94 80 C9 80 10 1A 29
80F0- 7F C9 20 10 0E 48 A9 5B
80F8- 20 9D B1 68 09 40 20 9D
8100- B1 A9 5D 20 9D B1 4C 94
8108- 80 48 C9 B1 D0 02 E6 0A
8110- C9 B2 D0 02 C6 0A C9 B2
8118- D0 02 E6 09 AA BC 25 B1
8120- B4 0B 24 0B 10 05 A9 A0
8128- 20 9D B1 A9 D0 85 0D A9
8130- D0 85 0C 68 AA A0 00 CA
8138- 10 10 B1 0C E6 0C D0 02
8140- E6 0D C9 80 10 F1 30 F2
8148- A0 00 B1 0C C8 AA 20 9D
8150- B1 8A 10 F6 24 0B 50 05
8158- A9 A0 20 9D B1 4C 94 80
8160- A0 00 B1 00 E6 00 D0 02
8168- E6 01 60 A2 0D 86 0E A6
8170- 0A 10 02 A2 00 E0 06 30
8178- 02 A2 06 CA 30 0E AC 03
8180- 80 88 30 F7 A9 A0 20 9D
8188- B1 4C B1 B1 60 8D A0 A0
8190- A0 A0 A0 BA 00 8D A0 A0
8198- A0 A0 A0 A0 00 09 80 20
81A0- ED FD E6 0E 60 40 40 40
81A8- 40 40 40 40 40 40 40 00
81B0- 00 40 40 40 40 40 40 00
81B8- 40 40 40 40 40 00 00 40
81C0- 40 40 40 40 40 00 40 40
81C8- 00 00 40 40 40 40 00 40
81D0- 40 40 40 40 00 40 40 00
81D8- 40 40 40 40 40 40 40 40
81E0- 40 40 40 40 40 00 C0 00
81E8- 00 C0 C0 40 C0 00 00 00
81F0- 00 00 C0 C0 00 00 00 00
81F8- 00 00 00 00 00 00 00 00
8200- 00 00 00 00 00 00 00 00
8208- 00 00 00 00 00 00 00 00
```

Program Conversion With Sinclair BASIC And TI BASIC

Julie Knott and Dave Prochnow

Program conversion between BASIC dialects is often easier than imagined. This tutorial demonstrates the compatibility of TI BASIC and Sinclair BASIC and includes helpful tables and sample conversion programs.

Program conversion can be an easy and convenient operation. Virtually every home computer uses BASIC, which, because it's easy to learn and to manipulate, is ideal for ready-made language conversion. However, no two BASICs are created equal. For many years the industry's standard was Microsoft BASIC, then different dialects began to emerge. Manufacturers would use the Microsoft format and introduce nuances and subtleties in the structuring, labeling each of these alterations an "improvement" of BASIC. But many were only changes in the protocol—the manner in which a command is expressed. And it becomes relatively easy to convert BASIC dialects if the major differences are in protocol or syntax.

Two versions of BASIC which lend themselves to such a program conversion are Sinclair BASIC and TI BASIC. Sinclair BASIC, used in the Timex/Sinclair-1000, is unique in that all keywords are single-stroke entries. For example, the P key stands for the PRINT command. (The use of a touch-membrane keyboard dictates this procedural necessity.)

Texas Instruments TI-99/4 and 4A use TI BASIC, which is more conventional in that each individual letter has to be typed—PRINT would require five keystrokes.

There are only slight variations between Sinclair BASIC and TI BASIC, but their similarities allow for easy program conversion. By studying which statements and commands are equivalent for both BASICs, and what substitutions are necessary, program conversion can be relatively simple. Also, you can virtually double your software by translating programs published for the other machines.

For the sake of brevity, the following glossary does not contain all of the keywords in Sinclair BASIC and TI BASIC—only those words which are confusing, complicated, or not directly translatable have been listed. For a more complete listing, consult the appropriate user's manual.

Sinclair BASIC

AND — a logical operator, often used in IF-THEN statements

ACS — function that gives the arc cosine of an angle in radians

ASN — function that gives the arc sine of an angle in radians

AT — used in a PRINT statement to give a location at which to PRINT

BREAK — stops program execution, key activated and may not be included as a command in a program

CLEAR — deletes all variables from memory

CLS — clears the screen

CODE — a string function used to obtain the numeric value of a given character

CONT — resumes execution of a program following a report code

COPY — copies the contents of the screen to printer

DELETE — erases keywords and characters while programming

FAST — fast mode, a time-saving mode for increased RUN speed

FUNCTION — function mode

GRAPHICS — graphics mode

INKEY$ —used in IF-THEN statements as a conditional statement, executes exclusive of ENTER

LLIST — lists the contents of a program listing to a printer

LOAD — loads a prerecorded program from cassette tape to the computer's memory

LPRINT — PRINTs to printer

NOT — inverts the truth value of an expression

OR — a logical operator, used in conditional statements

PAUSE — creates a time delay while the program is RUNning.

PEEK — gives the value of the byte at a specific address in memory

PI — gives the value of PI

PLOT — draws a pixel at a given location

POKE — puts a numeric value into memory at a specific address, erasing the previous one

SCROLL — scrolls the screen up one line, eliminating the top line

SLOW — slow mode, the standard operating mode

UNPLOT — erases a pixel at a given location

USR — calls a machine language routine at a specific memory address

TI BASIC

APPEND — an open mode, allows data to be added at the end of the existing file

ASC — ASCII value or character code

BASE — option base

BREAK — sets breakpoints in a program, used for error checking

BYE — erases memory, returns to title screen

CALL — special subprogram to obtain color and sound

CLOSE — closes the association between a file and a program

CONTINUE (CON) — continues a program after a breakpoint

DATA — stores data

DEF — defines user-established functions in a program

DELETE — removes a program or data file from a filing system

DISPLAY — prints on screen only

ELSE — conditional part of IF-THEN/ELSE statement

END — terminates program, similar to STOP

EOF — End-Of-File, determines if the end of a file has been reached on an accessory device

FIXED — files with a specified length, used with RELATIVE or SEQUENTIAL

INTERNAL — file type recorded in machine language

NUMBER (NUM) — automatic line number generator

OLD — loads a previously SAVEd program

ON — a conditional numeric expression, used with ON-GOTO or ON-GOSUB

OPEN — prepares to use data files stored in accessory device

OPTION — option base, sets lower limit of array subscripts to 1 instead of 0

OUTPUT — transfers data out of a program

PERMANENT — file life

POS — position

READ — reads data in DATA statements

REC — points to a specific record in a RELATIVE file

RELATIVE — defines a file with FIXED

RESEQUENCE (RES) — reassigns line numbers

RESTORE — identifies which DATA to use with the next READ

SEG$ — string segment, substring

SEQUENTIAL — defines a file, used with FIXED or VARIABLE

SUB — part of GO SUB

TRACE — outlines the order that statements will be performed when the program is RUN

UNBREAK — removes breakpoints

UNTRACE — cancels TRACE

UPDATE — an open mode, for reading and writing into files

VARIABLE — defines a varying length file, used with SEQUENTIAL

Special Subprograms Used With Graphics And Sound In TI BASIC

Each subprogram is preceded by CALL (for example, CALL CLEAR)

CLEAR — erases the entire screen

COLOR — specifies screen character colors

SCREEN — changes screen color

CHAR — defines user-special graphic characters

HCHAR — places a character and repeats it horizontally

VCHAR — similar to HCHAR except repetition is vertical

SOUND — produces tones and noises of different duration, frequency, and volume

GCHAR — reads a character anywhere on the screen

KEY — transfers character directly from keyboard to program without ENTER

JOYST — inputs data with remote controllers

Easy Conversions

Many of the commands and statements of these two BASICs are directly translatable. Table 1 shows the direct BASIC equivalents for Sinclair BASIC and TI BASIC. The only major differences between these two dialects are in their nomenclature.

Several dialects of BASIC have an ON-GOTO statement expressed as:

```
ON x GOTO w,y,z
```

where x is the value of a numerical expression and w, y, and z are line numbers. This statement is available in TI BASIC, but not in Sinclair BASIC. Through the use of conditional expressions, the

Sinclair BASIC substitution is:

```
GOTO (w AND x=1)+(y AND x=2)+(z AND x=3)
```

The operators AND and OR would make this possible.

The translation of many program lines requires only the replacement or substitution of a word unique to that particular BASIC. Several of the more common functions and statements are evaluated in this manner in Table 2. The following Sinclair BASIC line will await the pressing of the Y key, exclusive of ENTER:

```
100 IF INKEY$ <> "Y" THEN GOTO 100
```

To perform the same statement in TI BASIC, replace INKEY$ with the KEY subprogram, as follows:

```
100 CALL KEY(0,K,Z)
110 IF K<>89 THEN 100
```

The main difference is in the structuring. The KEY subprogram (subprograms are obtained with CALL) uses three variables to establish where the key is originating, its ASCII code, and its status. In this example the ASCII code of 89 represents the Y character.

TI BASIC has the ability to store expressions and assign values to these variables with the statements DATA, READ, and RESTORE (see the glossary). Vast arrays can be developed and initialized with this method. Sinclair BASIC is not directly convertible with DATA, READ, and RESTORE. A large battery of LET statements *could* crudely handle the data. Alternatively, a properly DIMensioned INPUT statement allows the creation of such an array. Upon completion, the INPUT statements are removed and a GOTO command is used for program starting (RUN erases the variable array).

String Handling

Strings can be equally bothersome. Slicing will supply usable substrings in Sinclair BASIC. A string expression's parameters govern the start and finish of the slice. No special statement is necessary:

```
A$(x TO z)
```

with x representing the starting number and z the finish. For example:

```
"COMPUTE" (4 TO 7) = "PUTE"
```

The statement SEG$ (A$,x,y) in TI BASIC has the same result, but, again, with different nomenclature. X is the number of the start for the substring and Y is the length of the substring. For example:

```
A$="COMPUTE"
SEG$ (A$,4,4)="PUTE"
```

While string slicing is easily translated, the TI BASIC user-defined function is not. DEF allows the definition of functions within a program.

```
DEF X$="Y"
```

The string function's name is X and the string expression is Y. VAL and string variables can be user-defined in Sinclair BASIC.

```
LET X$="Y"
VAL X$= Y
```

This is a very limited and a "sometimes-maybe" proposition. DEF has the ability to also handle numeric functions. This ability, as well as using parameters in argument evaluation, is beyond VAL's means.

When attempting a program conversion you may run across a few Sinclair BASIC terms that are completely unfamiliar to you. The terms USR, PEEK, and POKE are not procedures for the examination of some strange alien creature. They are primarily associated with direct access to memory. To call a machine language routine that begins with a specific address, USR is used. This will start a machine language program running. POKE is used by the T/S-1000 to store a numeric value at a specific address in the computer's memory. For example:

```
POKE 17529, 38
```

POKEs the value 38 into address 17529. Conversely, the PEEK command is used to read certain addresses to see what is stored there. The PEEK command is followed by the address to be PEEKed.

```
PRINT PEEK 17529
```

would PRINT the number 38. When you are translating a program from Sinclair BASIC which contains USR, PEEK, and POKE statements, you must find out what they accomplish and then interpret that into TI BASIC.

PRINTing on the screen is accomplished by a blending of line and row markers. Memory conservation techniques notwithstanding, PRINT can be used to move the PRINT line. For example:

```
PRINT
PRINT
PRINT "COMPUTE"
```

Sinclair BASIC also allows the movement of PRINT with AT and TAB.

```
PRINT AT x,y
```

and

```
PRINT TAB y
```

TAB moves the PRINT position a prescribed number of spaces to the right. Even though TAB is present in TI BASIC, the vocabulary is different. Line changes are accomplished with colons (:). Duplicating the above examples,

```
PRINT::...(x) TAB (y)
```

and

```
PRINT TAB (y)
```

X is the number of colons necessary to equal the

value of the line number (x) in the Sinclair BASIC example.

The Timex/Sinclair lacks color and sound features, but these features are of importance on the TI-99/4. TI BASIC's color and sound statements are subprograms that begin with CALL. Clever usage of Sinclair BASIC's character set can duplicate some of these color combinations. As a rule, however, TI BASIC CALL subprograms should be removed and not directly substituted in a program conversion to Sinclair BASIC. This allows concentration on the program's more important graphics. Consultation with Texas Instruments' *User's Reference Guide* will provide the proper protocol for development and inclusion of color and sound subprograms in a Sinclair BASIC converted to TI BASIC program.

To illustrate the principles of program conversion, examine these sample programs. While each program is unique in its results, the approach is similar and convertible. The purpose of this program is to display the entire character set along with the character codes.

T/S-1000 Version

```
10 FORA=0 TO 255
20 LET A$ = CHR$ A
30 PRINT AT 10,13; A
40 PRINT AT 7,10; A$;"{6 SPACES}"
50 PRINT AT 7,17; A$;"{6 SPACES}"
60 PRINT AT 13,10; A$;"{6 SPACES}"
70 PRINT AT 13,17; A$;"{6 SPACES}"
80 NEXT A
```

TI-99/4 Version

```
100 FOR A=32 TO 127
110 B$=CHR$(A)
120 CALL CLEAR
130 CALL SCREEN(2)
140 PRINT TAB(13);B$;TAB(18);B$
150 PRINT
160 PRINT TAB(14);A
170 PRINT
180 PRINT TAB(13);B$;TAB(18);B$
190 PRINT ::::::::
200 FOR S=3 TO 16
210 CALL SCREEN(S)
220 CALL SOUND(400,110+80*(S-3),1)
230 NEXT S
240 NEXT A
```

In line 10 of the Timex/Sinclair example, a loop establishes the number of character codes to be examined (the entire character set is 0 to 255). Note that the characters with codes 67–127 cannot be printed and will show on the screen as question marks. Lines 20 and 30 PRINT the code or numeric value for each character. The arrangement of the printed characters is defined in lines 30–40. In this way, you can easily interpret the delay, and read the code value and the character almost simultaneously. This program will RUN until BREAK is pressed.

Table 1:
Reference Chart Of BASIC Equivalencies

Sinclair BASIC	= TI BASIC
ABS	ABS
ATN	ATN
CHR$	CHR$
CODE	ASC
COS	COS
DIM	DIM
EXP	EXP
FOR	FOR
GOSUB	GOSUB or GO SUB
GOTO	GOTO or GO TO
IF	IF
INPUT	INPUT
INT	INT
LEN	LEN
LET	LET
LN	LOG
LOAD	OLD
NEW	NEW
NEXT	NEXT
PRINT	PRINT
RAND	RANDOMIZE
REM	REM
RETURN	RETURN
RND	RND
RUN	RUN
SAVE	SAVE
SGN	SGN
SIN	SIN
SQR	SQR
STEP	STEP
STOP	STOP
STR$	STR$
TAB	TAB
TAN	TAN
THEN	THEN
TO	TO
VAL	VAL
CLS	CALL CLEAR

Table 2:
Substitution Chart For BASIC Nonequivalents

Sinclair BASIC	= TI BASIC
NEW	BYE
PRINT	DISPLAY
GOTO (W AND X=1) +(Y AND X=2)+(Z AND X=3)	ON X GOTO W,Y,Z
IF X THEN GOTO Y	IF X THEN (Y)
LET X=Y+Z	LET X=Y+Z or X=Y+Z
PAUSE or FOR X=Z TO Y NEXT X	FOR X=Z TO Y NEXT X
INKEY$	CALL KEY
A$(X TO Z)	SEG$(A$,X,Y)
PI	4*ATN (1)
LET X$="Y" VAL X$=Y	DEF X=Y
STOP	END or STOP
PRINT AT X,Y	PRINT...(X) TAB(Y)
PRINT TAB Y	PRINT TAB(Y)
ASN 1	π /2 or 4*ATN(1)/2 or 2*ATN(1)
IF X=Y THEN GOTO A GOTO Z	IF X=Y THEN A ELSE Z

Commodore 64 ROM Generations

Jim Butterfield, Associate Editor

Commodore products are often subject to changes in logic. Not marketing logic or pricing logic (although they change too), but the internal logic that drives the machines: the programs in ROM. This has been true of PET/CBM and various disk systems. This article traces differences in two major ROM releases of the Commodore 64 computer, plus a third released with the SX-64 portable computer.

Two Environments

The first 64s used ROM set 1. Before releasing a European version of the 64, Commodore developed ROM set 2. ROM 2 is unique in that it's the same for North America and Europe, yet recognizes and copes with differences between the two environments. More on that later.

Programs developed on ROM set 1 sometimes didn't seem to work on ROM set 2. This was particularly true when the screen was set up using a POKE statement. For example, a user clearing the screen and then typing the command POKE 1500,1 will print a letter A around the middle of the screen, but with ROM 2 this letter is "invisible." Many games and educational programs using the screen this way couldn't make the transition from ROM 1 to ROM 2; attractive graphics would become invisible and the effect would be lost.

I have met a third ROM recently; it's used in the SX-64 portable computer. There are small differences: For example, disk activities are given preference over tape, and screen POKEs are once again legal.

In all cases, the BASIC language in ROM is not changed (addresses $A000 to $BFFF). All changes are in the Kernal ROM, which resides at addresses $E000 to $FFFF.

All three ROM sets are very similar; the differences are largely cosmetic. Sometimes, of course, cosmetic differences are enough to prevent a particular program from working in a satisfactory manner; but there's a strong bond between all models I have examined.

The Tape Pause

When you give a tape LOAD command, the computer blanks the screen and searches for a program "header" on the tape. When it finds a program, it reports the name with a message, FOUND XXXXX, unblanks the screen, and waits. When you touch a key (preferably the Commodore Logo key), the screen blanks once again and the program starts to load.

ROM 1 waits forever. If you don't press a key, it keeps waiting. ROM 2, however, waits only a few seconds and then proceeds with the program load activity. ROM 3 for the SX-64 doesn't have a cassette tape connection, so it doesn't do either.

Why does the screen need to blank? Here's the reason: The screen interferes very slightly with the processor. Roughly once every 1/2000 second, the processor chip is stopped briefly to allow the video chip to get extra information from memory. This is no hardship except when we need to read or write tape.

When cassette tape is active, the processor needs to time events precisely. It can't afford to miss even the brief time lapse that the video chip might cause. So it turns the screen off in order to get the most efficient timing "edge."

Technical note: The "Find Tape Header" subroutine at $F761 is changed in ROM 2 so that it calls a new subroutine at $E4E0 to allow time-out. The same coding is used in the SX-64 ROM, but it's not useful since this machine can't use tape.

Screen Clear

When ROM 1 clears the screen, it sets the foreground color of all screen locations to white. As a result, it's easy to POKE screen memory and have white characters appear.

ROM 2 changed all that. When the screen clears, the foreground color of all characters is set to the background color. If you POKE to an unused location, you'll end up printing blue on blue, which makes it invisible. The character is indeed there: You can see it if you place the cursor over that position. But it's not much use to the viewer.

Commodore may have done this to reduce screen "sparkle"—colored or white flashes that appear randomly on the screen. Whatever the reasoning, it caused writers of software some anguish if their existing programs POKEd the screen a good deal. Many Commodore demonstration programs lost their appeal on the new machines. All programs would still run, but the screen wouldn't look right.

With the new SX-64 ROM, we're back to allowing screen POKEs. It may be too late for software writers, but when the SX-64 clears the screen, it sets the foreground color of all screen locations to the cursor color. That's better than ROM 1, which sets white only—you have a chance to choose the POKE color.

Technical note: The Clear-a-Line subroutine at $E9FF was changed slightly to call a new subroutine at $E4DA; this sets character color to background color on ROM 2. On SX-64 ROM, character color is set to the value from $0286, the current "cursor" color.

Different Crystal Speeds

ROM 1 was designed for North American use. ROM 2 was designed for worldwide use, and considerable thought was put into creating a universal design. When power is applied to the computer, ROM 2 does some interesting detective work.

Very early in the game, ROM 2 set the raster interrupt to fire at scan line 622. Here's the trick: There is no line 622 on North American sets; if the interrupt signal fires, we must be elsewhere.

Depending on the continent, the ROM sets up timing for the clock and RS-232 transmission. What's happening here is that the two different types of machine are driven at different "crystal" speeds, and the program must compensate for this to allow consistent overall speed.

The programmer on a ROM 2 system must keep in mind that the raster interrupt register in the video chip has already been used by the system; it cannot be assumed to be zero.

Technical note: The table at $ECB9 which sets up the video chip has been changed to include the raster interrupt. The Power-Up Reset program itself has been changed at $FCFB by the insertion of a call to a new subroutine at $FF5B. If line 622 (Europe) is detected, address $02A6 is set to 1 to signal "European System." This new location, $02A6, is used to set up the timer which creates "jiffies"—1/60-second interrupts. It will also be checked if the RS-232 channel is opened, and timing information extracted from the appropriate table.

Small Stuff

ROM 1 had troubles if you tried to PRINT# to a device that wasn't there; ROM 2 has its act together a little better.

SX-64 ROM identifies itself with a new message: SX-64 BASIC V2.0, in case you didn't notice that it was an SX-64 you had.

If you hold down SHIFT and press RUN/STOP on the SX-64, you'll get a load/run from *disk*; the screen reads LOAD":*",8 ... RUN. This data is stored in an area of memory that usually contains the message PRESS PLAY, but you won't be using the cassette this time so you won't miss that message. Any attempt to use a cassette on the SX-64, by the way, will result in an ILLEGAL DEVICE NUMBER message.

The differences are not great. Most users will spot only the tape pause and the screen POKE as operational differences.

Serious programmers will appreciate the fact that changes have been made as "patches," which means that previous entry points have not moved; they are still in the same places that they used to be. A call to a machine language subroutine at a given location will still be good.

There are still things that many users would like to see improved in Commodore 64 BASIC and Kernal. In particular: The INPUT statement is uncomfortable at times, and certain types of screen editing work awkwardly. You may have a wish list of your own. It seems quite likely that we'll see another ROM system one of these days.

Coming Soon

Commodore is said to be working on new ROM systems for the 64 and its peripherals. Compatibility is expected to be retained with previous ROMs, but certain operational annoyances will be eliminated.

Watch for a new Kernal ROM—we expect it to be coded 901227-03. It will fix up a couple of problems associated with screen usage.

The Commodore 64, like the VIC-20, behaves oddly if an INPUT statement is written with a lengthy prompt; if the prompting message is long enough, the user input will need to be typed onto the next line of the screen. In such a case, the computer receives a peculiar input: As well as reading what the user has typed in, it reads its own prompt message.

A more serious problem arises if a user types in a line longer than 80 characters, and then backs up using the Delete key. The too-long line goes beyond two rows on the screen, of course; but when the user backs up, the computer *might* stop working.

The above problems are expected to be fixed when the new version 3 chip is released. In addition, some of the above-noted changes for the SX-64 will also be implemented—for example, screen POKEs.

Commodore is also said to be working on new logic for printers and disk units. Watch for them, too.

Commodore 1541 Generations

Tracking the generations of Commodore's 1541 disk drive is not unlike reading a mystery novel. Unfortunately for 1541 owners, Commodore so far has not written the last page in which the mystery is revealed, so we can only examine the clues and speculate.

Clue No. 1: The original 1541 had a "long" circuit board which extended the length of the drive. This board probably was the same as was in the 1540 drive, predecessor to the 1541.

Clue No. 2: Both the 1540 and the original version of the 1541 had white cases.

Clue No. 3: Later versions of the 1541 have brown cases, and a "short" board which extends about half the length of the drive. Our sources tell us that the short board is a redesigned long board and that when the circuit board was redesigned, timing problems showed up in the drive.

Clue No. 4: ROM chips bearing four different part numbers have been seen in 1541 drives. During a teleconference on the Commodore Information Network on March 29, 1984, a Commodore Research & Development representative gave the part number of the latest ROM as 901229-05. (The suffix 05 indicates the ROM version.) ROM chips with suffixes 01, 02, and 03 also have been seen in 1541 drives.

Clue No. 5: During the teleconference, the Commodore representative said that one of the changes incorporated into the 05 ROM version had to do with the serial bus. (Peripherals such as the 1541 and the 1525 printer connect to the Commodore 64 through the serial bus.)

Clue No. 6: Owners of the 1541 have reported problems when trying to use two 1541s; occasionally, when a program accesses one of the drives, the system locks up. Problems also have been reported involving lockup on systems with one 1541 drive and the Commodore 1526 and MPS-801 printers.

Clue No. 7: 1541 users report an intermittent problem when saving files to disk using the replace option (SAVE "@0:filename",8). Instead of replacing the intended file, the drive's operating system writes over another file on the disk, and changes the directory pointers so that the intended file is no longer accessible. A similar problem has been reported in the Commodore 4040 drives. At the teleconference, the Commodore representative said he'd never experienced this problem. However, he also said that the 4040 and 1541 used the same basic operating system.

Clue No. 8: A technical representative with Integrated Computer Repairs (ICR), of Santa Mesa, California, told us that his company repairs and updates the 1540 and 1541 drives. ICR claims that merely replacing the ROM chip with an 05 version is not enough; they also make hardware changes, modifying the short circuit board.

Clue No. 9: Overheating problems have been reported with the 1541. After the drive has been on for several hours, some users report input-output errors and other problems.

Clue No. 10: In the past, Commodore representatives have said that the 1541 ROM changes were "mainly cosmetic."

Clue No. 11: ICR claims that the drives they have updated no longer have lockup problems. It is not clear whether their update solves the save-with-replace problem.

Clue No. 12: COMPUTE! made several telephone calls to Commodore Business Machines, Inc., asking Commodore to respond to the above items. Ms. Susan West, of the Public Relations Department, promised to find a technical representative who could answer our questions. We never heard from a technical representative, and Ms. West failed to return our subsequent calls, the last placed as this article was going to press.

So, it seems that Commodore has issued at least four different versions of the 1541 disk drive, for reasons which are known only to them. And problems may exist (or have existed) not only in the ROM chips, but also in the board circuitry. Finally, it appears that Commodore is unwilling to help us solve the mystery. C

Atari MacroDOS: Part 2

Jerry Allen

Last month we introduced "Atari MacroDOS" and presented a BASIC program which loads the MacroDOS machine language. This month we'll look at some technical details of MacroDOS and present a disassembly of the program.

Assembler users can alter the MacroDOS commands table (CMDTAB) if they so desire. Just remember to change lines which check for command input to reflect the new command letter. Also, revise TAB1 if necessary.

You can append another AUTORUN.SYS program to the end of MacroDOS, such as a menu loader for BASIC.

Assuming you have MacroDOS up and running as AUTORUN.SYS, enter DOS, then load the other AUTORUN.SYS from another disk. SAVE with APPEND ("D:AUTORUN.SYS"). Enter the beginning and ending addresses at the @ prompts. If necessary, return to the cartridge and POKE or otherwise change the INIT and RUN addresses. Return to DOS and SAVE with append again to pick up the addresses just altered.

Finding Load Addresses

If you can't figure out the load addresses, use this program:

```
10 OPEN#2,4,0,"D:YOURPROG.OBJ
20 FOR I=1 TO 6
30 GET#2,A
40 PRINT A
50 NEXT A
60 CLOSE#2
```

The first two bytes should be a header of 255 ($FF). The next four bytes will be the beginning and ending addresses of the load (two-byte numbers in low byte, high byte format).

If the file loads to multiple address areas (including RUN and INIT) after the first block of memory is loaded, OS checks for a new header of 255,255. If it is there, the header is ignored, and the next four bytes will be the new from-to load addresses.

Loading With Page 6

Loading RUN with page 6 (1536) would look like:

```
$E0(224),$02(2),$E0(224),$02(2),$00(0),$06(6)
```

(without the header). With a little math and modification of the program, you could find all the load addresses of any compound load file.

If you don't want to type the programs in, send $3, and a disk or tape with an SASE mailer.

Jerry Allen
1906 Carnegie #E
Redondo Beach, CA 90278

MacroDOS, Machine Language Source Code

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

```
0200 ;EQUATES
0210 ICBC  = $342
0220 ICBAL = $344
0230 ICBAH = $345
0240 ICBLL = $348
0250 ICBLH = $349
0260 ICBAX = $34A+16
0270 MEMLO = $2E7
0280 LBUF = $580
0290 INBUF = $F3
0300 CIX = $F2
0310 FR0 = $D4
0320 FR1 = $E0
0330 GETR = $05
0340 GETC = $07
0350 PUTC = $0B
0360 PUTR = $09
0370 OPEN = $03
0380 CLOSE = $0C
0390 AXIO = $0C
```

```
0400 AXOUT = $08
0410 AXAP = $09
0420 AXDR = $06
0430 FRØZ = $DA44
0440 IFP = $D9AA
0450 FPI = $D9D2
0460 FPASC = $D8E6
0470 FMOVE = $DDB6
0480 ASCFP = $D8ØØ
0490 LO = $ØØFF
0500 HI = $Ø1ØØ
0510 CIO = $E456
0520 WARMST = $E474
0530 DOSVEC = $ØA
0540 DOSINI = $ØC
0550 OLDDOS = $179F
0560 JMPINI = $17Ø5
0570 JMPRUN = $17Ø8
0580 ;
0590 ;
0600   *= $1CFC ;change this addr
     for
0610 ;larger versions of DOS2.ØS
0620 ;
0630 ;
0640 ST JSR CLSE ;to be sure
0650   STY CIX ;set f.p. pointer
0660   DEY
0670   STY $2E3;clear INIT/RUN
0680   STY $2E1
0690   STY $FF ;reset load flag
0700   LDX #LBUF&LO ;init flt. pt
     .
0710   LDY #LBUF/HI
0720   STX INBUF
0730   STY INBUF+1
0740 DISCMD LDY #TAB1-CMDTAB-1
0750 L1 TYA ;DISPLAY COMMANDS
0760   PHA
0770   LDA CMDTAB,Y
0780   JSR PRINT
0790   PLA
0800   TAY
0810   DEY
0820   BPL L1
0830   LDA #AXOUT ;init aux
0840   STA ICBAX
0850   JSR GTREC ;get command
0860   LDY #4 ;GET COMMAND
0870   LDA LBUF
0880 L2 CMP TAB1,Y
0890   BEQ SPECMD
0900   DEY
0910   BPL L2
0920 DIR CMP #'D ;DIRECTORY
0930   BNE WDS
0940   LDA #Ø6
0950   STA ICBAX
0960   JSR ASKDN ;drive #?
0970   JSR OPN   ;open
0980 L3 LDX #$1Ø
0990   JSR GTREC ;get formatted l
     ine
1000   JSR PTRECØ ;print it
1010   BPL L3
1020 SPECMD LDA TAB2,Y ;SPECIAL C
     MDS
1030   PHA ;save cmd
1040   CMP #$FE ;check if format
1050   BEQ FMT
1060   JSR PFN
1070 EX PLA ;retrieve command
1080   JSR EXCMD ;do it
1090   BPL ST
1100 FMT LDA #'? ;FORMAT
1110   JSR PRINT ;sure?
1120   JSR GTREC
1130   LDA LBUF
1140   CMP #'Y ;is there a yes?
1150 B1 BNE ST ;if not start over
1160   JSR ASKDN ;get drive#
1170   BMI EX ;execute
1180 ASKDN LDA #'D ;GET DRIVE#
1190   JSR PRINT
1200   LDA #'#
1210   JSR PRINT
1220   JSR GTREC
1230   LDA LBUF
1240   STA ADRDIR+1 ;change D#
1250   STA ADRDOS+1 ;just in case
     WDS
1260   LDY #6
1270 L4 LDA ADRDIR,Y ;move filena
     me
1280   STA (INBUF),Y
1290   DEY
1300   BPL L4
1310   RTS
1320 PFN LDA #'F ;PROMPT FILENAME
1330   JSR PRINT
1340   LDA #'N
1350   JSR PRINT
1360   LDA #'?
1370   JSR PRINT
1380   JMP GTREC
1390 WDS CMP #'W ;WRITE DOS.SYS
1400   BNE LOD
1410   JSR ASKDN
1420   INY
1430 L5 LDA ADRDOS,Y
1440   STA (INBUF),Y
1450   DEY
1460   BNE L5
1470   JSR OPN
1480   BPL B1
1490 LOD CMP #'L
1500   BNE SAV
1510   STA $FF
1520   JSR PFN
1530   LSR ICBAX ;8>4
1540   JSR OPN
1550 L6 JSR GETCR2 ;get hdr in pa
     irs
1560   LDA #$FF ;check headr and-
1570   CMP FRØ ;disregard $FF'S
1580   BNE SK2
1590   CMP FRØ+1
1600   BEQ L6
1610 SK2 JSR FMOVE ;FPØ to FP1
1620   JSR GETCR2
1630   JSR SUBTR ;subtr HI-LO &ex
     ecute
1640   JSR CHKIN
1650   BEQ L6
1660 CHKIN LDA $2E3
1670   BEQ SK12
1680   JSR JMPINI
1690   LDA #Ø
1700   STA $2E3
1710 SK12 RTS
1720 SAV CMP #'S ;SAVE FUNCTION
1730   BNE RUN
1740   JSR PFN
1750   LDY #$FF ;check if append
     (/)
1760 L7 INY
1770   LDA (INBUF),Y
1780   CMP #'/
```

```
1790   BNE SK4
1800   INC ICBAX ;9=append
1810   LDA #$9B
1820   STA (INBUF),Y
1830 SK4 CMP #$9B
1840   BNE L7
1850   JSR OPN ;open for write
1860   LDA #$FF ;start headr
1870   STA FR0
1880   STA FR0+1
1890   JSR PUTCR2 ;write it
1900   JSR INPCON ;get from#
1910   JSR PUTCR2 ;write it
1920   JSR FMOVE  ;store it
1930   JSR INPCON ;get to#
1940   JSR PUTCR2 ;write it
1950   JSR SUBTR  ;find len and s
     ave
1960   BPL B2 ;the rest
1970 RUN CMP #'@ ;RUN
1980   BNE CART
1990   JSR INPCON ;get #
2000   JMP (FR0)  ;jump indirectl
     y
2010 CART CMP #'C ;CARTRIDGE
2020   BNE ADOS
2030   JMP WARMST
2040 ADOS CMP #'! ;ESC TO ATARI D
     UP
2050   BNE HEX
2060   LDA #$40 ;reset DOSINI for
      no-
2070   STA DOSINI ;trouble later
2080   LDA #$15
2090   STA DOSINI+1
2100   JSR $1540 ;fast reset DOSV
     EC
2110   JMP OLDDOS
2120 HEX CMP #'$ ;HEX TO DEC
2130   BNE DEC
2140   JSR HASCI ;hex to int
2150   JSR IASC ;int to dec
2160   BPL B2
2170 DEC CMP #'. ;DEC TO HEX
2180   BNE B2
2190   JSR DASCI ;dec to int
2200   JSR IHASC ;int to hex
2210 B2 BPL LSTCNC+2
2220 OPN LDA #OPEN ;IOCB MAIN SET
     UPS
2230 EXCMD LDX #$10
2240   BNE GTR2
2250 PTREC0 LDX #0
2260 PTREC LDA #PUTR
2270   BNE GTR2
2280 GETREC0
2290 GTREC LDA #GETR
2300 GTR2 STA ICBC,X
2310   LDA #$1E ;max rec length
2320   STA ICBLL,X
2330   LDA #LBUF/HI
2340   STA ICBAH,X
2350   LDA #LBUF&LO
2360 PGIN STA ICBAL,X
2370   LDA #0
2380   STA ICBLH,X
2390 ICB JSR CIO ;let the OS take
      over
2400   BMI ERR
2410 RTN2 RTS
2420 PUTCR2 LDA #PUTC
2430   BNE GETC2
2440 GETCR2 LDA #GETC
2450 GETC2 LDX #$10
2460   STA ICBC,X
2470   LDA #2
2480 JG STA ICBLL,X
2490   LDA #0
2500   STA ICBAH,X
2510   LDA #FR0
2520   BNE PGIN
2530 CLSE LDA #CLOSE
2540   BPL EXCMD
2550 PRINT STA FR0
2560   LDX #0
2570   LDA #PUTC
2580   STA ICBC,X
2590   LDA #1
2600   BPL JG
2610 ERR CPY #$03 ;ERR next read
     OK
2620   BEQ RTN2
2630   CPY #$88 ;ERR EOF OK too
2640   BEQ CINI
2650   TYA      ;store ERR
2660   PHA
2670   LDA #$C5 ;inverted E for e
     rror
2680   JSR PRINT
2690   JSR FR0Z ;clear FP0
2700   PLA      ;retrieve ERR
2710   STA FR0
2720   JSR IASC ;int to dec
2730 LSTCNC PLA ;clr stack of ret
     urn
2740   PLA
2750   JMP ST ;do not pass GO
2760 CINI LDA $FF
2770   BEQ LSTCNC
2780   JSR CHKIN
2790 JRUN LDA $2E1
2800   BEQ LSTCNC
2810   JSR JMPRUN
2820   BNE LSTCNC
2830 INPCON LDA #'@ ;HEX OR DEC#
     TYPE
2840   JSR PRINT ;the @ means AT/
     TO
2850    JSR GTREC
2860    LDY LBUF
2870    CPY #'.
2880    BEQ DASCI
2890    CPY #'$
2900    BEQ HASCI
2910    BNE ERR ;bad input
2920 DASCI JSR ASCFP ;DEC TO INT
2930    JMP FPI
2940 HASCI JSR FR0Z ;HEX TO INT
2950   LDY #1
2960 LB LDA (INBUF),Y
2970   CMP #$9B
2980   BEQ RTN
2990   SEC ;convert each digit
3000   SBC #$30
3010   CMP #$0A
3020   BMI SK7
3030   SBC #7
3040 SK7 LDX #4 ;times 16
3050 LA ASL FR0
3060   ROL FR0+1

3070   DEX
3080   BNE LA
3090   ORA FR0 ;add in new bits
3100   STA FR0
3110   INY
3120   BPL LB
3130 RTN RTS
```

```
3140 IHASC LDY #0 ;INT TO HEX SUB
     RT
3150     LDX #1
3160 LC LDA #$F0 ;hi mask
3170     AND FR0,X
3180     LSR A    ;roll into low b
     its
3190     LSR A
3200     LSR A
3210     LSR A
3220     JSR CONVH ;go conv to dig
     it
3230     LDA #$0F ;lo mask
3240     AND FR0,X
3250     JSR CONVH
3260     DEX
3270     BPL LC ;one more time
3280     BMI LE ;set eol and rtn
3290 CONVH CMP #$0A ;INT TO HEX D
     IGIT
3300     BMI SK9
3310     ADC #6 ;carry set
3320 SK9 ADC #$30 ;carry clr
3330     STA (INBUF),Y
3340     INY
3350     RTS
3360 IASC JSR IFP ;INT TO DEC
3370     JSR FPASC
3380 AREC LDY #0 ;CLR HI BIT/MAKE
      REC
3390 L9 LDA (INBUF),Y
3400    INY
3410    CMP #$80 ;find hi bit char
3420    BMI L9
3430    AND #$7F ;mask it
3440    DEY
3450    STA (INBUF),Y
3460    INY
3470 LE LDA #$9B ;set EOL
3480    STA (INBUF),Y
3490    JMP PTREC0
3500 SUBTR INC FR0 ;inclusive so
     inc
3510     BNE SK5 ;TO address
3520     INC FR0+1
3530 SK5 LDX #$10
3540     LDA FR1
3550     STA ICBAL,X
3560     LDA FR1+1
3570     STA ICBAH,X
3580     SEC ;CALC LENGTH
3590     LDA FR0
3600     SBC FR1
3610     STA ICBLL,X ;STORE AT IOC
     B
3620     LDA FR0+1
3630     SBC FR1+1
3640     JMP ICB-3 ;exec same cmd
     as last
3650 ;
3660 ;
3670 INIT JSR $1540 ;SRESET INIT
3680 JI LDA #END&LO
3690    STA MEMLO
3700    LDA #END/HI
3710    STA MEMLO+1
3720    LDA #ST&LO
3730    STA DOSVEC
3740    LDA #ST/HI
3750    STA DOSVEC+1
3760    RTS
3770 ;
3780 ;
3790 CMDTAB ;all spaces are one s
     pace
3800     .BYTE ">",$9B,$9B,"SD",$A
     1
3810     .BYTE " $>",$AE," .>",$A4
3820     .BYTE " SD",$D7," NR",$C0
3830     .BYTE " DO",$CC," VA",$D3
     ,$9B
3840     .BYTE "TR",$C3," TM",$C6
3850     .BYTE " SR",$C5," MN",$D2
3860     .BYTE " *N",$D5," KL",$AA
3870     .BYTE " RI",$C4,$9B
3880 TAB1
3890     .BYTE "RE*UF"
3900 TAB2
3910     .BYTE $20,$21,$23,$24,$FE
3920 ADRDIR
3930     .BYTE "D1:*.*",$9B
3940 ADRDOS
3950     .BYTE "D1:DOS.SYS",$9B
3960 END ;end after boot init exe
     cuted
3970 ;
3980 ;
3990 INIT1 LDA #INIT&LO ;BOOT INI
     T
4000   STA DOSINI
4010   LDA #INIT/HI
4020   STA DOSINI+1
4030   JMP JI
4040 ;
4050 ;
4060     *= $2E2 ;LOAD AND GO INIT
      ADR
4070     .WORD INIT1
4080     .END
```

Commodore Garbage Collection

Part 2

Jim Butterfield, Associate Editor

Last month, we looked into some of the causes of garbage collection delays, and investigated some of its working mechanisms. It's time to put our knowledge to work by developing some rules.

The following program will help us see the rules by means of examples:

```
100 DIM A$(800)
110 FOR J=1 TO 800
120 A$(J)="A"
130 NEXT J
140 PRINT "X"
150 PRINT FRE(0)
160 PRINT "Y"
```

Rules of Garbage Collection

Rule 1: There are *static* (in place) strings and *dynamic* (created) strings. Only dynamic strings have garbage collection consequences.

Proof: RUN the above program which contains only static strings. There will be no significant delay between the printing of X and Y. Now change line 120 to read:

```
120 A$(J)=CHR$(65)
```

RUN once again; there will be a significant pause between the printing of X and Y.

Rule 2: Garbage collection time depends on the number of dynamic strings you *keep*, not what you throw away.

Proof: Change line 120 to read:

```
120 A$(J)=CHR$(65):A$(J)="A"
```

RUN the program. Even though we're throwing away a large amount of garbage (the first A$(J)=), there's no significant delay.

Rule 3: Performing a garbage collection saves you no time on the next one.

Proof: Enter line 120 as:

```
120 A$(J)=CHR$(65)
```

RUN and note the delay. Now type: GOTO 140. Note that the delay is exactly the same as before; the previous collection saved us no time.

Rule 4: Doubling the number of strings will multiply the delay by 4. Mathematically, we can say that the time varies as the square of the number of strings.

Proof: Change the value of 800 in lines 100 and 110 to 400. RUN and note that the delay between the printing of X and Y drops to one-quarter of the previous time.

This last rule is the killer. You might work out a test program using ten strings, and when your program works satisfactorily expand to one thousand items. But your garbage collection time doesn't increase by a factor of 100; it jumps to 10,000 times the original delay. This could become crippling.

Fixing The Problem

If you know what to look for, you can usually avoid massive garbage collection delays. There's no single technique that will do the job. It's best to investigate what's causing the garbage and decide on the appropriate action to eliminate the problem.

Here's a list of techniques to get around the

SUPER FORTH 64'

TOTAL CONTROL OVER YOUR COMMODORE-64™
USING ONLY *WORDS*

MAKING PROGRAMMING FAST, FUN AND EASY!

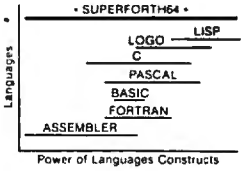
MORE THAN JUST A LANGUAGE...
A complete, fully-integrated program development system.

Home Use, Fast Games, Graphics, Data Acquisition, Business
Real Time Process Control, Communications, Robotics, Scientific, Artificial Intelligence

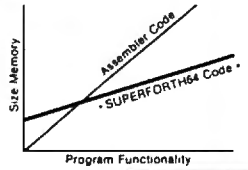
A Powerful Superset of MVPFORTH/FORTH 79 + Ext. for the beginner or professional

- 20 to 600 x faster than Basic
- 1/4 x the programming time
- Easy full control of all sound, hi res, graphics, color, sprite, plotting line & circle
- Controllable SPLIT-SCREEN Display
- Includes interactive interpreter & compiler
- Forth virtual memory
- Full cursor Screen Editor
- Provision for application program distribution without licensing
- FORTH equivalent Kernal Routines
- Conditional Macro Assembler
- Meets all Forth 79 standards+
- Source screens provided
- Compatible with the book "Starting Forth" by Leo Brodie
- Access to all I/O ports RS232, IEEE, including memory & interrupts
- ROMABLE code generator
- MUSIC-EDITOR
- SPRITE-EDITOR
- Access all C-64 peripherals including 4040 drive
- Single disk drive backup utility
- Disk & Cassette based. Disk included
- Full disk usage — 680 Sectors
- Supports all Commodore file types and Forth Virtual disk
- Access to 20K RAM underneath ROM areas
- Vectored kernal words
- TRACE facility
- DECOMPILER facility
- Full String Handling
- ASCII error messages
- FLOATING POINT MATH SIN/COS & SQRT
- Conversational user defined Commands
- Tutorial examples provided, in extensive manual
- INTERRUPT routines provide easy control of hardware timers, alarms and devices
- USER Support

SUPER FORTH 64' *is more powerful than most other computer languages!*



SUPER FORTH 64' *compiled code becomes more compact than even assembly code!*



A SUPERIOR PRODUCT in every way! At a low price of only **$96**

Call:
(415) 651-3160

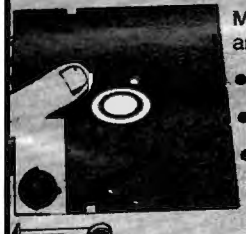
PARSEC RESEARCH
Drawer 1776, Fremont, CA 94538

Take this ad to your local dealer or B. Dalton Book store. Phone orders also accepted. Immediate delivery! Dealer inquiries invited. CA residents must include tax.

or C.O.D.

® PARSEC RESEARCH Established 1976 — Commodore 64 & VIC-20 TM of Commodore

*REDUCES DISKETTE COST 50%!



Make use of the back of your 5¼" Diskettes and SAVE

- ***NIBBLE NOTCH*** tools make it easy.
- Adds the precise notch where you need it.
- Doubles diskette space or MONEY BACK!

NIBBLE NOTCH I

Cuts square notch for users of Apple, II, II+, IIe, III, Franklin and Commodore.

only **$14.95** each
add $1.50 each P&H ($4.50 each foreign P&H)

NIBBLE NOTCH II

Cuts square notch and ¼ inch round *"index hole."* For use with computers other than those shown above.

only **$21.90** each
add $1.50 each P&H ($4.50 each foreign P&H)

Florida Residents Add 5% Sales Tax

SATISFACTION GUARANTEED OR YOUR MONEY BACK!

ORDER TODAY!

SEND CHECK OR MONEY ORDER TO:

NIBBLE NOTCH COMPUTER PRODUCTS
DIVISION OF CORTRAN INTERNATIONAL
4211 NW 75th TERRACE • DEPT. 8 7 • LAUDERHILL, FL 33319

PAT. PEND. — ALL TRADEMARKS ARE ACKNOWLEDGED

For the
Commodore 64

SuperTerm

Now only **$119.95**

Telecomputing with a difference!

SuperTerm — the only software that communicates with them all! Information networks such as CompuServe; business and university mainframes; free hobby bulletin boards.

Professionals and students: SuperTerm's VT102 emulation gets you on-line in style. Advanced video features, graphics, full-screen editing, 80/132 column through sidescrolling, extended keyboard — perfect for EDT, DECMail, etc. Even download your workfiles and edit off-line! Full printer and editor support; other emulations available.

Researchers and writers: SuperTerm's built-in text editor helps you create, edit, print, save, send and receive text files — articles, stories, reports, inventories, bibliographies — in short, it's your **information work station.** Access CompuServe, Dow Jones Information Network, Dialog/Knowledge Index, Western Union's Easylink, The Source, and many more. Optional Sprinter accessory saves printing time and $ (see below).

Computer hobbyists: Join in the fun of accessing hundreds of free bulletin board systems (BBS) for Commodore, Apple, TRS-80, etc. Text mode with all BBS systems; up/downloading with Commodore BBS systems (Punter protocol). Special protocol for up/downloading with other SuperTerm owners. Popular "redial-if-busy" feature for use with automodems.

Get the information you need, for business or for fun, with **the software that communicates with them all!**

Requires: Commodore 64, disk drive, and suitable manual- or auto-modem. Printer optional. Software on disk w/free backup copy. Extensive manual in deluxe binder.

SuperTerm's
SPRINTER Accessory $69.95

With the Sprinter accessory, SuperTerm can perform **concurrent printing** — as text appears on your screen, it's simultaneously printed on your printer. Includes all necessary hardware for connecting your **parallel printer** and computer via the cartridge port. Simply plug-in and go. Free utility software for printing and listing as a stand-alone interface.

Requires: parallel printer such as Epson, Gemini, Microline, C.Itoh. (Min. speed 35 cps.)

Commodore 64 is a trademark of Commodore Electronics, Ltd.

(816) 333-7200

Send for a free brochure.

MAIL ORDER: Add $1.50 shipping and handling ($3.50 for C.O.D.); VISA/Mastercard accepted (card# and exp. date). MO residents add 5.625% sales tax. Foreign orders payable U.S.$, U.S. Bank ONLY; add $5 shp/hndlg.

MIDWEST MICRO inc.

311 WEST 72nd ST. • KANSAS CITY • MO • 64114

garbage collection hang-up.

1. Don't Move Strings Around

Suppose we are writing a program to input several names and sort them into alphabetical order. It would seem logical to move the names so as to put them into the right place. Don't. Use an index array, which contains only numbers: Move the index values, not the strings.

A simple example:

```
100 PRINT "INPUT TEN NAMES"
110 DIM N$(10),I%(10)
120 FOR J=1 TO 10
130 PRINT "NAME";J;
140 INPUT N$(J)
150 I%(J)=J
160 NEXT J
170 PRINT "SORTING..."
180 FOR J=9 TO 1 STEP -1
190 FOR K=1 TO J
200 IF N$(I%(K)) <= N$(I%(K+1)) GOTO 220
210 I=I%(K):I%(K)=I%(K+1):I%(K+1)=I
220 NEXT K,J
230 FOR J=1 TO 10
240 PRINT N$(I%(J))
250 NEXT J
```

The above program uses a bubble sort technique, which is notoriously inefficient; but the point here is that the strings N$(..) are never moved. Thus, there can be no garbage collection. Note that the index array must be initialized before use—see line 150.

2. Clean Up Between Blocks

Suppose you're reading in a large file of students from various classes. For a number of reasons—especially processing convenience and shortage of memory—you don't read in all the students. Instead, you read and process a class at a time.

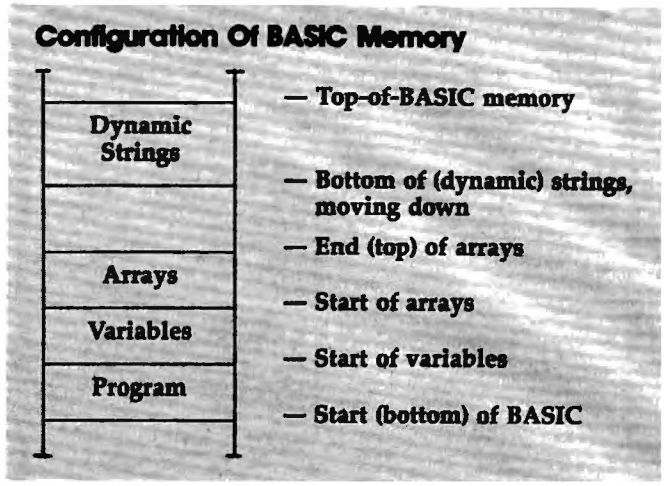
Before reading in the next class, set all student names, to null strings. Now, force a garbage collection with a statement such as Z=FRE(0). There will be few or no strings to keep, so garbage collection will be fast. When the next block of data—the next class—comes in, it will have freshly cleaned memory to use.

3. Do Local Cleanups

Many programs like to build strings from GET statements. The code often looks like this:

```
500 PRINT "TYPE IN YOUR NAME"
530 N=""
540 GET K$:IF K$="" GOTO 540
550 IF K$=CHR$(13) GOTO 600
560 N$=N$+K$
570 GOTO 540
600 REM CONTINUE .....
```

This sort of thing creates a lot of garbage. Every time line 550 is executed, a new N$ is created and the old one is thrown away; and N$ gets bigger and bigger all the time. There's also garbage from K$, but it's only a single character at a time.



If N$ and K$ were our only strings, we'd have no problem. Garbage collection time depends only on what you keep, not what you throw away; and keeping two strings isn't much work. However, if this were part of a program which also had a thousand names and addresses, we'd be in trouble; everything would need to be reclaimed, and the delays would become impractically long.

Local Collection

If we're careful, we can get around this problem by setting the stage for a "local" collection. We might reason as follows: During the above code, N$ and K$ are our only working strings. If we make all the other strings disappear momentarily, we may generate all the garbage we like, since garbage collections will be virtually instantaneous. When we're finished, we must carefully force one last collection to get rid of any leftover garbage, and then make these missing strings reappear.

We can do this by *temporarily* moving the top-of-BASIC pointer down to match the dynamic string pointer. This will fool the garbage collection routine into thinking that there are no dynamic strings except the ones we have just created. But we must remember to put the top-of-BASIC pointer back when the job is finished, or we'll suffer permanent loss of memory.

The top-of-BASIC pointer may be found on the VIC and 64 at addresses 55 and 56. We must save the values there so that we can replace them later, and then use the contents of the string pointer (51 and 52) to change the top-of-BASIC pointer. (In the PET/CBM, the top-of-BASIC pointer is at 52 and 53, and the string pointer is at 48 and 49. We'll show the programming for the VIC/64 below, but you may adjust it for your machine.)

Here's how we would change the above coding to eliminate garbage collection dangers:

```
500 PRINT"TYPE IN YOUR NAME"
510 A1=PEEK(55):A2=PEEK(56)
520 POKE 55,PEEK(51):POKE 56,PEEK(52)
530 N=""
540 GET K$:IF K$="" GOTO 540
550 IF K$=CHR$(13) GOTO 580
560 N$=N$+K$
570 GOTO 540
580 Z=FRE(0)
590 POKE 55,A1:POKE 56,A2
600 REM CONTINUE.....
```

It seems complex, and you must indeed program with great care. But it solves the problem.

4. Use Numeric Values

Who says that everything that seems alphabetic must be a string? A month can be coded 1 to 12; a grade of A to F can be a numeric from 1 to 6.

Where the number of possible strings is limited—a class, a region, an airline—using a numeric system is quite feasible. You can always look up the string you want by using the number as an index and getting the name out of an array.

I wouldn't recommend that we all lose our names and become numbers within the computer. But a little sensible data reduction can save a lot of garbage collection.

5. Brute Force

Sometimes conventional methods fail. Your data consists of a large number of names which have been read in from a file. You need to make changes to a substantial number of these names. There seems to be no way you can control the amount of garbage. What then?

Use The Disk

When all else fails, write out all your strings to disk. Set the strings to null values and force a garbage collection—this will take place instantaneously. Now read them back in to the newly cleaned-up memory.

You can watch the string pointer (addresses 51 and 52 on the VIC/64), and when it seems to be getting near the danger point, initiate this whole operation. At least it will be under your control; you can print a message to the user (TAKE A BREAK WHILE I UNSCRAMBLE MY BRAINS), and may even get the bonus of having generated a data backup or checkpoint in case of loss of power.

And it's a lot better than having the machine go dead for twenty minutes or more.

NEWS&PRODUCTS

Stress Reduction Software

Relax, a computer-controlled biofeedback system aimed at stress reduction, is available from Synapse Software for the Atari, Commodore, Apple, and IBM personal computers.

The system allows the user to observe and measure his or her stress levels on a video monitor or television set, and provides a method to attempt to reduce those levels.

A headband with three electromyograph (EMG) sensors measures tension in the forehead's frontalis muscle. The software converts these measurements into visual patterns designed to monitor the stress level. An audio tape has a program of therapeutic relaxation exercises, and a workbook provides guidelines for reducing stress and establishing a personal stress management profile.

Relax is available for $139.95.

Synapse Software
5221 Central Avenue
Richmond, CA 94804
(415) 527-7751

Apple Educational Games

Methods and Solutions, Inc., has announced its Mindplay line of educational software games that teach children from four years of age and up skills in measurement, following directions, memory, map reading, tactics, vocabulary, grammar, art, and mathematics.

The six educational games in the series include *Bake & Taste*, programs that teach youngsters to measure and follow directions; *Dyno-Quest*, a game of memory, map reading, tactics, and the discovery of dinosaurs; *Picture Perfect*, a joystick-based game that teaches children to draw and to color shapes, designs, and animals; *Race the Clock*, a matching game of words and hidden pictures; *Cat 'n Mouse*, a maze game using word and picture associations; and *Math Magic*, a monster-filled arcade game that teaches addition and subtraction.

The games are priced from $34.95, and are available for the Apple II family of computers and for the IBM PC and PCjr. They will be available for the Commodore 64 this fall.

Methods and Solutions, Inc.
300 Unicorn Park
Woburn, MA 01801
(617) 933-3298

Coleco Adam Data Packs

Victory Software has introduced blank data packs for the Coleco Adam computer. The blank, preformatted tapes store about 250 pages of information.

The tapes are available for a suggested price of $3.98.

Victory Software has also announced its new line of games for the Adam, including *Bounty Hunter* ($19.95), an Old West text adventure game.

Victory Software Corporation
1410 Russell Road
Paoli, PA 19301
(800) 243-1515
(215) 296-3787

Apple, IBM Classroom Software

Classmate, a classroom grading and attendance software package, has been released by Davidson & Associates for the Apple II, IIe and II+, and the IBM PC, PCjr and XT.

The program allows users to enter, modify, and store an unlimited number of class lists for up to 51 students. It stores grades, attendance records and teacher comments, and computes weighted averages, graphs grade distribution, class rankings and final grades, and displays or prints out all records.

The program also can sort by student name or class designation, and can display or print out individual scores, either on a particular assignment or for all assignments.

The program will give out either a single student's or a full class's complete or partial record.

In addition, the program can generate individualized parent and student reports.

Classmate is available for $49.95.

Davidson & Associates
6069 Groveoak Place, #12
Rancho Palos Verdes, CA 90274
(213) 383-9473

Foreign Language Tutorials

Soflight Software, a division of M. P. Computer Services Corporation, has introduced a new product line of foreign language development software.

The programs were de-

signed for the Apple II and IIe, with software for the Atari and IBM PC and PCjr to be available in the future.

One disk drive is required to run the program.

All programs teach 1000 of the most common words in the target language. Where words have more than one meaning, the program allows for those other meanings, along with English translation.

The package retails for $56.95. Languages currently available include Spanish, French, German, Italian, Biblical Hebrew, modern Hebrew, and Arabic. Latin, Russian, Polish, Swedish, and classical Greek will be available in the near future.

Each language program is menu-driven with sequential review, random review, and quiz options.

Soflight Software
2223 Encinal Station
Sunnyvale, CA 94087
(408) 735-0871

Personal Finances Software

A software product designed to help consumers make personal financial decisions has been announced by Electronic Arts. Called *Financial Cookbook*, the program contains "recipes," or formulas, that produce answers about money matters.

Through the program's 32 different recipes, users can figure such data as returns on investments, effective tax shelters and IRAs, effects of inflation, mortgage calculations, and tax rates.

Each recipe asks the user to enter variables, such as interest or inflation rates, and then makes calculations based on those numbers.

Calculations for 11 basic tax shelters available to most consumers are found in the recipes. The instruction manual includes a tutorial, recipe instructions, and index.

Financial Cookbook is available for the entire Apple II line, the IBM PC and PCjr, Commodore 64, and Atari 800.

Suggested retail price is $50.

Electronic Arts
2755 Campus Drive
San Mateo, CA 94403
(415) 571-7171

Text Adventure For Youngsters

Infocom has announced *Seastalker*, an interactive text adventure game for ages 9 and up.

In it, players aboard the specially equipped submarine Scimitar must save the Aquadome, earth's first undersea research station.

Unfortunately, the Scimitar hasn't been tested in deep water, and the crew of the Aquadome may have a traitor in its ranks. If the right course isn't charted, players might end up as shark bait.

Solving hints are included in the game package.

Seastalker is available for the Apple II, Atari, Commodore 64, IBM PC and PCjr, and TI-99/4A at a cost of $39.95.

Infocom, Inc.
55 Wheeler St.
Cambridge, MA 02138
(617) 492-1031

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.* ©

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 178,165. 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
```

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: [illegible] Enter these characters with the Atari logo key, {A}.

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -		Cursor Up
{DOWN}	ESC CTRL =		Cursor Down
{LEFT}	ESC CTRL +		Cursor Left
{RIGHT}	ESC CTRL *		Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

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

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

Commodore PET/CBM/VIC/64

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

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

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

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

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

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

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

Use the following tables when entering special characters:

VIC And 64

When You Read:	Press:		See:
{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.

CAPUTE!

Modifications Or Corrections To Previous Articles

Atari Snertle

Program 3 (p. 94) of this math tutorial from the May issue has a bug in its subtraction routine. In those cases when the answer to the displayed problem should be zero, a zero will not be accepted as the correct result. Donald Carlson points out that line 362 should read as follows:

```
362 IF Q=2 AND K<=L THEN M=L-K
```

64 Hi-Res Graphics Editor

The notes to this program (May issue, p. 82) failed to state the required starting and ending addresses to use when typing the MLX portion of the editor (Program 2). The values are 49152 for the start and 51553 for the end. Also, the series of steps required to set up the program may seem cumbersome. Andy Van Duyne has provided this short program, which will perform all the steps for you:

```
10 IF FL=0 THEN FL=1:LOAD"HIRES/ML",8,1
20 PRINT"{CLR}{2 DOWN}POKE642,128:POKE44,
   128:POKE32768,0:NEW"
30 PRINT"{3 DOWN}LOAD"CHR$(34)"HIRES/BAS"
   CHR$(34)",8"
40 PRINT"{HOME}";
50 POKE 198,6:POKE 631,13:POKE 632,13:POK
   E 633,13
60 POKE 634,82:POKE 635,213:POKE 636,13
```

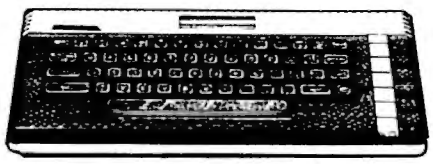
The program assumes you have used the filenames HIRES/ML for the machine language portion (typed in with MLX) and HIRES/BAS for the BASIC portion (Program 3). Change these names in lines 10 and 30 to match the names you used. To use the program with tape, change the 8 to a 1 in lines 10 and 30. You must have the BASIC portion saved on the tape immediately following the machine language portion.

The screen dump feature of the editor will not work with the new Commodore 1526 printer, since this model does not have the dot-addressable graphics feature of the Commodore 1525 printer. ©

COMPUTER MAIL ORDER

HOME COMPUTERS

ATARI



1010 Recorder	$74.00
1020 Color Printer	$249.00
1025 Dot Matrix Printer	$349.00
1027 Letter Quality Printer	$309.00
1030 Direct Connect Modem	$119.00
1050 Disk Drive	$399.00
CX30 Paddle	$12.00
CX40 Joystick	each $8.00
CX77 Touch Tablet	$64.00
CX80 Trak Ball	$48.00
CX85 Keypad	$105.00
488 Communicator II	$229.00
4003 Assembler Editor	$47.00
4011 Star Raiders	$33.00
4012 Missile Command	$29.00
4013 Asteroids	$29.00
5049 VisiCalc	$159.00
7079 Logo	$79.00

600XL	$189
800XL	$299
1200XL	CALL
1400XL	CALL

7101 Entertainer	$69.00
7102 Arcade Champ	$75.00
8026 Dig Dug	$33.00
8030 E.T. Phone Home	$33.00
8031 Donkey Kong	$39.00
8033 Robotron	$35.00
8034 Pole Position	$39.00
8036 Atari Writer	$79.00
8040 Donkey Kong, Jr.	$39.00
8043 Ms. Pacman	$39.00
8044 Joust	$39.00

ALIEN VOICE BOX

Atari	$119.00	Apple	$149.00

PRINTERS

AXIOM

AT-100 Atari Interface Printer	$239.00
GP-100 Parallel Interface	$199.00
AT-846 Interface for Atari	$89.00

BMC

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

CENTRONICS

122 Parallel	$299.00
739-1 Parallel	$199.00
739-3 Serial	$249.00

C.ITOH

Gorilla Banana	$209.00
Prowriter 8510P	$379.00
Prowriter 1550P	$599.00
A10 (18 cps)	$569.00
8600 P	$929.00
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	$949.00
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	$379.00
8025 Dot Matrix	$689.00
2010/15/30	$749.00
3510/15/30	$1369.00
7710/15/30	$1798.00

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	$399.00
Tractor Feed	$119.00

SILVER REED

500 Letter Quality	$449.00
550 Letter Quality	$569.00
770 Letter Quality	$899.00

STAR

Gemini 10X	$299.00
Gemini 15X	$399.00
Delta 10	$449.00
Serial Board	$75.00
Radix 10	$599.00
Radix 15	$699.00

TOSHIBA

1340	$869.00
1351	$1699.00

TRANSTAR

120P	$489.00
130P	$689.00
315 Color	$499.00

MEMORY BOARDS

Axlon 32K	$59.00	Intec 32K	$59.00
Axlon 48K	$99.00	Intec 48K	$85.00
Axlon 128K	$299.00	Intec 64K	$99.00
		Intec Real Time Clock	$29.00

DISK DRIVES FOR ATARI

PERCOM

AT 88-S1	$369.00
AT 88-A1	$259.00
AT 88-S1 PD	$449.00
RFD 40-S1	$369.00
RFD 40-S2	$699.00
RFD 44-S1	$539.00

INDUS

GT Drive	$379.00

RANA

1000	$329.00

TRAK

AT-D2	$389.00

MODEMS

ANCHOR

Volksmodem	$59.00
Mark II	$79.00
Mark VII (Auto Ans/Auto Dial)	$119.00
Mark XII (1200 Baud)	$299.00
TRS-80 Color Computer	$99.00
9 Volt Power Supply	$9.00

HAYES

Smartmodem 300	$209.00
Smartmodem 1200	$499.00
Smartmodem 1200B	$449.00
Micromodem IIe	$269.00
Micromodem 100	$299.00
Smart Com II	$89.00
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	$269.00

ZENITH

ZT-1	$339.00
ZT-10	$309.00
ZT-11	$369.00

DISKETTES

MAXELL

5¼" MD-1	$24.99
5¼" MD-2	$34.99
8" FD-1 (SS/DD)	$39.00
8" FD-2 (DS/DD)	$49.00

VERBATIM

5¼" SS/DD	$26.99
5¼" DS/DD	$36.99

BIB

5¼" Disk Head Cleaner	$14.99

ELEPHANT

5¼" SS/SD	$16.99
5¼" SS/DD	$21.99
5¼" FD/DD	$26.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Rip-n-File 10	$3.99
Rip-n-File 50	$17.99
Rip-n-File (400/800 ROM) Holder	$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	$54.99
Apple Analog	$37.99

AMIGA

3100 Single	$13.99
3101 Pair	$19.99
Joyboard	$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 PADS

Atari (ROM)	$79.00
C-64 (ROM)	$79.00
IBM	$99.00
Apple/Franklin	$85.00

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 4	$749.00

BMC

12" Green	$88.99
12" Green Hi-Res	$119.99
9191-13" Color	$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

HX-12 RGB	$529.00

SAKATA

100	$269.00

TAXAN

210 Color RGB	$289.00
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	$149.99
Pi 4, 9" Amber	$139.99
1400 Color	$269.99

QUADRAM

Quadchrome 8400 Color	$529.00

ZENITH

ZVM 122 Amber	$109.00
ZVM 123 Green	$89.99
ZVM 135 Color/RGB	$469.99

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

canada
Ontario/Quebec
800-268-3974
Other [illegible]0-268-4559
In Toronto [illegible]8-0866,Dept 105
Order Status Number: 828-0866
2505 Dunwin Drive, [illegible] 3B
Mississauga, Ontario, Canada L5L1T1

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

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. CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada.

MasterCard

COMPUTER MAIL ORDER

APPLE/FRANKLIN DISK DRIVES

MICRO-SCI

A2	$219.00
A40	$299.00
A70	$319.00
C2 Controller	$79.00
C47 Controller	$89.00

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

FRANKLIN

ACE 1000 Color Computer.....CALL
ACE Family Pack System.....CALL
ACE PRO PLUS System.....CALL
ACE 1200 Office Mgmt System.CALL
"NOT IIe EXPENSIVE"

SANYO

MBC 550.....CALL
MBC 555.....CALL

MBC 1100	$1499.00
FDD 3200—320K Drive	$389.00
MBC 1150	$1899.00
MBC 1200	$1849.00
FDD 6400-640K Drive	$469.00
MBC 1250	$2099.00
PR 5500 Printer	$599.00

NEC

PC-8201 $489

NEC

PC-8221A Thermal Printer	$149.99
PC-8281A Data Recorder	$99.00
PC-8201-06 8K RAM Chips	$105.00
PC-8206A 32K RAM Cartridge	$329.00

SHARP

PC-1500A $165.00
PC-1250A $88.00

CE-125 Printer/Cassette	$128.99
CE-150 Color Printer/Cassette	$171.99
CE-155 8K RAM	$93.99
CE-161 16K RAM	$134.99
CE-500 ROM Library	$29.99

HEWLETT PACKARD

HP 10C	$51.99
HP 11C	$69.99
HP 12C	$88.99
HP 15C	$88.99
HP 16C	$88.99
HP 75C	$749.99
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

HP 71B $449.00
41CV $208.00
41CX $249.00

commodore

CBM 8096	$869.00
CBM 9000	$999.00
B128-80	$769.00
8032 to 9000 Upgrade	$269.00
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
Silcon 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 Plus ..... each	$279.00
InfoPro	$179.00
Administrator	$399.00
Power	$79.00

M8D

SD1 Disk Drive	$349.00
SD2 Disk Drive	$599.00

SX-64 Portable $839

CBM 8032	$599
CBM 64	$199
C1541 Disk Drive	$249.00
C1530 Datasette	$69.00
C1520 Color Printer/Plotter	$129.00
M-801 Dot Matrix Printer	$219.00
C1526 Dot Matrix/Serial	$299.00
C1702 Color Monitor	$249.00
C1311 Joystick	$4.99
C1312 Paddles	$11.99
C1600 VIC Modem	$59.00
C1650 Auto Modem	$89.00
Logo 64	$49.00
Pilot 64	$39.00
Word Pro 64 Plus	$59.00
Parallel Printer Interface	$49.00
Calc Result 64	$65.00
Calc Result Easy	$39.00
Codewriter 64	$75.00
Quick Brown Fox	$49.00
MCS 801 Color Printer	$499.00
DPS 1101 Daisy Printer	$459.00
Magic Voice Speech Module	$54.00
Desk Organizer Lock	$49.00
Victex Telecommunications	$34.95

IBM

NEC PRINTERS

NEC 2050	$899.00
NEC 3550	$1639.00

PERCOM/TANDON DISK DRIVES

5¼" 320K Floppy	$229.00
5 Meg Hard w/Controller	$1049.00
10 Meg Hard w/Controller	$1349.00
15 Meg Hard w/Controller	$1599.00
20 Meg Hard w/Controller	$1899.00

AMDEK

310A Amber Monitor	$169.00
Color 2	$419.00
Color 4	$749.00

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	$89.00
Parallel Interface Board	$89.00
64K RAM Chips Kit	$59.00

CONTINENTAL SOFTWARE

1st Class Mail/Form Letter	$79.00
Home Accountant Plus	$88.00

MICROPRO

WordStar Professional Pack	$389.00
InfoStar	$299.00
SpellStar	$159.00
CalcStar	$99.00

MICROSTUF

Crosstalk	$105.00

MICROSOFT

MultiPlan	$159.00

ASHTON-TATE

dBASE II	$389.00
Friday!	$185.00

IUS

EasyWriter II	$249.00
EasySpeller	$119.00
EasyFiler	$229.00

LOTUS

Symphony	CALL
1-2-3	$329.00

PROFESSIONAL SOFTWARE

PC Plus/The Boss	$329.00

SYNAPSE

File Manager	$89.00

SOFTWARE

	C-64	Atari	IBM	Apple
Electronic Arts				
Axis Assassin	$17.95	$17.95	N/A	$17.95
M.U.L.E.	$29.95	$29.95	N/A	N/A
Pinball Construction	$29.95	$29.95	N/A	$29.95
Cut & Paste	$39.95	$39.95	N/A	$39.95
Hard Hat Mack	$29.95	$29.95	N/A	$29.95
InfoCom				
Witness	$32.95	$32.95	N/A	$32.95
Infidel	$32.95	$32.95	$32.95	$32.95
Starcross	N/A	$29.95	$29.95	$29.95
Planetfall	$32.95	$32.95	$32.95	$32.95
Enchanter	$32.95	$32.95	$32.95	$32.95
Zork	N/A	$29.95	$29.95	$29.95
AtariSoft				
Dig Dug	$37.95	N/A	$29.95	$29.95
Defender	$37.95	N/A	$29.95	$29.95
Robotron 2084	$29.95	N/A	$29.95	$29.95
PacMan	$37.95	N/A	$29.95	$29.95
Donkey Kong	$37.95	N/A	$29.95	$29.95
Spinnaker				
Story Machine	$24.95	$24.95	$29.95	$29.95
Cosmic Life	$24.95	$24.95	N/A	N/A
Up for Grabs	$29.95	$29.95	N/A	N/A
Kindercomp	$21.95	$21.95	$24.95	$21.95
Snooper Troops	$29.95	$29.95	N/A	$29.95
Delta Drawing	$29.95	$29.95	$39.95	$39.95
Kids on Keys	$29.95	$29.95	N/A	N/A
Hey Diddle Diddle	$21.95	$21.95	$21.95	$21.95
Facemaker	$24.95	$24.95	$24.95	$24.95
Rhymes & Riddles	N/A	$21.95	$24.95	$24.95
Dynatech				
Code Writer	$79.95	$79.95	$175.95	$155.95
VisiCorp				
VisiCalc	$159.95	$159.95	$159.95	$159.95
VisiCalc Advanced	N/A	N/A	B/A	$269.95
Visitrend/Plot	N/A	N/A	$199.95	$199.95
VisiFile	N/A	N/A	$199.95	$169.95
VisiSchedule	N/A	N/A	$199.95	$199.95
Desktop Plan	N/A	N/A	$199.95	$169.95
Business Forecast Model	N/A	N/A	$75.95	$75.95
StretchCalc	N/A	N/A	$75.95	$75.95
VisiTutor Calc	N/A	N/A	$59.95	$59.95
VisiTutor Advanced	N/A	N/A	$75.95	$75.95
VisiTutor Word	N/A	N/A	$59.95	$59.95
pfs:				
Write	N/A	N/A	$89.95	$79.95
Graph	N/A	N/A	$89.95	$79.95
Report	N/A	N/A	$79.95	$79.95
File	N/A	N/A	$89.95	$79.95
Solutions:* as low as	N/A	N/A	$16.95	$16.95

*Call on Titles

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

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

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

INTERNATIONAL ORDERS: All shipments 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.

VISA

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

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

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

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 **TO ORDER** VISA

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

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

RESTON 64
MINER 2049 ROM .$29.75

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

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

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

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

commodore 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

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
PROGRAMMERSG ...$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

SSI
KNIGHTS OF DESERT $26.75
EAGLES.....$26.75
TIGERS IN SNOW $26.75
COMBAT LEADER $26.75

SPINNAKER
KINDERCOMP D/R ..$20.75
STORY MACH D/R...$26.75

DATASOFT
SPELL WIZZARD $34.75
TEXT WIZZARD..... $34.75
WORD WIZZARD $46.75
ZAXXON C/D..... $26.75

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

PERCOM
AT88S1 ..$249.00
RDF40SI $269.00

GEMINI 10 ..... $259.00

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

VIDEO & COMPUTER REVOLUTION

BSD ENTERPRISES

800-346-5001 OR 212-438-4111

9 AM TILL 9 PM
800 346 5001
212 438 4111
SUNDAY THRU FRIDAY

ALL PRODUCTS ARE IN ALPHABETICAL ORDER. WE CARRY A FULL LINE OF LISTED BRANDS.
IF YOU DO NOT SEE WHAT YOU WANT, CALL AND ASK FOR IT.

WHOLESALERS, DEALERS AND OTHER DISTRIBUTORS INQUIRIES INVITED...
212-438-4111 ext. #18

COMPUTERS

ALTOS

586-20	5599 95
586-40	6999 95
986-40	8399 95

ATARI

600XL	135 95
800XL	234 95
1200XL	*CALL
1400XL	*CALL
1450XLD	*CALL

APPLE

IIe Single D D	*CALL
IIe Duo D D	*CALL
III	*CALL
LISA	*CALL
MACINTOSH	*CALL

COLECO

ADAM	524 95

COLUMBIA

1600 1V	2249 95
1600 4V	3399 95
1600 VP	2149 95

COMMODORE

VIC 20	75 95
C-64	175 95
EXEC 64	679 95

COMPAQ

COMPAQ	*CALL
COMPAQ +	*CALL

CROMEMCO

FULL LINE	*CALL

EAGLE

SPIRIT 2	2299 95
SPIRIT XL	3299 95
IIE-1	1435 95
IIE-2	1799 95
IIE-3	2199 95
IIE-4	3275 95
PC-1	2019 95
PC-2	2349 95
PC-XL	3059 95
1620	3039 95
1630	4349 95
1640	5599 95

EPSON

2X-10	*CALL

FRANKLIN

ACE 1000	*CALL
ACE FAM PACK	*CALL
ACE PRO +	*CALL
ACE 1200 OMS	*CALL

HEWLETT-PACKARD

41C	129 95
41CV	189 95
41CX	225 95
71B	369 95
75C	689 95
150	2899 95

I B M

PC	*CALL
PC jr	*CALL

I B M

PC	*CALL
PC JR	*CALL
PC-XT	*CALL

KPYPRO

II	*CALL
IV	*CALL
10	*CALL

NEC

PC8201A	539 95
HO1	1869 95
HO2	2344 95
HO3	2854 95
HO4	2378 95

OLIVETTI

M18 DT2	*CALL
M18-PT	*CALL

RADIO SHACK

100	664 95
4	829 95
4P	1499 95
12	2324 95
168	3299 95
2000	2299 95

SANYO

550	*CALL
555	*CALL
1100	1199 95
1150	1479 95
1160	1749 95
1200	1399 95
1250	1819 95
4000	1539 95
4050	1889 95

SHARP

PC 1250	75 95
PC 1500	135 95
PC 5000	*CALL

TELEVIDEO

TELEPORT PORT	*CALL
800A	999 95
802	2449 95
803	1799 95
802H	4299 95
806/20	4749 95
B16/40	8799 95
1602	*CALL
1603	*CALL

TEXAS INSTRUMENTS

PRO	1669 95

TIMEX-SINCLAIR

1000	21 95
2068	*CALL

DISK DRIVES

AMDEK

5¼"	199 95

ATARI

1050	279 95

COMMODORE

154	179 95

COMREX

10 MB	*CALL

CORVUS

6 MB	1559 95
11MB	2125 95
20 MB	2799 95

DAYONG

10 MB	1499 95
15 MB	1929 95
21 MB	2199 95

EAGLE

320 KB floppy	349 95
10 MB	1749 95
32 MB	2799 95

FRANKLIN

ACE 2	*CALL

MICRO-SCI

A-2	169 95
XL	139 95
A-82	359 95

PEGASUS

10 MB	1069 95
23 MB	1599 95

PERCOM

AT 88-S1	279 95
AT 88-S1PD	365 95
AT 88-A1	219 95
RFD 40-S1	345 95
RFD 40-S2	569 95
RFD 40-A1	219 95
RFD 44-S1	399 95
RFD 44-S2	679 95
5 MB	1299 95
10 MB	1499 95
15 MB	1899 95
20 MB	2229 95

QUADRAM

QUADISK-ENTIRE LINE	*CALL

RANA

1000	259 95
ELITE I	239 95
ELITE II	389 95
ELITE III	499 95
FOR I B M	239 95

SANYO

3200	*CALL
6400	*CALL
EDD & EFD	*CALL

TALLGRASS

12	*CALL
20	*CALL

TECMAR

10 MB	1995 95
15 MB	2299 95
33 MB	3099 95

XEROX

5¼" DUAL	569 95
8" DUAL	1049 95

MODEMS

ANCHOR

SIGNALMAN	59 95
MARK I	59 95
MARK II	59 95
MARK III	79 95
MARK V	79 95
MARK VI	159 95
MARK VII	89 95
MARK XII	239 95

CHARACTER GENERATOR

Attention all VIC 20 owners now you yes you can put any and all titles you want in any place in any color on your video tape for only

$79.95

HAYES

CHRONOGRAPH	169 95
SMARTCOM I	59 95
SMARTCOM II	74 95
MICROMODEM II	209 95
M/M II +	225 95
M/M IIE	209 95
M/M 100	259 95
S/M 300	199 95
S/M 1200	449 95
S/M 1200B	399 95

NOVATION

CAT	119 95
J-CAT	85 95
SMART CAT 103	145 95
SMART CAT 103/212	345 95
212 AUTO CAT	509 95
APPLE-CAT II	229 95
212 APPLE CAT	519 95
212 APPLE CAT UPGRADE	289 95
ACCESS 1-2-3-	369 95
D-CAT	129 95
EXEC INFONE	699 95
MAXI INFONE	849 95

TRANSEND

TRANSPAK I	265 95
TRANSPAK II	299 95
TRANSPAK III	369 95
P C MODEM	235 95
P C. TRANSPAK	324 95

U.S. ROBOTICS

PASSWORD	299 95
AUTO DIAL	399 95

MONITORS

AMDEK

12" GR	109 95
12" AMB	119 95
13" COL	249 95
13" COL RGB	389 95

BMC

12" GR	69 95
12" AMB	72 95
13" COL (9191)	199 95

COMMODORE

170	189 95

HITACHI

CM 1481	*CALL

EAGLE

12" GR	229 95
14" COL RGB	479 95

GORILLA

12" GR	69 95
12" AMB	79 95

NEC

9" GR	119 95
12" GR	89 95
12" AMB	135 95
12" COL	249 95
12" COL RGB	389 95

PANASONIC

12"	GR
12" GR	89 95
12" AMB	135 95
12" COL	249 95
12" COL RGB	389 95

PANASONIC

12" GR	129 95
12" AMB	139 95
10" COL	219 95
13" COL RGB	309 95

PRINCETON GRAPHICS

12" COL RGB	469 95

SAKATA

12" GR	*CALL
13" COL	*CALL

QUADRAM

QUADCHROME	459 95
QUADCOLOR I	179 95

SANYO

ENTIRE LINE	*CALL

TAXAN

12" GR	99 95
12" AMB	109 95
12" COL	239 95
12" COL RGB	259 95

USI

FULL LINE	*CALL

ZENITH

FULL LINE	*CALL

PLOTTERS

AMDEK

DXY 100	519 95
AMPLOT II	*CALL

ENTER

SWEET-P	419 95
SWEET-P GRAPH PKG	669 95
SWEET-P PRIME	619 95

HEWLETT-PACKARD

7470 A	769 95
7475	1389 95

MANNESMAN TALLY

PIXY-3	*CALL

PANASONIC

VP 6801-A	1179 95

BMC

X-Y	749 95

PRINTERS

ATARI

1020	189 95
1025	269 95
1027	246 95

ATARI

1020	189 95
1025	269 95
1027	246 95

AXIOM

ENTIRE LINE	*CALL

BROTHER

HR 15	389 95
HR 25	589 95

BMC

BX 80	199 95
PB 401	569 95
BX 180	249 95

C. ITOH

PROWRITER	299 95
STARWRITER	849 95
PRINTMASTER	1139 95

CENTRONICS

ENTIRE LINE	*CALL

COMMODORE

1526	225 95
MPS-801	179 95

ANADEX

ENTIRE LINE	*CALL

COMREX

ENTIRE LINE	*CALL

DAISYWRITER

1500	739 95
2000	879 95

DATA PRODUCTS/IDS

PRISM 80	899 95
REST OF LINE AND CONFIGURATIONS	*CALL

DIABLO

620	819 95
630	1559 95

EPSON

ENTIRE LINE	*CALL

GORILLA

BANANA	129 95

JUKI

6100	389 95

NEC

3550	1529 95
REMAINING LINE	*CALL

OKIDATA

80	129 95
82	269 95
83	499 95
84	899 95
92	369 95
93	639 95
2350	1859 95
2410	2099 95

MANNESMAN TALLY

160L	539 95
180L	739 95
SPIRIT 80	269 95

OLIVETTI

P-41	*CALL

PANASONIC

KXP 1090	249 95
KXP 1160	999 95

QUADRAM

QUADJET	599 95

SANYO

5500	589 95

SHARP

125, 150	*CALL

SILVER REED

500	369 95
550	499 95
770	799 95

SMITH CORONA

TP-1, TP-2	*CALL

STAR MICRONICS

CEMINI 10	*CALL
GEMINI 15X	*CALL
DELTA 10	*CALL

TOSHIBA

1350	*CALL

TRANSTAR

ENTIRE LINE	*CALL

TERMINALS

ADDS

VPT-A1	449 95
VPT-A2	449 95
VPT-COL	1049 95
VPT-3A	449 95

HAZELTINE

ESPRIT I	439 95
ESPRIT II	459 95
ESPRIT III	529 95
ESPRIT COLOR	759 95
ESPRIT 6310	459 95

LEAR SK-6LER

ADM-3A	509 95
REMAINING LINE	*CALL

SANYO

COLOR RGB	519 95

TELEVIDEO

910	529 95
925	669 95
950	879 95
970	959 95

MULTI FUCTION BOARDS

AST

FULL LINE	*CALL

QUADRAM

FULL LINE	*CALL

GRAPHIC CARDS

AMDEK	*CALL
HERCULES	*CALL
PLANTRONICS	*CALL
QUADRAM	*CALL
USI	*CALL

DISKETTES

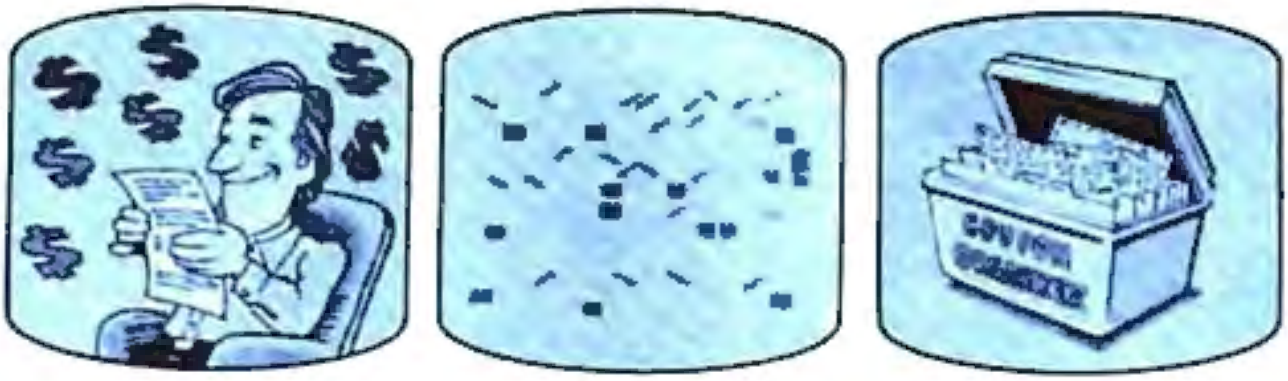
FULL LINE	*CALL

*CALL

WE DO NOT ONLY MEET PRICES...
WE BEAT
EVERY LEGITIMATE PRICE
800-346-5001
212-438-4111

TO ORDER-Simply dial toll free 800-346-5001 or 212-438-4111. You may order with Master Card or Visa, or you may send a money order, cashier's check, or certified chect to: **BSD ENTERPRISES**, 1781 64th St., Brooklyn, N.Y. 11204, and add approximate shipping, handling and insurance charges. Credit cards for phone orders only. All items subject to availability and price change. **NO TAX** for orders shipped out of state. **ALL MERCHANDISE SHIPPED BRAND NEW, FACTORY FRESH AND 100% GUARANTEED.** For customer service please dial **212-438-4111, ext. #17.**

Advertisers Index



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.

347801

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	111
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 9/30/84 CO784

Place
Postage
Here

COMPUTE! Reader Service
P.O. Box 2141
Radnor, PA 19089

"The Complete CARDCO Line"

. . . and still growing!

CARDCO provides "Commodore-ready" computer accessories that will enhance your utilization of Commodore-64 and VIC-20 Computers, increase their capability, and add to your enjoyment and skill. AND, they're available for use with other personal computers, too.

Designed with the user in mind, CARDCO offers fine accessories including Printer Interfaces with and without graphics, Expansion Interfaces, Memory Expansions, Cassette Interfaces, Numeric Keypads PLUS "NOW" Software for your VIC-20 and C-64. These programs include the "WRITE NOW" Word Processor, "MAIL NOW" Mailing List, PRINTER UTILITY PROGRAMS on Tape and on Disk, "SPELL NOW" Spell Checker, "GRAPH NOW" including "PAINT NOW", and "FILE NOW".

CARDCO has three new Letter Quality PRINTERS with your choice of drumhead design (8 1/2" carriage), Daisy Wheel Design (13 inch carriage) and Daisy Wheel Design (11 inch carriage). "Commodore-ready" . . . plus; with compatible input for PC, PC jr., TRS-80 and many more personal computers. CARDCO's NEW "DATA CASSETTE RECORDER/PLAYER" is also "Commodore-ready" and ready for instant shipment at prices that will amaze you.

CARDCO will constantly increase its line with unique and new products to enhance the enjoyment of computer owners.

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