Light Pens And Graphics Tablets: How To Use Them

COMPUTE!

$2.95
May
1984
Issue 48
Vol. 6, No. 5
£2.25 UK $3.25 Canada
02193
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

3-D Surface Plotting, A Valuable Graphics Technique

Picture Perfect, A Friendly, Creative Drawing Program For Youngsters

Pentominos, A Fascinating Puzzle-Solving Program For Commodore 64, VIC-20, TI-99/4A, PC And PCjr, Atari, And Apple

Plus Snertle, A Challenging Game For The VIC-20, Atari, Commodore 64, And IBM PC And PCjr

And More

0 71486 02193 05

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.

WordPro 3 Plus™/64 and SpellRight Plus™ are trademarks of Professional Software Inc.
The WordPro Plus Series was designed and written by Steve Punter of Pro-Micro Software Ltd.
SpellRight Plus was designed and written by Dwight Huff and Joe Spatafora of SpellMaster Systems, Inc.
Some printers may not support certain WordPro 3 Plus functions and/or require an interface. Please check with your dealer.
Commodore 64™ is a trademark of Commodore Electronics Ltd.

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

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

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

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

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

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

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

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

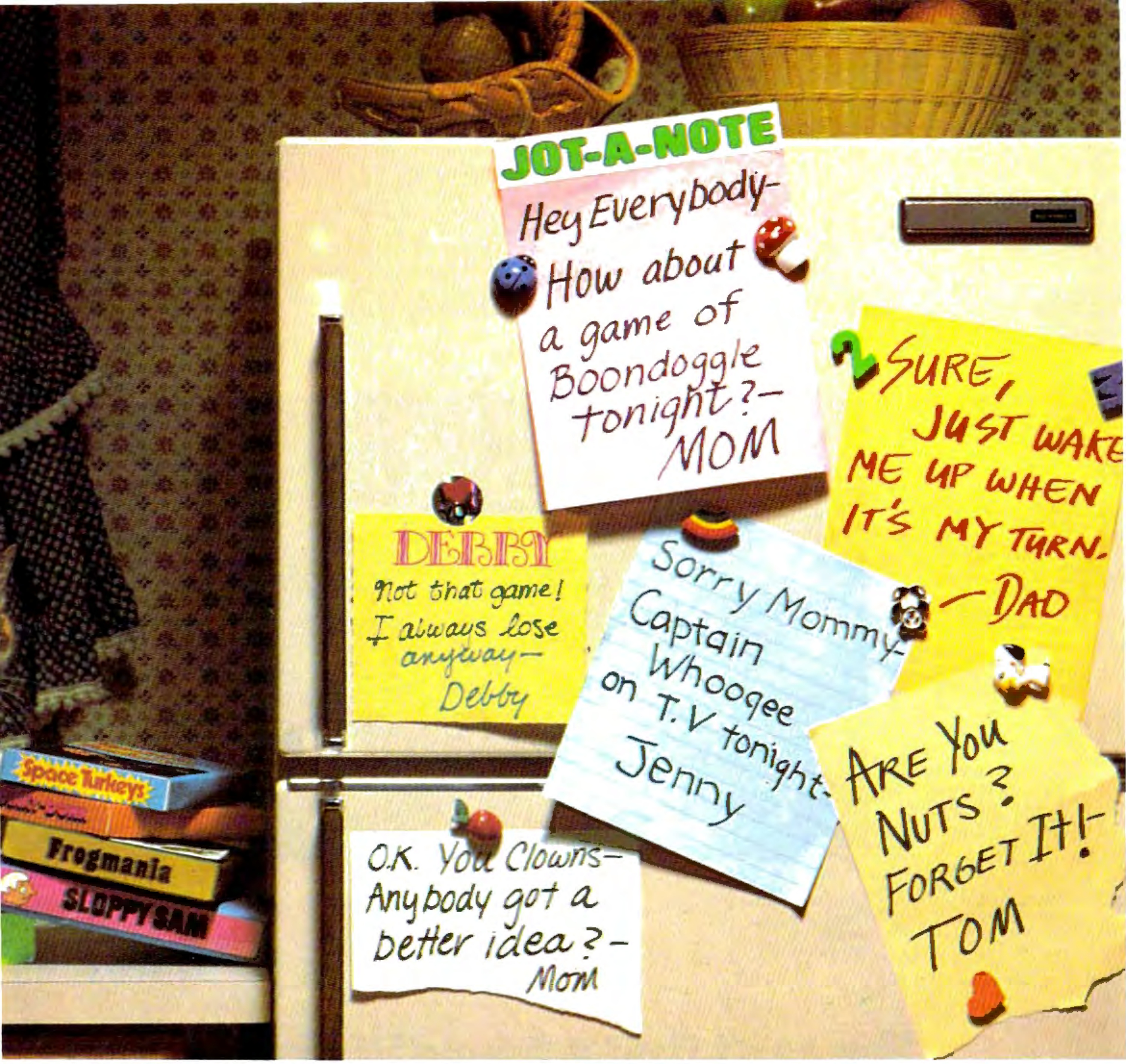
Spinnaker games are available for ColecoVision® and Coleco Adam.™ Commodore 64™ and Atari® home computers.

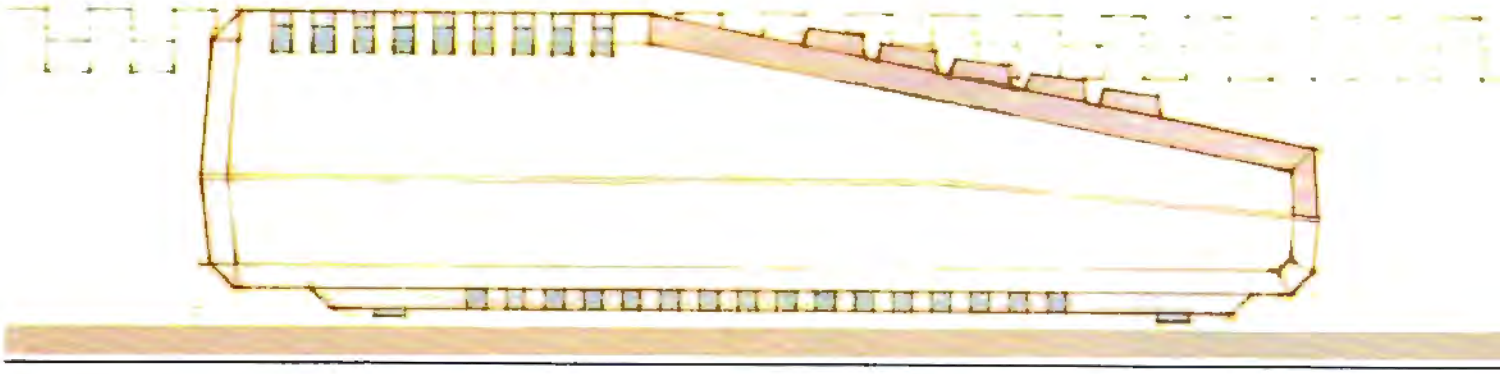
It's New! JUKEBOX.™

It's a musical strategy game that's kind of a cross between a chess match and a dance contest. To keep the music bopping, you've got to keep your feet hopping – but don't get boxed in, or the jukebox will stop!
Ages 8 - Adult.

It's New! AEGEAN VOYAGE.™

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


JOT-A-NOTE
Hey Everybody-
How about a game of Boondoggle tonight?-
MOM
SURE, JUST WAKE ME UP WHEN IT'S MY TURN.
-DAD
DEBBY
not that game! I always lose anyway-
Debby
Sorry Mommy-
Captain Whoogee on T.V tonight-
Jenny
ARE YOU NUTS? FORGET IT!-
TOM
O.K. You Clowns-
Anybody got a better idea?-
Mom
Space Turkeys
Frogmania
SLOPPY SAM

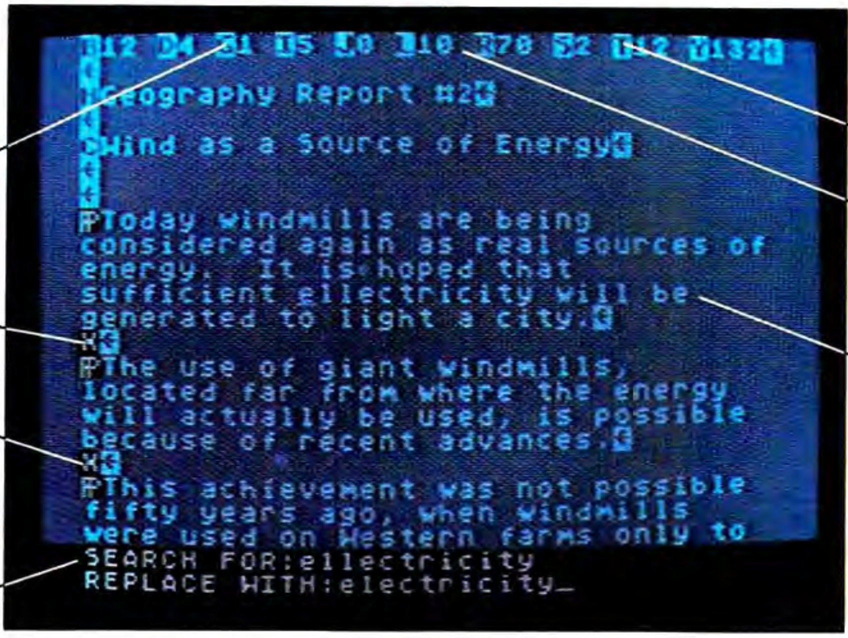


Prints with built-in format or lets you create your own: center, underline. Boldface, elongated, proportional and condensed print.

Inserts and deletes characters, lines or blocks of text.

Duplicates and moves blocks of text.

Searches for and/or replaces words or phrases, one at a time or all at once.



Changes line spacing, margins, print type, paragraph indents anywhere in the document.

Instantly reformats.

Enters text easily, with word wrap, one main menu and helpful prompts.

Creates multi-line headers and footers.

Prints double columns, form letters, multiple copies of a page or document, at the touch of a key.

Numbers pages and creates section numbers automatically.

Corrects spelling (with extra software).

Chains and merges files.

Has "print preview": you can check your copy before it's printed.

You can't find a friendlier, more powerful word processor at twice the price. New AtariWriter.™ Under $100.

Now you can do multi-featured word processing at home, simply. At a family budget price.

Our ROM-based cartridge technology means you can use new AtariWriter on any ATARI® Home Computer (even 16K) for personal and business correspondence, term papers, committee reports, mailings, etc.

It also lets you choose between cassette and disk storage systems.

One very special AtariWriter feature: you can correct as you write, without switching back and forth between Create and Edit modes.

And our memory buffer offers an "undo" command to let you change your mind, and restore text you've just deleted.

Check into our remarkable AtariWriter, and our choice of letter quality and dot matrix printers, also reasonably priced, at Atari dealers. Call 800-538-8543 for dealer nearest you. In California, call 800-672-1404.

You'll do more with Atari Home Computers.

May 1984 Vol. 6, No. 5

FEATURES

EDUCATION AND RECREATION

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 179 before typing in 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.

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, Richard Mansfield, senior editor of COMPUTE!, *expresses some concerns in this guest editorial about the way programming is taught in schools.*
Robert Lock
Editor In Chief

Which computer language is best? Ask that question at a computer club and you're sure to start a debate. But computer users rarely have much choice in the matter. If you buy a personal computer, you'll get BASIC. It's usually built into the computer. If you learn programming at school, you'll get Pascal. It's built into the curriculum.

Of course, other languages can be purchased for personal computers and are sometimes taught in schools, but BASIC and Pascal are by far the most common ways that most people are learning to communicate with computers.

Why is it that Pascal isn't built into consumer computers and BASIC is frowned upon by academics? What's the difference between these languages? Is Pascal the easier language to learn? Or is it just the easier language to *teach*.

The goal of a teacher is to pass knowledge, even wisdom sometimes, to the student. Good teaching accomplishes this transfer with a minimum of damage to the student's creativity and freedom of thought. But like all human activities, teaching can go awry.

On the first day of driver education, the teacher told me and the two young women in my group that we couldn't get into the car until we'd promised to follow the Three Rules of Good Driving. Evidently there had been some hair-raising moments in the past and these rules were for everyone's safety. 1. Keep your eyes on the road at all times. 2. Keep both hands on the wheel at all times. 3. Always use the turn signal, but also roll down the window and signal with your hand too. This last rule struck us as perhaps excessive. For one thing, we'd never seen anyone driving like that. And doesn't rule 3 violate rule 2?

Never mind, that's the way to drive. As the weeks progressed, other strange rules were added: Don't adjust the mirrors or the seatbelt while in motion, never converse with other passengers, and so on. We followed the rules, but of course discovered later that these were not realistic guidelines. Some of what we had been taught were the Rules of Good Driver Education, as distinct from rules of good driving.

I suggest that Pascal is not easier to learn than BASIC. Nor is Pascal more flexible or faster to program in than BASIC. In fact, Pascal has no significant advantage over BASIC save one—it is easier to grade.

That's because Pascal and languages like it stress *structured programming*. Pascal has more rules than BASIC. For example, in BASIC you can create variables anytime you want to. Just say INCOME = 15000 and that's that. In Pascal, you must define your variables at the start of the program. You must declare whether they're integer, string, floating point, etc.

Another rule associated with Pascal is program formatting: Loops should be indented, each programming event should be on its own line, and subroutines should be set off by additional spacing.

A third rule is possibly the most confining: You are not allowed to GOTO. In BASIC, this command allows you to branch to any other instruction in the program. And you can keep on branching at will. Pascal permits branching, but you must always return to the place from which you branched.

Forbidding GOTO branches is the keystone of structured programming, and it has an important effect on the way a student approaches programming. Before actual programming can begin, the programmer must plan the structure of the program. This is analogous to the requirement imposed by some English teachers that no one should begin writing an essay until they've first constructed a detailed outline. In Pascal classes, flowcharts abound.

Pascal, of course, is not a terrible way to program computers. And BASIC isn't perfect. They differ mainly in the psychological effects they have on programmers. But if the primary virtue of Pascal is that it is the easier language to teach, maybe some questions should be raised. The most important question might be—is Pascal the best language to learn?

What's worrisome about Pascal's emphasis on preplanning and its blizzard of rules is that such academic programming might be the only experience many people will ever have with computer programming. They might assume that all computer languages are restrictive. They might never go on to discover that communicating with computers can be an exceptionally rewarding, even entertaining, pastime.

follow our path to the future of home computing.

Welcome to the next generation of home computer software and hardware...from **Futurehouse**. We help you get the most out of your computer. The most personal productivity, the most education, the most entertainment. The most for your money. Follow our path to the future...

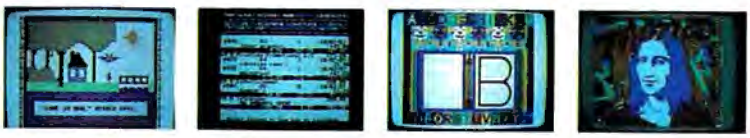
For your financial future...the **Complete Personal Accountant** is an award-winning line of money management software.

For your creative future...the **Edumate Light Pen** is a low cost, high performance peripheral which draws, entertains and teaches. It is rated the best in its price range and out performs even the most expensive light pens.

For your child's future...**Playground Software**, our educational series, uses the **Edumate Light Pen** and S.A.M. (Software Automatic Mouth) to teach and delight your children. The series includes **Alphabet Construction Set**, a unique program that teaches children how to draw the letters of the alphabet.

For your artistic future...with **Peripheral Vision** and an **Edumate Light Pen** you can create sophisticated works of art on your screen. Choose from dozens of advanced graphics routines and then save your artwork to disk or print it on your printer.

Let **Futurehouse** lead you into the future with quality products for your home computer. Contact your local dealer or order direct 1-800-334-SOFT. Don't wait for the future...it's here.



futurehouse

Futurehouse products are available for Commodore 64/Vic 20, Atari, TRS-80 Color, and IBM PC jr. computers. When ordering please specify computer, cassette or disk and memory.

Complete Personal Accountant, Edumate Light Pen, Playground Software, and Alphabet Construction Set are trademarks of Futurehouse, Inc. Box 3470, Chapel Hill, NC 27514. Commodore 64, Vic 20, Atari, TRS 80 Color, and IBM PC jr. are registered trademarks of Commodore Electronics, Ltd., Atari, Inc., Tandy Corp., and International Business Machines Corp., respectively.

Coming In Future Issues

Choosing The Right Printer

A Survey Of Inexpensive Printers

Three Challenging Games For Several Computers: Pests, Devastator, And Olympiad

Lightning Sort

All About Commodore Cassettes

Atari Macrodos

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

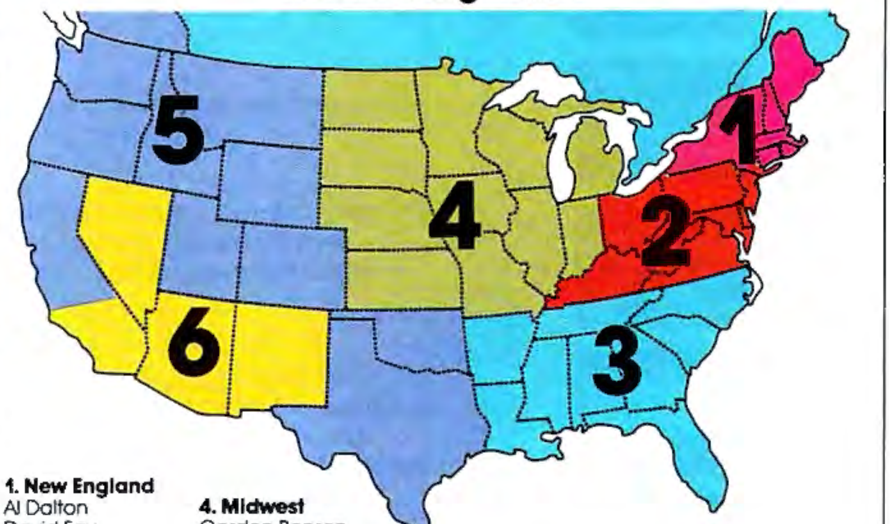
TOLL FREE Subscription Order Line
800-334-0868 In NC 919-275-9809

COMPUTE! Subscription Rates (12 Issue Year):

US (one yr.) $24
(two yrs.) $45
(three yrs.) $65
Canada and Foreign Surface Mail $30

Air
Europe, Australia $42
Middle East, Central America and North Africa $52
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

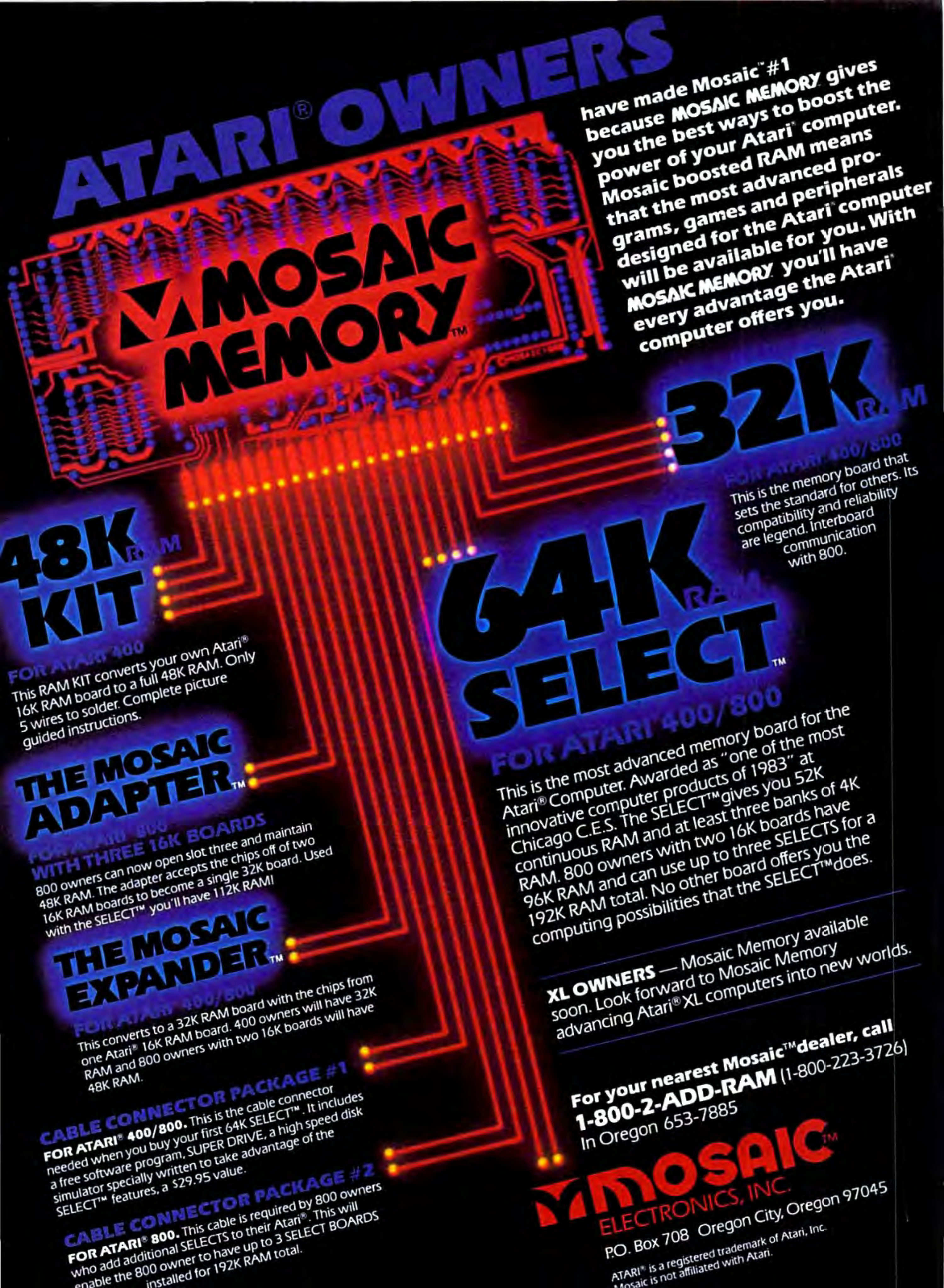
6. Southwest
JoAnn Sullivan
619-941-2313
Ed Winchell
213-378-8361

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

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE!, P.O. Box 914, Farmingdale, NY 11737. Include a note indicating your preference to receive only your subscription.


ATARI® OWNERS
have made Mosaic™ #1 because MOSAIC MEMORY gives you the best ways to boost the power of your Atari® computer. Mosaic boosted RAM means that the most advanced programs, games and peripherals designed for the Atari® computer will be available for you. With MOSAIC MEMORY you'll have every advantage the Atari® computer offers you.
MOSAIC MEMORY™
32K RAM
FOR ATARI 400/800
This is the memory board that sets the standard for others. Its compatibility and reliability are legend. Interboard communication with 800.
48K RAM KIT
FOR ATARI 400
This RAM KIT converts your own Atari® 16K RAM board to a full 48K RAM. Only 5 wires to solder. Complete picture guided instructions.
64K RAM SELECT™
FOR ATARI 400/800
This is the most advanced memory board for the Atari® Computer. Awarded as "one of the most innovative computer products of 1983" at Chicago C.E.S. The SELECT™ gives you 52K continuous RAM and at least three banks of 4K RAM. 800 owners with two 16K boards have 96K RAM and can use up to three SELECTS for a 192K RAM total. No other board offers you the computing possibilities that the SELECT™ does.
THE MOSAIC ADAPTER™
FOR ATARI 800 WITH THREE 16K BOARDS
800 owners can now open slot three and maintain 48K RAM. The adapter accepts the chips off of two 16K RAM boards to become a single 32K board. Used with the SELECT™ you'll have 112K RAM!
THE MOSAIC EXPANDER™
FOR ATARI 400/800
This converts to a 32K RAM board with the chips from one Atari® 16K RAM board. 400 owners will have 32K RAM and 800 owners with two 16K boards will have 48K RAM.
XL OWNERS — Mosaic Memory available soon. Look forward to Mosaic Memory advancing Atari® XL computers into new worlds.
CABLE CONNECTOR PACKAGE #1
FOR ATARI® 400/800. This is the cable connector needed when you buy your first 64K SELECT™. It includes a free software program, SUPER DRIVE, a high speed disk simulator specially written to take advantage of the SELECT™ features, a $29.95 value.
CABLE CONNECTOR PACKAGE #2
FOR ATARI® 800. This cable is required by 800 owners who add additional SELECTS to their Atari®. This will enable the 800 owner to have up to 3 SELECT BOARDS installed for 192K RAM total.
For your nearest Mosaic™ dealer, call
1-800-2-ADD-RAM (1-800-223-3726)
In Oregon 653-7885
MOSAIC™ ELECTRONICS, INC.
P.O. Box 708 Oregon City, Oregon 97045
ATARI® is a registered trademark of Atari, Inc.
Mosaic is not affiliated with Atari.

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

Can Disks Be Mailed?

Should disks be mailed, and if so, what is the proper way to mail them?

Brian Mangan

Disks can be mailed, as long as they are enclosed in a snugly fitting, rigid package. Many office supply stores sell padded jackets (called mailers) especially made for 5¼-inch disks. Also, for what it's worth, many users write a message on the outside of the mailer, to warn mail handlers that the package contains a magnetic recording which can be damaged by electromagnetic fields.

Commodore Sequential Append

I recently made a discovery that I think will help programmers using Commodore disk drives to create and use sequential files. In addition to writing a sequential file (OPEN 2,8,2,"SEQFILE,S,W") and reading a sequential file (OPEN 2,8,2, "SEQFILE,S,R"), it is possible to append a sequential file. This is a great help; rather than having to rewrite the entire file when additions are made (OPEN 2,8,2,"@0:SEQFILE,S,W"), all you have to do is use an A in place of the W when you open the sequential file for writing: OPEN 2,8,2, "SEQFILE,S,A". The DOS finds the end of the file and simply adds on the new data. You use the regular PRINT#2 statement to accomplish this.

Steve Gibson

Disabling The Atari Break Key

I want to inform your readers about a technique I discovered that disables the Atari's BREAK key, but does not need to be reexecuted after each GRAPHICS command. It is so simple that I wonder why no one has ever mentioned it, or if it conflicts with something that I have not yet found out:

POKE 566,143:POKE 567,231 to disable

and

POKE 566,84:POKE 567,231 to enable

The preceding statements change the BREAK key interrupt vector to point to address 59279 ($E73F) which contains a machine code PLA and RTI instruction used by the OS. This method will work only with the OS B ROMs, which contain the interrupt vector for the BREAK key.

Neil Weisenfeld

A TI Quit Fix

Have you ever hit FUNCTION+ instead of SHIFT + while you are typing in a program? It's extremely frustrating to see all your work go down the drain. Here's a way to disable the QUIT key on the TI.

To do this you will need either the Mini Memory or Editor/Assembler cartridge or Extended BASIC and the 32K Memory Expansion. This is because the console BASIC does not contain the CALL LOAD subprogram (better known as POKE). Whenever you turn your computer on, type the following line in the command mode: CALL LOAD(–31806,16). This will disable the QUIT key. If you are using Extended BASIC, use CALL INIT::CALL LOAD(–31806,16). If you wish to return to the Master Title Screen, you can still do so by typing BYE.

Credit for this information goes to the documentation that comes with the TI Forth package.

By the way, does anybody know of a comprehensive memory map for the TI?

Davin A. Trulsen, Jr.

What's An EPROM?

I would like to know what EPROMs are and what they are used for.

Bob Cullen

EPROM stands for Erasable Programmable Read Only Memory. EPROMs are memory chips which can "remember" programs even when the computer's power is switched off. Important machine language programs like the BASIC language or the computer's operating system are often permanently stored in ROM, but standard ROM can be programmed only once (when the chip is made). EPROMs, on the other hand, can be programmed by any computer user with a relatively simple peripheral device, the EPROM programmer. EPROMs can also be erased by exposing them to ultraviolet light. You could use an EPROM to store any machine language program you use frequently—even to make your own game cartridges.

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

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

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

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

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

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

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

CompuServe

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

800-848-8199

In Ohio call 614-457-0802

An H&R Block Company

64 Sprite Collisions

I have a Commodore 64, and am having trouble with collision detection with sprite graphics. I use the following line to check for collisions:

```
IF (PEEK(53278)ANDX) = X THEN action
```

This is easy to convert to machine language. In all of my programs, this statement is unreliable. Sometimes it detects a collision between two sprites when they aren't colliding, other times it doesn't detect a collision when they are touching, and other times it works just fine.

I've read in past articles that this problem may be caused by "sparkle" on the 64, and that the solution to the problem is to relocate screen memory. I tried that, and it didn't help.

I've also found that by putting a PRINT PEEK(53279) in my programs, the collision registers work every time. But I don't know how to PRINT a PEEK in machine language.

Eric Rotenberg

First, sparkle can cause spurious collisions with sprites, but you have to relocate the character set, not the screen, to disable the sparkle. Second, be aware of the nature of the collision register. It is set when two sprites collide, and stays set, even after the sprites have moved away from each other.

Also, the register is cleared when you try to read it, so you can't keep doing an LDA or a PEEK to check for different collisions. The first PEEK resets the register. If the sprites are still touching, they will then set the collision register again. When you are checking for a collision, save the results of the first PEEK for later use.

BASIC B For The Atari 400 And 800?

1. Is Atari going to make a Revision B of BASIC, as found in the new XL series on cartridge or other form for the 400 and 800 computers?

2. I've been having trouble with my BASIC cartridge. *Pac-Man* works just fine, but when I plug in BASIC, either the screen goes blank, or I get two clicks and the screen goes blank, or it goes right into memo pad mode. This happens after I put in any other cartridge. Can anyone help me?

Kevin Bailey

As far as we know, Atari has no plans for offering an upgraded BASIC.

Even though ROMs are sturdy, solid-state devices, they can be damaged by static electricity or by being dropped. It's a good idea to ground yourself (by touching something made of metal) before you operate any computer equipment. But your BASIC's not necessarily bad. You may just need to clean the contacts.

Normally, the contacts are not exposed, but you can stick a pencil or paper clip into the slot to lower the protective hood. Then, using a swab and rubbing alcohol, thoroughly clean the contacts, then let the cartridge dry. Incidentally, this is also a recommended procedure for your Operating System board and other RAM boards. You may also want to try some TV tuner cleaner in place of the rubbing alcohol.

We don't know of any problems with one cartridge leaving the machine in a state that prevents it from running another cartridge, especially since the power is cut off between cartridge changes. If any other readers are having similar problems, or have a cure, please write in.

Slow TI BASIC

In his review of *Robot Runner* for the TI-99/4A in COMPUTE!, January 1984, Tony Roberts stated that games written in BASIC on the TI are notoriously slow because the microprocessor can't interpret BASIC fast enough. I want to clear up any implication that the TMS9900 CPU in the 99/4A is at fault.

TI BASIC is indeed slow, due to the unusual architecture of the machine and the design of the BASIC interpreter. First of all, the RAM in which BASIC programs are stored is not CPU RAM. The 16K of RAM in the 99/4A is maintained by the TMS9918A video display processor (VDP). There are only 256 bytes of CPU RAM in the 99/4A console.

Every time the microprocessor accesses or RUNs a BASIC program, it must request the program from the VDP one byte at a time, one statement at a time. This causes a great increase in execution time, because the microprocessor must wait for the VDP. While the TMS9900 microprocessor is a word-oriented (16 bits) chip, the VDP works in bytes.

The second reason why TI BASIC is so slow is that the interpreter itself is not written in machine language. It is written in another high-level language known as Graphics Programming Language, or GPL. The GPL interpreter is also built into the 99/4A console. Thus, whenever a BASIC program is RUN, a *double interpretation* takes place. This is similar to writing a BASIC interpreter in BASIC for an IBM PC. It is really amazing that the TMS9900 can run BASIC as fast as it does, considering.

Chris Clark

Use Of COMPUTE! Programs

Concerning the "Readers' Feedback" of September 1983, you stated that the programs in COMPUTE! are not in the public domain, and that only people who own a specific issue of COMPUTE! can have access to the programs in that issue. My question is, what if a computer club takes out a

A SOFTWARE STAR IS BORN

MovieMaker™ brings the art of computer animation to your Atari Home Computer®. And turns you into the director. No programming is necessary, just imagination, flair and a desire to experiment. MovieMaker™ is a powerful, innovative tool that lets you devise the action, set the scene, create the actors and stage the action. The "Compose" mode lets you draw characters and background, with a special "Mirror" function for quick and easy duplication of shapes. The "Zoom" lets you add incredible detail for astounding realism. When you "Record," you film and edit up to 300 frames of continuous action, controlling speed, sequence, colors, camera angles and layers of sound. And the "Smooth" function eliminates all flicker. The results are so professional, you'll want to film one dream after another. And you can, without ever leaving your keyboard. That's the reason why MovieMaker™ is the best selling software from Creative Pastime™.

MovieMaker™ is a Creative Pastime™ from Reston Software™

A Prentice-Hall Company
11480 Sunset Hills Rd.
Reston, VA 22090
Available at your local bookstore and computer retailer, or call us at
(800) 336-0338

Coming soon for Apple,™ IBM-PC,™ Commodore 64™
For the Atari 800/1200® Computer
Atari is a registered trademark of Atari, Inc.

subscription to COMPUTE!? Would that club be allowed to place those programs in those issues in its library for all members? And what if a school or public library takes out a subscription? Could everyone who is allowed access to the library be allowed access to those programs in those issues?

Gary Lee Crowell

Sorry, the answer in each case is no. You can only use the programs in an issue of COMPUTE! *if you own a copy of that issue.*

VIC Video Typewriter

I have written a short program that transforms your VIC into a typewriter (without any annoying syntax errors). I use it to practice my typing after school. To disable the program, use the f1 key.

Vicky Cwiertnie

```
10 PRINTCHR$(8):PRINTCHR$(14)
20 POKE36879,26:PRINT"{CLR}"
30 PRINT"** VIDEO TYPEWRITER **"
40 GETA$:IFA$=""THEN40
50 IFA$="{F1}"THENEND
60 IFA$=CHR$(13)THENPOKE36878,15:POKE3687
   6,220:FORX=1TO50:NEXT:POKE36876,0
70 PRINTA$;:GOTO40
```

Atari Tape Verify

Here is a one-line program which verifies that an Atari tape file is recorded properly. The utility works whether you CSAVE, LIST, or PRINT (data) to the tape. It performs essentially the same as Michael J. Barkan's "Atari Verify" (COMPUTE!, August 1983), but is much shorter. This utility can be LISTed to tape and ENTERed from tape, but since it is so short, it is easy to enter it from the keyboard in direct mode (without the line number). Just use this line:

```
0 CLOSE #1:OPEN #1,4,0,"C:":FOR A=1 TO
  400:GET #1,A:NEXT A
```

After recording a file on tape and while the program or data is still in memory, enter and run this utility. Rewind the tape to the beginning of the file and push PLAY. The utility will read the entire file, one character at a time, to insure that the file is recorded properly. Operation will end with an error code. If you get this code, the file was read successfully, showing that it is good:

136 END OF FILE

If you get one of the following error codes, save the file again, since it could not be read by the computer:

138 DEVICE TIMEOUT
140 SERIAL BUS ERROR
143 DATA FRAME CHECKSUM ERROR

The same variable is used for loop control and to hold each character as it is read from tape. This way, the loop never ends and will check any length of file. This variable can be changed to one of those in your program, if desired, to avoid adding to the Variable Name Table of your program.

Douglas J. Wilder

TI Randomness Test

Richard Mansfield's article "Zones Of Unpredictability, Part 2" ("The Beginner's Page," COMPUTE!, December 1983) included a program called "Randomness Test." Since it wouldn't work on my TI-99/4A, I wrote a similar program. It takes several thousand cycles to get close to even distribution for each number, but it's fun to let it run.

Gaston Porterie

```
100 CALL CLEAR
110 PRINT "TEST OF THE RANDOM NUMBE
    R","FUNCTION ON THE TI-99"::::
    :::
120 PRINT "PLEASE WAIT..."
130 T=T+1
140 RANDOMIZE
150 X=INT(10*RND)+1
160 A(X)=A(X)+1
170 FOR I=1 TO 10
180 P(I)=INT(A(I)/T*100)
190 NEXT I
200 IF T/100<>INT(T/100)THEN 130
210 CALL CLEAR
220 PRINT "AFTER":T;"CYCLES":"OF RA
    NDOMIZATION"
230 PRINT
240 PRINT "RANDOM","%","NUMBERS","O
    CCURRENCE"
250 S=0
260 FOR I=1 TO 10
270 PRINT I,P(I)
280 S=S+P(I)
290 NEXT I
300 PRINT "","---"
310 PRINT "TOTAL",S;"%"
320 GOTO 130
```

Easy DATA Statements

Here is a one-liner that I have found very useful while programming many statements that are almost identical. Used in the direct mode it can yield a set of DATA statements that fill the screen. The program can just as easily use POKE, or REM statements, or any combination of these.

```
FOR X=100 TO 300 STEP 10:PRINT X "DATA":
  NEXT X
```

Chuck Cole

Constant 1541 Errors

Ever since I bought my 1541 disk drive, I have been getting the errors 23 READ ERROR and 27 READ ERROR. This not only happens on my

disks, but also on prepackaged disks. I have read what these errors mean in Appendix B of my disk users guide, but these descriptions don't tell me much.

Could you please give me more information on these errors, and tell me what I can do about them?

Jay Elmore

The fact that this occurs both on your own disks and on commercial disk programs strongly indicates a hardware problem. Ask the dealer from whom you purchased the drive for the address of the nearest service center and have the drive checked out.

Sprite Data Problems

I am a Commodore 64 owner and I have a question about sprites. I understand how to create a sprite and move it around the screen. I also know how to move more than one sprite, if the data for them is the same. My problem occurs when I have more than one set of data. I can't seem to get both sprites on the screen at the same time. The *Programmer's Reference Guide* doesn't have an example with two sets of data. I would appreciate it if you would help me out.

Seth Hausman

Jim Butterfield replies:
I can think of two possible problems with your sprites:

1. You may have forgotten to link each sprite to its drawing in memory. With normal memory mapping, sprite 0 needs to have its drawing number (usually 11, 13, 14, or 15) placed into memory address 2040, sprite 1 into 2041, and so on up to sprite 7 into address 2047. If you use drawing number 11, the drawing of the sprite should be in addresses 704–766 decimal; for number 13, addresses 832–894; for number 14, addresses 896–958; and for 15, 960–1022.

2. Many sprite register addresses control all eight sprites at the same time. To turn sprite 0 on, you would POKE 53269,1; to turn sprite 1 on, you would POKE 53269,2; to turn them both on, *you would add 1 and 2 and POKE 53269,3. The following table shows the bit values for each sprite:*

```
Sprite 0 – 1
       1 – 2
       2 – 4
       3 – 8
       4 – 16
       5 – 32
       6 – 64
       7 – 128
```

Thus, to turn on sprites 0, 2, and 4, we add 1+4+16 and POKE 53269,21.

Be sure that you keep the difference between a sprite number and a drawing number clear in your mind. Several sprites can use one drawing (or "definition"); or a single sprite can be switched from one drawing to another as it moves its arms, legs, tentacles, or whatever.

Using Atari Cartridge Memory

I have an Atari 800, and am currently writing a text-adventure game using the Assembler Editor cartridge. I hope to run the program without the cartridge when I'm finished. How can I use the 8K block of RAM used by cartridge (not to mention all those zero-page pointers that the cartridge uses)? Does it have to go to waste? I hope not, because I'll need all the memory I can get for this thing.

John Bushakra

No, the memory need not be wasted, but you cannot test the program with the Assembler Editor. Just define the memory you need, then assemble your program to disk. The object code will not go into memory, but will become an executable object file on the disk. The syntax is:

```
ASM,,#D:filename
```

You can then take all the cartridges out of your machine, boot DOS, then Load Binary File. If you make these the last two lines of your machine code

```
*=$2E0
.WOR START
```

where START is a label for the start address, your program will run automatically after it is RUN. Otherwise, you'll have to use Run At Address to start your program from DOS.

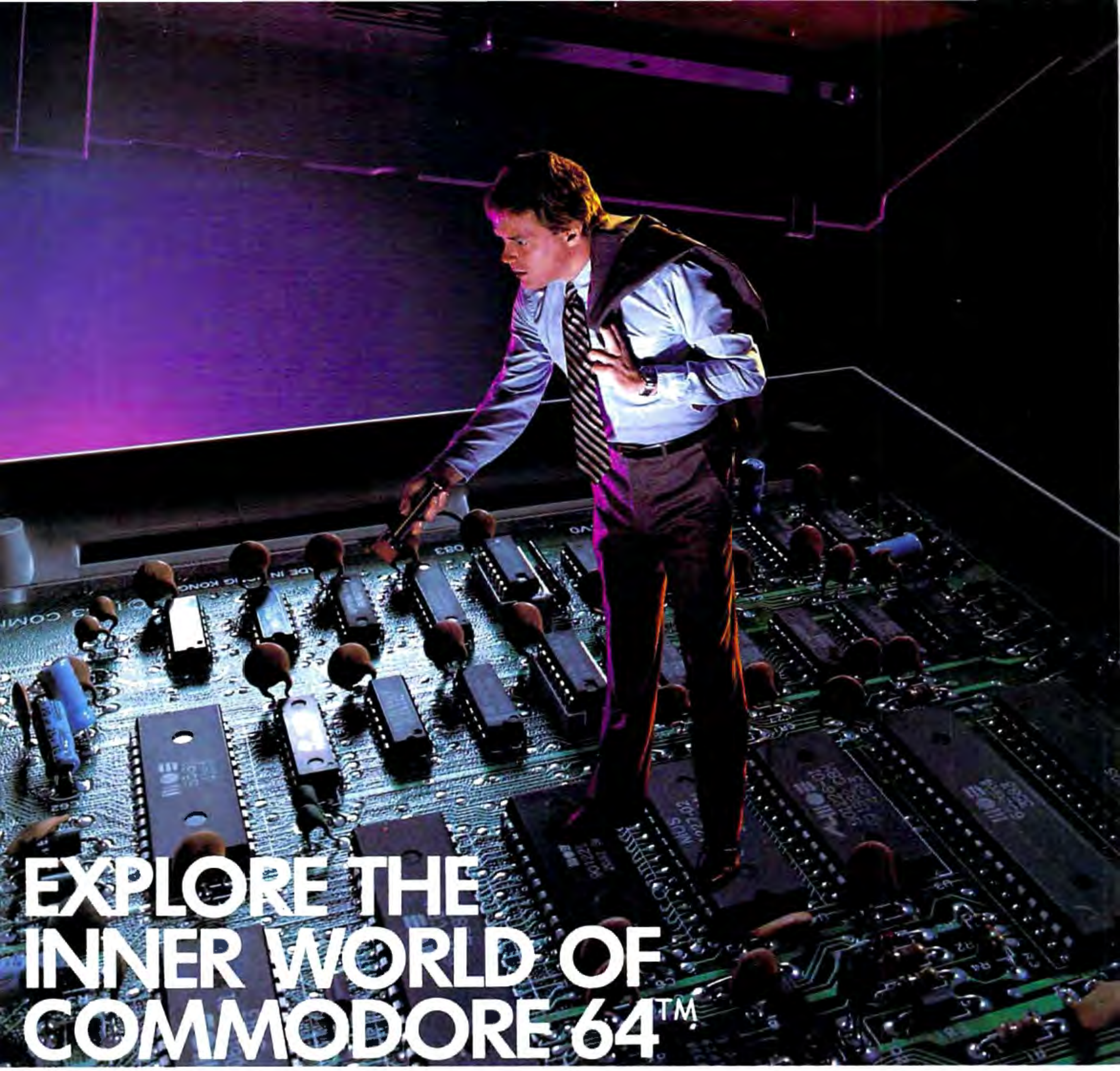
More Solutions For TI Cartridge Loading Problems

In the January 1984 "Readers' Feedback," I read a question about TI-99/4A cartridge loading problems. The problem was with lockup of the keyboard and broken screen display patterns after insertion of a program cartridge. The remedy given by COMPUTE! was to clean the contact strips of the program cartridge. I've found this to help, yet also discovered that this is not necessarily the complete solution. The cartridge connector extension that protrudes from the main circuit board may also be at fault. To remedy the problem means disassembling the computer, cleaning the contacts on both sides and both ends of the cartridge connector extension. This solved all of the problems I had encountered.

Richard Winslow

About four months after buying my TI, I had the same problem with loading the cartridges. I solved the problem by taking apart the computer and straightening the bracket which the cartridge plugs into. (It was bent.) Works perfectly now.

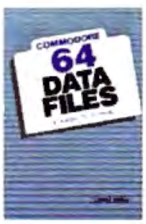
David L. Jones



Whether you're a beginner or an experienced user, Reston can expand the world of the Commodore 64™ for you.

COMMODORE 64™ COLOR GRAPHICS: A BEGINNER'S GUIDE, by Shaffer and Shaffer, explains how the Commodore 64 operates and teaches you how to read, understand and write simple basic programs for generating color graphics. Each topic includes a BASIC programs, line-by-line explanations, and illustrations of what the screen should look like.

COMMODORE 64™ DATA FILES, A BASIC TUTORIAL, by David Miller, is a step-by-step tutorial which takes the mystery and misery out of creating files. You'll learn how to manipulate and create your own files for home, hobby, business, educational, and investment purposes.



ADDING POWER TO YOUR COMMODORE 64™, by Steve Cates and Vahe Guzelimian, uses a first-of-its-kind utility approach to help you master more of the advanced computing power of your machine than you ever though possible. You'll get an inside look at the workings and advanced features, all in an easy-to-understand style.

MASTER MEMORY MAP: COMMODORE 64™, by Pavelko and Kelly, is a clearly written, friendly guide to all the Commodore 64™'s memory locations — places inside the computer which act in special ways. You'll learn lots of special uses, including how to make music or create special characters for video games.

You can find these guided tours of the Commodore 64™ at your local bookstore or computer store. Or order directly from Reston at (800) 336-0338.

Reston Computer Group
A Prentice-Hall Company
11480 Sunset Hills Road
Reston, Virginia 22090

COMMODORE 64 is a trademark of Commodore Electronics

I discovered that slightly lifting the back of an in-place cartridge seems to improve contact. So I cemented a small rectangle of soft black plastic about .1 inch thick onto the cartridge insertion area to lift each cartridge a little.

I also use a "Widgit" (Navarone Industries) that holds three command modules and prevents some wear and tear on contacts.

John K. Newell

VIC Video Revisited

I made some modifications to Jim Butterfield's program in "Visiting The VIC-20 Video, Part 4" (COMPUTE!, August 1983) that I think your readers will be interested in. Although the program is a little slow, the patterns that result are remarkable.

The program is short, but because of the loops, it runs for a while. To get some really interesting three-dimensional patterns, try inputs with a value of 1.02, 1.03, 1.04, etc. The input sets a step value for line 600. An input of 2 will give a gray field.

William B. Broome

```
100 POKE56,22:CLR
105 INPUT"{CLR}PATTERN #";C
110 POKE36869,222
120 POKE36866,144
130 POKE36867,32
200 FORJ=6144TO8191
210 POKEJ,0:NEXTJ
300 FORJ=0TO255
310 POKEJ+5632,J
320 NEXTJ
330 FORJ=37888TO38911
340 POKEJ,0:NEXTJ
600 FORJ=6100TO8800 STEPC
610 X=128
620 FORK=JTOJ+7
630 POKEK,PEEK(K)ORX
640 X=X/2
650 NEXTK,J
700 GOTO700
```

Try adding STEP C to line 620 as well, to get another interesting effect.

Another Kerosene Warning

A letter in your January 1984 issue questioned the use of kerosene heaters near a home computer. You compared the emissions of a kerosene heater to those of a gas stove, and suggested the use of an electrostatic air cleaner as a precaution.

The sulfur content of most kerosene fuels is high enough to create sulfur oxide levels that are technically in violation of EPA clean air regulations. These sulfur oxides can corrode exposed metals and cause problems with electrical contacts. Besides the corrosion of metals, the sulfur oxides can cause health problems.

Since the combustion products of kerosene are gases and not particulates, an electrostatic air cleaner will not help clean the air of sulfur oxides. An activated charcoal filter may help, but this is not a common appliance in most homes. Corrosion problems may not occur with other electronic appliances, such as televisions or radios, because the components are soldered or otherwise permanently fixed inside the appliance.

Those appliances which use exposed electrical contacts, such as game cartridges and computer keyboards, are most prone to corrosion by sulfur oxides. Readers should avoid the use of kerosene heaters in a home with a computer, electric typewriter, or silver tea set.

W. J. Tolonen

VIC Graphics And Super Expander

I'd like to share something with other readers who have VIC Super Expanders. We find that it interferes with some programs written for the unexpanded VIC, especially ones with custom characters. When I find such a program, or am told to "remove all expansion devices" for a certain program, I add the following as the first line in the program (or enter it in the direct mode before RUNning the program), and the program runs fine with the Super Expander left in. (In the case of two-part programs, insert the line in the second part.)

```
POKE51,30: POKE55,0: POKE52,30: POKE56,30:
  POKE646,6
```

This has worked on every program I've encountered so far, and what's nice about it is that you are left with the additional memory as well.

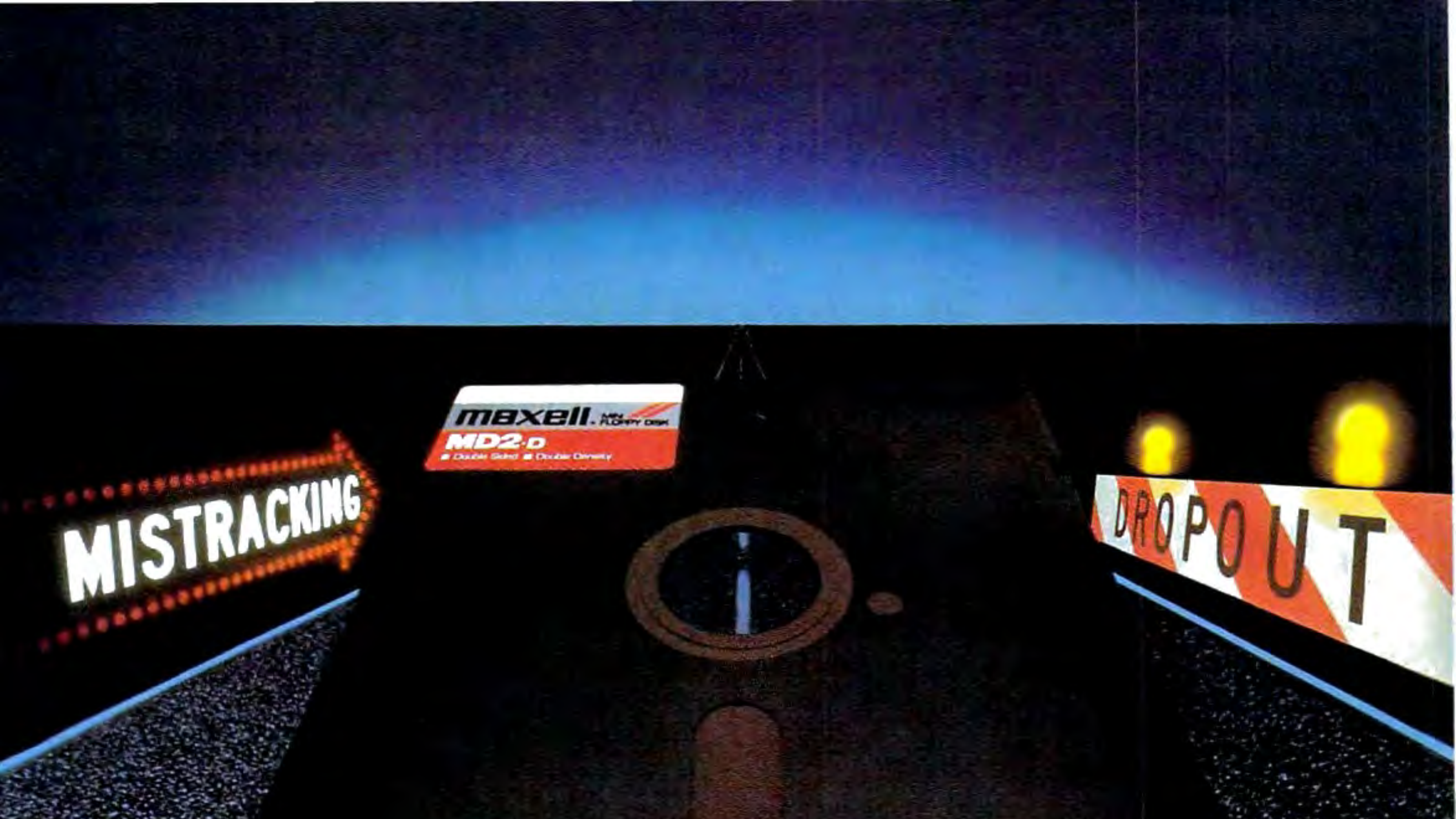
Robert M. Bleich

The *Whiz Kids* Were Right

I would like to point out an error in your response to David Smith's question about *Whiz Kids* (COMPUTE!, February 1984). You mention that the sprinkler system that they turned on should have shorted out the terminal they were using. You may have missed this, but it did short out. For a while they were successful in keeping the terminal dry by standing over it, using their jackets as umbrellas. But when the water did get to the terminal, it shorted out.

Karen Wilson

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 road to floppy success is paved with Gold Standards.

Maxell can speed your success in computing. Helping you avoid the traps that can block the way to the information you've stored. After all, our disk has outpaced every other in performance tests. And earned a lifetime warranty.

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

How good is Gold? Maxell's the disk that many floppy drive manufacturers trust to put new equipment through its final paces.

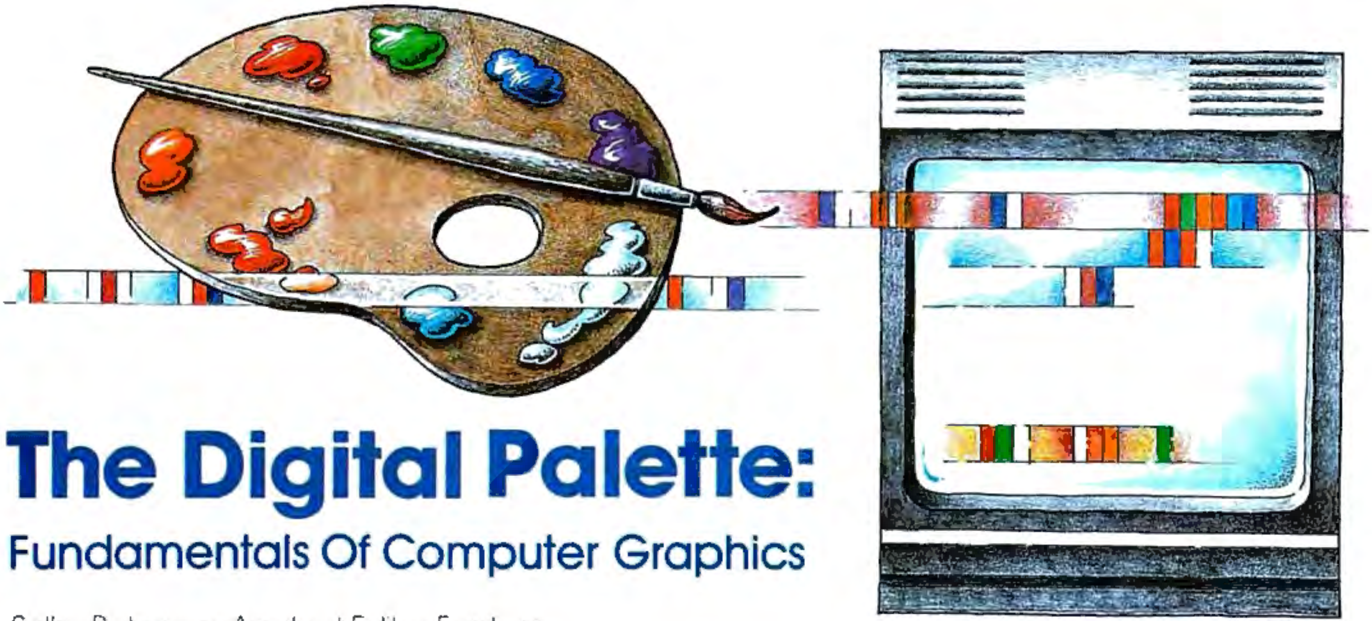
And the unique way we pack our oxide particles and bind them together means quality for the long run.

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



The Digital Palette:

Fundamentals Of Computer Graphics

Selby Bateman, Assistant Editor, Features

Your computer screen is a colorful gateway to the world of digitized graphics. But to appreciate all the video magic that's available, you first have to understand the basics. Here's a look at the pixel power behind your computer's video display.

"I remember standing back with everyone else, saying 'There's no way; I'm never going to touch this thing. It's not creative enough,' " says Kari Beims with a laugh. "I took a computer graphics class; it was full of people like me who are in the graphic arts field. And they were *petrified*."

The "thing" that petrified Beims and her classmates was, of course, a computer; a machine which, when used as a graphics tool, can appear so novel, so daunting, and ultimately so seductive that artists at first exposure may be simultaneously attracted and repelled.

Beims and many others have changed their views about the computer as a graphics machine. An artist at Maximus, Inc., of McLean, Virginia, Beims now uses and helps promote her company's new *Visualizer*, one of a new breed of graphics software packages for home computers that is as easy to use as it is useful.

"It's a lot of fun," Beims says. "*I* can do it, and I have no programming experience. I mean, I walked into here knowing nothing about computers."

An Undeniable Attraction

Graphic artists like Beims may be among the most perceptive analysts of how well a computer functions as a graphics tool. And although the polls haven't yet closed, Beims and a growing number of other artists are predicting a landslide in favor of the computer.

"It's getting—I hate to use this term because everybody says it—but it's getting *user friendly*; it truly is," Beims says. "People are starting to ask graphic artists what they need, and the results are beginning to show up in the computer software and the peripherals. And we need that."

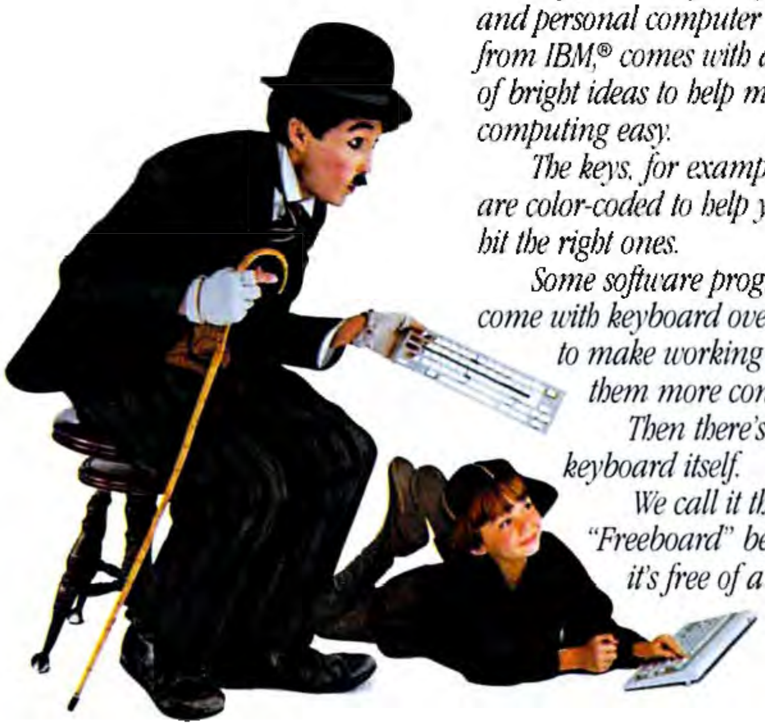
The glow of a computer screen is the first thing that attracts many people to computing. You don't have to know anything about pixels, raster scans, character sets, and RGB monitors to appreciate that something special, something new and powerful is possible when *you* can decide what appears on a TV screen.

If you want to create colorful, high-resolution graphics for games, business, or art, today's microcomputers have the capabilities to produce almost any image you wish. Peripherals such as light pens, graphics tablets, and touch screens will free you from the restrictions of the keyboard. And graphics software packages are becoming surprisingly easy and powerful tools for designing anything from pie charts to paintings.

You Don't Have To Be A Genius

A 20-year fascination with computer graphics led Joseph Deken to write the text last year for *Computer Images: State of the Art*, a full-color collection of computer artworks. Deken, an assistant professor of computer sciences and general business at the University of Texas at Austin, uses an Atari 800 and one of the IBM-compatibles at home. He believes you don't have to be a genius to understand and appreciate computer graphics.

"I use graphics to teach



PCjr, the new family and personal computer from IBM,® comes with a lot of bright ideas to help make computing easy.

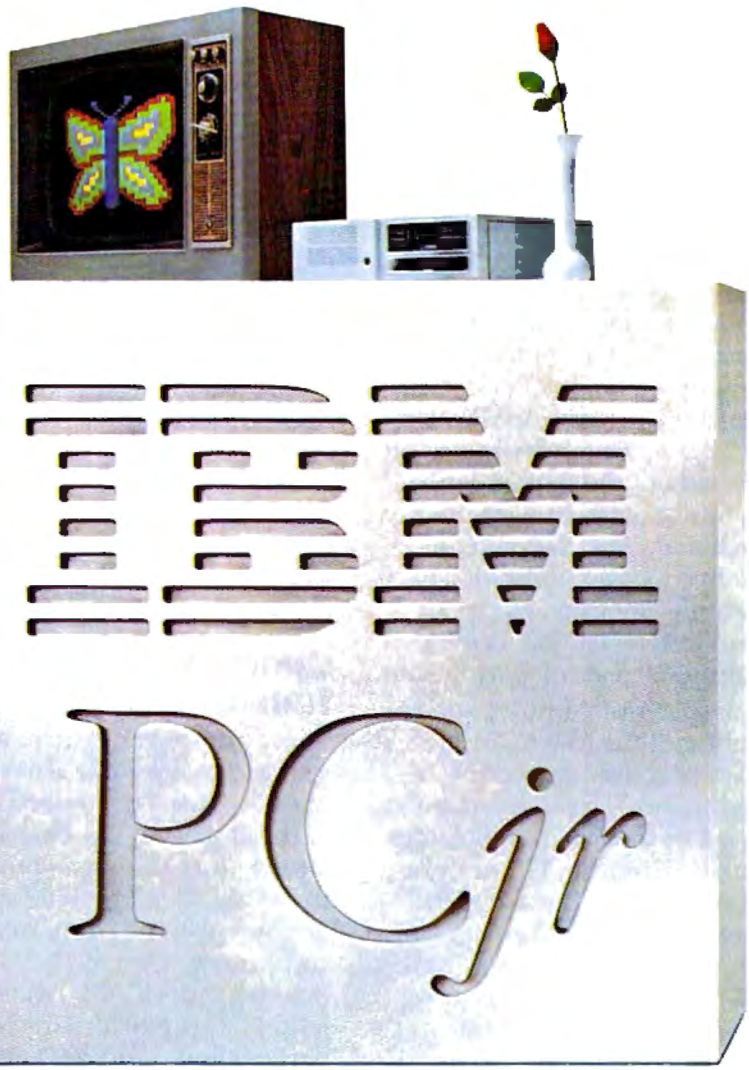
The keys, for example, are color-coded to help you hit the right ones.

Some software programs come with keyboard overlays to make working with them more convenient.

Then there's the keyboard itself.

We call it the IBM "Freeboard" because it's free of a...

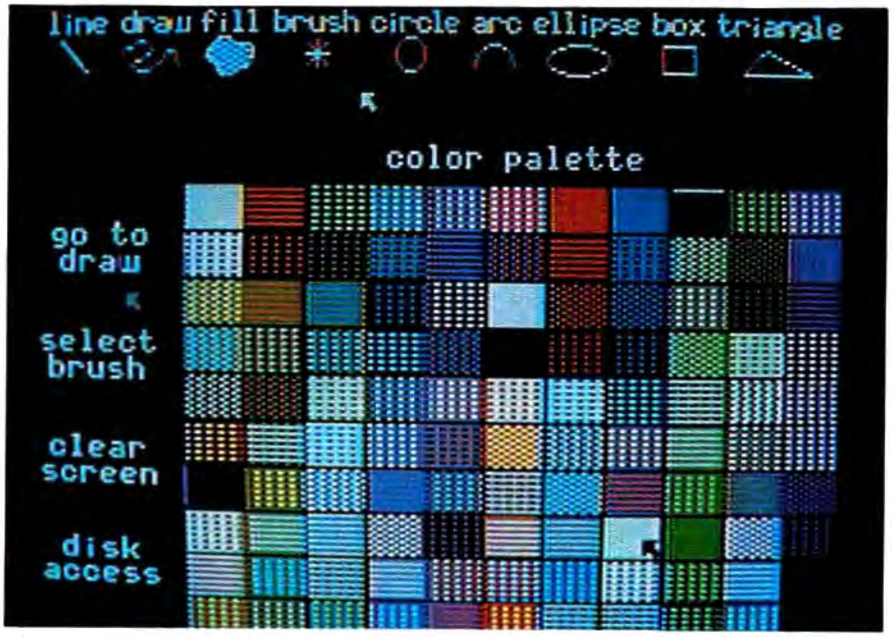
connecting cord.



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

about computers in introductory courses at the university," he says. "I'm always concerned with the stereotype that computers just have to do with mathematics. And I'm concerned with how to get students who aren't mathematically inclined interested in computers. Graphics winds up being a good vehicle for that."

An important first step in anyone's computer graphics education is the knowledge that capabilities costing thousands of dollars on mainframes just a few years ago are now available on inexpensive microcomputers. Learning the basics of computer graphics can now be inexpensive, simple, and often fascinating. Once you've picked up a few of the fundamentals, you'll probably appreciate your computer's graphics abilities even more.



The selection screen from Penguin Software's The Complete Graphics System.

Have Gun, Will Travel

Faster than the eye can follow—anywhere from 25 to 60 times a second—an electronic "gun" in your television or video monitor discharges a beam of electrons toward the screen. As the electrons hit a phosphor coating on the inside of the video display, the individual picture elements which make up your screen—called *pixels*—are lighted. This is a cathode-ray tube, the most common television and microcomputer screen display system.

Rapid-fire painting and repainting of the image on the TV screen is accomplished by the electron gun's repeated drawing of a set number of parallel lines (usually 525) from left to right and from top to bottom. This technique, known as a *raster scan*, occurs continuously and so rapidly that images appear to move smoothly across the screen.

The creation of characters and shapes on your computer screen is similar to the effects produced when thousands of college football fans use flip cards to spell messages of team support from the stands. The densely packed pixels flip either on or off, and from color to color, in response to directions from the computer. The more pixels, the higher the quality, or resolution, of the screen image.

A monochrome, or single-color, video display uses one electron gun. Red-green-blue (RGB) monitors use three electron guns, resulting in a higher resolution than the composite video you're used to seeing on a color television set.

Characters, Grids, And Turtles

There are several ways to create graphic images on a computer screen. First, you can make use of the alphanumeric characters—letters and numbers—built into the ROM (Read Only Memory) of your system. Many microcomputers, such as the Commodore 64 and VIC-20, have a parallel set of graphics characters—various lines, curves, and boxes—built into permanent memory. Using them as building blocks, you can combine characters into a variety of figures.

A more time-consuming, but flexible method for creating graphics is to manipulate the individual pixels. You tell the computer which pixels you want lighted and in what colors by communicating with it in a language, such as BASIC. In a personal computer which has a high resolution of, say, 320 x 200 pixels, there are 64,000 graphic points which you can potentially control. Locations in your computer's memory literally form a video map of what you can address on the screen.

(For more information on actually creating color graphics on your computer, see *COMPUTE!'s First Book of Atari Graphics, COMPUTE!'s First Book of Commodore 64 Sound and Graphics*, and other COMPUTE! books.)

Turtle graphics is a third way of producing images on your screen. Based on the Logo programming language, turtle graphics helps to teach programming and geometric principles. A small triangle on the screen—the imaginary turtle—can be directed to move about the screen, leaving an image in its wake. Intricate patterns can be achieved through this simple,

YOU'VE GOT TO DIG DEEP TO FIND A MORE EXCITING GAME THAN MINER 2049ER.™

Now, Reston brings Commodore VIC-20™ and Commodore 64™ owners one of the hottest home computer games ever created—**Miner 2049er.**™ Help the legendary Mountie, Bounty Bob™ chase the dastardly Yukon Yohan through an abandoned uranium mine—all ten levels of it. But you've got to avoid the deadly radioactive mutants. And watch out for pulverizers, explosives, slides and falls. And grab all the treasure you can along the way. **And** beat the clock. Any computer game you've played up to now has just been practice for Miner 2049er.™ It's fast and furious, a gold mine of action for the serious computer gamer. And Commodore VIC-20™ and Commodore 64™ owners can get it only from Reston.

A Creative Pastime™ from
Reston Software™

A Prentice-Hall Company
11480 Sunset Hills Rd. Reston, VA 22090
Available at your local computer retailer or call us at
(800) 336-0338.

Commodore VIC-20 and Commodore 64 are registered trademarks of Commodore Computer, Inc.

M[illegible] 2049er and Bounty Bob are registered trademarks of Big Five Software.

BOREDOM

BLAST OFF WITH BASIC GAMES FOR YOUR COMMODORE 64
David Busch

This is the first really fun approach to BASIC ever devised. The book containing time-tested, highly visual games makes it easy to learn BASIC programming on the Commodore 64 computer. It includes 25 games that take advantage of the Commodore's special features including graphics, sound and joysticks. The programs are even suitable for customizing for further fun.
1984/128pp/paper/ISBN 0-89303-333-2/$12.95

BLAST OFF WITH BASIC GAMES FOR YOUR VIC 20
David Busch

Twenty-five games make it simple to learn BASIC for your Vic 20. With twenty-five, time-tested, highly visual games, this book lets you take full advantage of the fun available on your Vic 20.
1984/128/paper 0-89303-334-0/$12.95

COMMODORE 64: AN INTRODUCTION TO BASIC PROGRAMMING AND APPLICATIONS
Larry Joel Goldstein & Fred Mosher

In this volume the master teacher has taken a hands-on approach to BASIC language and practical real-life applications. The book gives a complete and up-to-date account of what the Commodore 64 is and how it works. Page by page, the reader is introduced to DOS and BASIC. It shows how to program for personal and professional needs and includes programs for mailing lists, word processing, telephone lists, graphics, sound, and file handling—plus extensive coverage on structuring, planning, and debugging programs.
1984/288pp/paper/ISBN 0-89303-381-2/$14.95

ADVANCED BASIC PROGRAMMING FOR THE COMMODORE 64 AND OTHER COMMODORE COMPUTERS
Michael Richter

This is the next step for the Commodore user who has mastered the basics and wants to move on to more advanced software. In this book, the reader will learn how good programs are written, how to read and use them, how to know a good one when they see it, and how to gain knowledge through the experience of writing advanced software.
1984/204pp/paper/ISBN 0-89303-302-2/$14.95

COMMODORE 64: GETTING THE MOST FROM IT
Tim Onosko

Written for users with little or no previous programming experience, this handy guide is the only source with information on all three versions of the Commodore 64. It includes material on both the classroom model and the portable version. It offers a thorough and understandable introduction to the computer—including a solid approach to BASIC programming. It covers word processing, color graphics, and sound.
1983/320pp/paper/ISBN 0-89303-380-4/$14.95

BRAIN GAMES FOR KIDS AND ADULTS USING THE COMMODORE 64
John Stephenson

Between the front and back covers, this volume contains more than 40 exciting and challenging games. Designed to entertain and educate the reader, it teaches programming and the application of some very important mathematical concepts, without the pain. The games are short and precise and are targeted toward whose who learn best by doing. They can be played and enjoyed without an understanding of the mathematics involved. This is one volume you'll want to have for fun—for kids of all ages!
1984/160pp/paper/ISBN 0-89303-349-9/$12.95

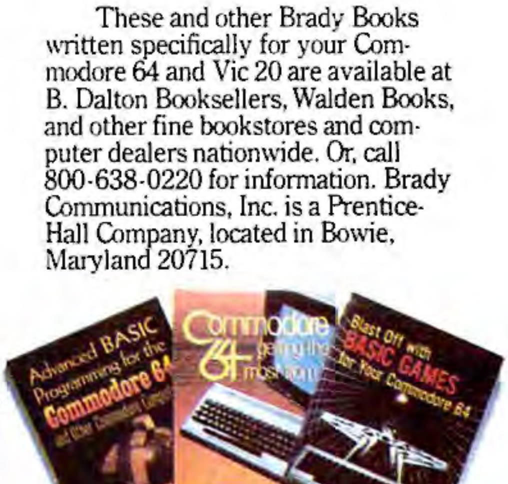
BRAIN GAMES FOR KIDS AND ADULTS USING THE VIC 20
John Stephenson

This is the volume especially for your Vic 20. Containing more than 40 challenging games, the book teaches programming and some basic mathematical concepts simply in the doing. Be sure and have this book to make your Vic 20 more fun.
1984/160pp/paper/ISBN 0-89303-347-2/$12.95

HANDBOOK OF BASIC FOR THE COMMODORE 64
Fred Mosher & David Schneider

For the beginner, here is the book to buy with your Commodore 64. It is simply organized by BASIC programming statements so while programming, the user can go directly to the information he needs without confusion or delay. This one-of-a-kind guide contains the same information as the BASIC reference manual supplied with your Commodore, rewritten especially for the new user. It presumes no knowledge of BASIC and explains the materials supplied in the manual.
1984/256pp/paper/ISBN 0-89303-505-X/$14.95

These and other Brady Books written specifically for your Commodore 64 and Vic 20 are available at B. Dalton Booksellers, Walden Books, and other fine bookstores and computer dealers nationwide. Or, call 800-638-0220 for information. Brady Communications, Inc. is a Prentice-Hall Company, located in Bowie, Maryland 20715.



BRADY

but subtly powerful graphics language system.

To Mimic More Closely

As microcomputers become more sophisticated graphics machines, a similar process is occurring with the devices used to draw and paint images on the computer screen. The graphics software now available relies more and more on such peripheral devices as joysticks, light pens, graphics tablets, touch screens, and mice (defined below).

These peripherals fall into three basic price ranges, according to research conducted by Koala Technologies, producer of the KoalaPad Touch Tablet and the Gibson Light Pen. You can find game paddles and most joysticks from $15 to $30; trackballs, mice, graphics tablets, and better light pens from $30 to $400; and precision tablets, called digitizers, from $400 on up. (See "Light Pens And Graphics Tablets" and "A Graphics Glossary," both in this issue.)

All of these devices attempt to mimic more closely the actual procedure of drawing or painting on a flat surface, as opposed to the more indirect, less satisfactory process of typing in graphics commands on a keyboard. Often, a joystick, light pen, or graphics tablet may be used in conjunction with keyboard commands to produce lines, boxes, circles, rays, points, and various fill patterns.

Other commands allow you to transpose and merge images, lock onto and move parts of a picture, magnify sections of an image for more detailed work, save pictures for later use, and select or change colors.

For Atari And Commodore

Kari Beims says that a mouse—a hand-controlled device which rolls on a flat surface to move the screen cursor—is the most successful device for creating graphics. "It's closer to the kind of tools you're used to working with. With a joystick or a light pen, it's a little different. The mouse lets you work flat-on; you've got more control. And it's closer to the actual production work you've done before," she says.

Software companies are

A Graphics Glossary

ASCII: (Pronounced *askey*) American Standard Code for Information Interchange. A standard code used in microcomputers to represent alphanumeric information (letters, numbers, and symbols). The capital letter A, for example, is represented in ASCII code by the number 65.

bitmap graphics: A high-resolution graphics plotting technique by which pixels (picture elements) on a computer screen are turned on and off.

CAD: Computer-aided design. The use of computer graphics to help in design development and modification, often eliminating the need to create costly or dangerous prototypes. CAD is usually associated with CAM, or computer-aided manufacturing.

character graphics: The text characters that appear on your computer screen when it is turned on, including letters, numbers, symbols, and punctuation marks.

CRT: Cathode-ray tube. A video display terminal, such as a television or video monitor, which uses a beam of electron particles to draw images on a screen's phosphor coating. The electron beam can write on the screen with a single beam or, more commonly, in a series of parallel lines to form an image. (See **raster scan** and **vector scan.**)

electron gun: The mechanism within a CRT which shoots a narrow beam of electrons at the screen, creating images. The beam is constantly redrawing the screen at speeds usually ranging from 25 to 60 times a second. Monochrome (single-color) displays use one electron gun. Red-green-blue (RGB) displays use three separate guns (one for each color), and have a higher resolution than the composite color displays found on television sets.

fractals: Geometric patterns which, when repeated, can create new patterns seemingly unrelated to the original forms. Especially useful in computer generation of detailed maps and duplication of the intricacies of many natural objects.

graphics set: The complete set of graphics characters that a computer can display.

graphics tablet: A pad, usually square or rectangular, on which the X and Y coordinates of the computer screen can be plotted by the use of a stylus, or, on some models, your finger, allowing you to create graphic images.

icons: Graphic symbols, most often used as visual representations of computer software options and procedures. For example, a paintbrush icon would represent the painting option in a graphics software package. Similarly, a trash can icon might indicate a delete option.

image processing: Computer enhancement and alteration of photographs and other graphic images by digitizing a picture into pixels, each of which is then measured for light and color intensity. The pixels can subsequently be manipulated to change the image.

light pen: A stylus which emits low-level electrical pulses and, when pulled across a computer screen, creates an image. Most often used in creating graphics and in interacting with software menu options.

Logo: An easy-to-use, graphics-oriented programming language originally developed to help children learn programming and the concepts of geometry.

mouse: A small hand-controlled device which rolls on a flat surface, allowing you to control the screen cursor, draw graphic images, and select from menu options.

phosphor coating: The coating inside a CRT that glows when struck by a beam of electrons. A computer activates an electron gun to draw and redraw graphic images at high speed on the coating, thus producing graphics that appear to move smoothly across the screen.

pixel: Picture element. The smallest graphic point addressable by a computer. Pixels are turned on or off to form the characters and graphic images on a computer screen.

raster scan: A video picture drawn by an electron gun which sweeps horizontally across the screen in a series of parallel lines at a high rate of speed. The most common method by which a microcomputer system displays a screen image.

resolution: The clarity of a video image based on the number of pixels available on the display screen. The more pixels there are, the higher the resolution and the more detailed the screen image.

RGB monitor: Red-green-blue monitor. A high-resolution color monitor which uses three electron guns to produce very clear and crisp images. By comparison, a color television would normally have a composite color video system in which the three primary colors would be blended, producing a lower quality video display.

simulation: Computer graphics created to model reality in appearance and usually in performance. Numerous airline companies, for example, use computer-generated flight simulations to help train their personnel.

sprite graphics: Sometimes called movable object blocks (MOBs), sprites are programmable graphics characters that can move around the screen independent of the primary screen image, the background.

touch screen: A video screen or plastic screen overlay which allows you to draw, write, and make menu selections from the screen at the touch of a finger or stylus.

turtle graphics: Closely associated with the Logo programming language, the turtle graphics system is most often used in an educational context, especially in teaching children about computer programming and geometric shapes. A triangular screen cursor (the turtle) moves across the face of a display monitor in response to directions entered into a computer, allowing the user to program a wide variety of geometric graphic images.

vector scan: A video picture drawn by the focused beam of an electron gun, much like a pencil's movement across a piece of paper. This produces a slower, but higher quality, video image than the parallel-line technique of a raster scan.

video chip: A tiny microprocessor on a silicon chip which handles the video data within a computer, assisting the central processing unit (CPU) by managing the screen image.

exploring all types of input devices in order to give users the right mix of creative flexibility and control. The newer packages offer more options, and a greater number of practical applications as well.

One such product is the *Visualizer* graphics animation package introduced by Maximus, Inc., at Softcon, the international conference and trade fair of the software industry, held recently in New Orleans. Available now for the Atari computer with at least 48K and disk drive, the $49.95 package should be ready for the Commodore 64 in May.

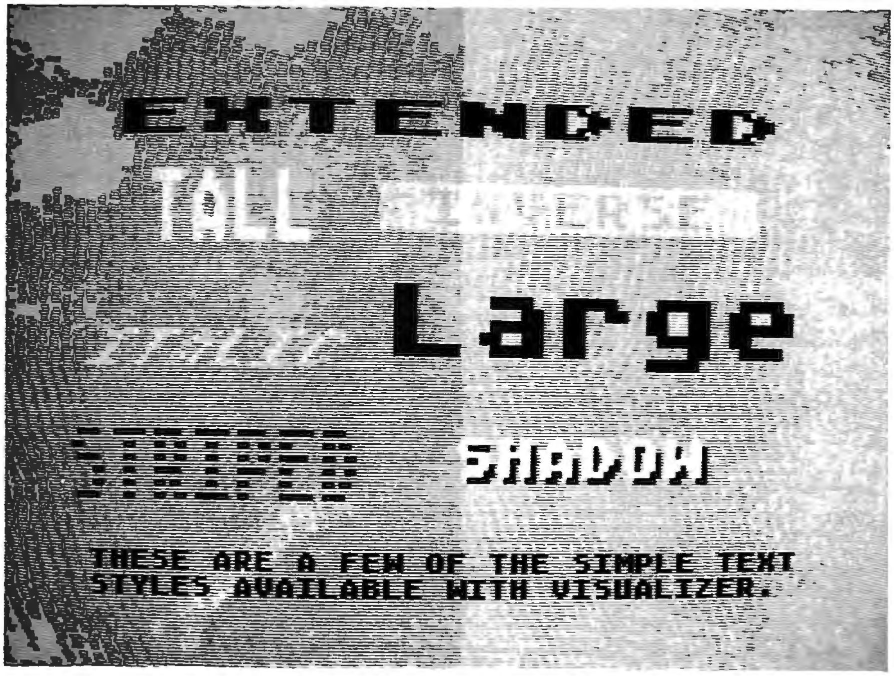
"The goal with *Visualizer* was to create a graphics program completely for the nonprogrammer, which would be useful besides just being a creative outlet," says Beims. "The *Visualizer* gives you the option, in addition to creating slides, of adding animation effects and putting together a slide show using up to 26 screens.

Synchronizing Slides And Sound

"You can synchronize them with an audio track so that you've got a customized audiovisual presentation. You can move the slides ahead manually, or use a timer through the computer," she says. "In addition to being a graphics program, it's useful. You can use it for teaching, training, business and sales presentations, retail advertising, and instructional lessons for the kids at home."

The package uses automatic drawing functions for circles, ovals, boxes, borders, and diagonals, and has 18 different text style options that can be used with graphics. The slides you create can be printed in black and white with an Epson MX (with Graftrax), FX, or RX series printer, or with a C. Itoh (NEC, Prowriter, or other) printer. A joystick is used for drawing, with a variety of colors and brush sizes available.

Two more recent graphics software products which combine usefulness with ease of use are *The Graphics Magician* (on disk for Apple, Atari, and Commodore at $59.95 and for IBM by the end of 1984) and *The Complete Graphics System* (on disk for Apple at $79.95), both from Penguin Software.



The Visualizer *by Maximus, Inc., allows a variety of text faces to appear on the same screen with animation graphics.*

"With *The Complete Graphics System*, I basically wanted to set up an all-in-one tool for people to use to create computer graphics," says Mark Pelczarski, founder and president of Penguin Software.

Hundreds Of Colors And 3-D Too

The package is compatible with most input devices, and allows three-dimensional line drawings to be reproduced on a variety of plotters. There are over 100 available colors and 96 brushes. Portions of any individual screen may be magnified from two to eight times for easier manipulation of pixels. And text may be added with graphics in a variety of ways.

Penguin also sells several programs which can be used in conjunction with *The Complete Graphics System. Additional Type Sets* ($19.95) provides 50 extra typefaces and character sets. *Map Pack* ($19.95) includes outline maps of all 50 states, the continents, the U.S., and Canadian provinces. *Transitions* ($49.95) is a presentation system which will let you organize picture disks and turn them into slide shows. More than 35 different screen wipes are available. That is, you may clear the screen from top, bottom, left, or right sides; use geometric-pattern clearing techniques and windshield wiper effects, among others. Finally, *Paper Graphics* ($49.95) is a utility that will let you print any high-resolution graphics screen to your printer.

"*The Graphics Magician* software is actually two different sets of programs. One is a drawing program geared toward people who are going to use it in [creating] other software—like educational software," says Pelczarski. "There's a huge amount of educational software out there that's been done using *The Graphics Magician*."

The Graphics Magician uses machine language animation routines with the same techniques that are used on arcade games. Up to 32 independent objects can be assembled in the animating process. The package also includes a high-resolution picture/object builder, which lets you store hundreds of color pictures on a single disk. More than 100 colors are available for use as well.

Screens A La Mode

Almost all of the top-selling microcomputers have extensive graphics capabilities, but you'll want to spend some time learning your own machine's features.

The Apple IIe, for example, has a high-resolution mode with six colors and 280 pixels horizontally by 192 pixels vertically. There is also a 40 x 48-pixel low-resolution mode with 16 colors. Apple's new Macintosh, which uses a monochrome display, has a whopping 512 x 342-pixel resolution. Obviously, very fine graphic detail is possible with this many pixels.

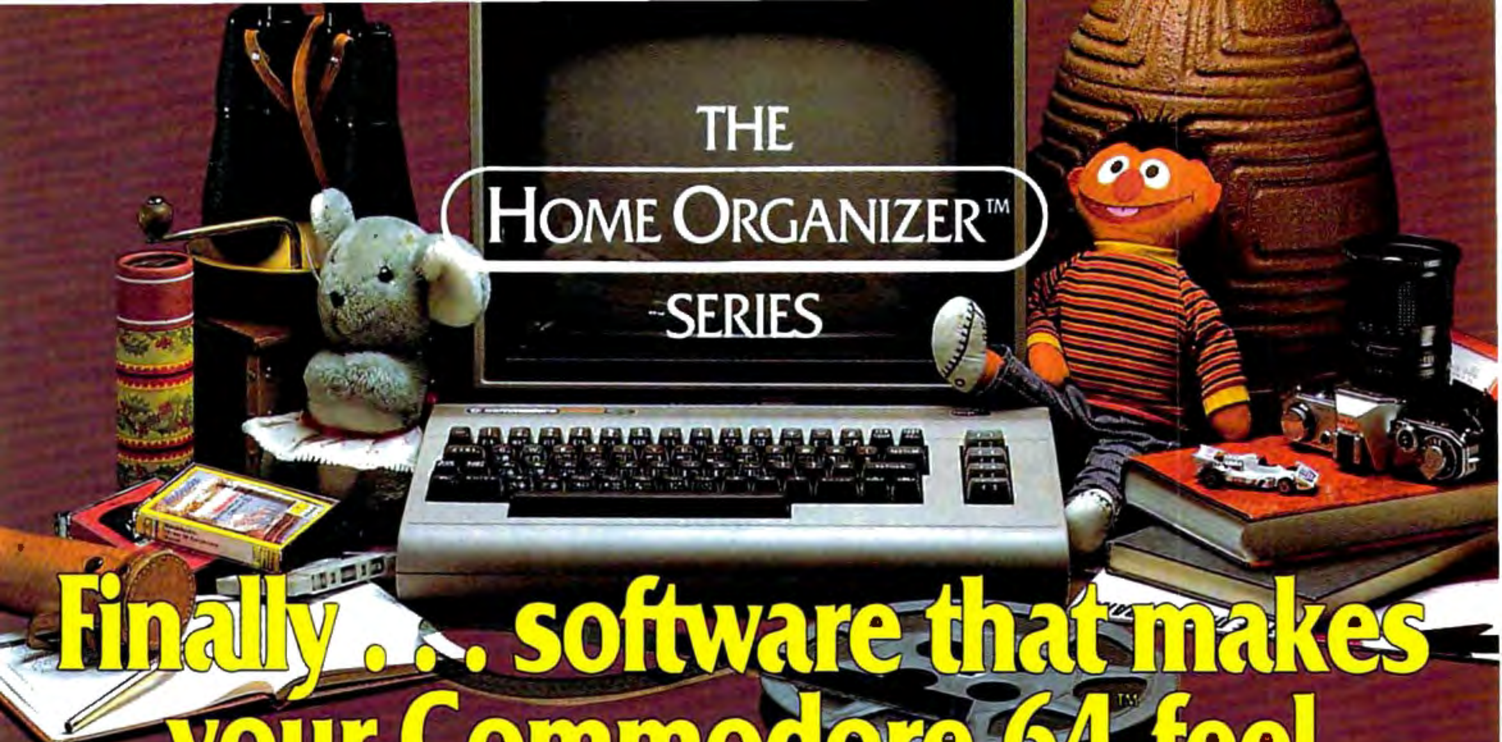
The Commodore 64 has 16 colors, several modes—including a 320 x 200 graphics mode—and eight independently programmable *sprites* (24 x 21-pixel movable screen objects), which allow you more opportunities to create animation. The VIC-20 also has 16 colors and a graphics resolution of 176 x 184.

The Atari 600XL and 800XL each have 320 x 192 graphics resolutions, as well as 256 colors (16 colors with 16 luminance levels for each color). But the Ataris also have 11 different graphics modes, or varying combinations of colors and pixel densities, which extend its graphics capabilities.

PC Pixels

IBM's PC and PCjr each have the same 320 x 200-pixel, four-color, high-resolution graphics mode, as well as the same 640 x 200, two-color mode. But the PCjr also has three other graphics modes that the PC doesn't: a 160 x 200, 16-color, medium-resolution mode; a 320 x 200, 16-color, high-resolution mode; and a 640 x 200, four-color, high-resolution mode.

Professional quality computer graphics systems may start



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

A scene from the animated graphic adventure, Ring Quest, *which was created with* The Graphics Magician *by Penguin.*

at about a 600 x 500-pixel resolution and go up from there. Color choices and luminance levels can go into the millions. Three-dimensional perspectives and complex simulations of actual processes, such as flying an airplane, are among the complex—and costly—features of some mainframe and mini computers.

A Child's Garden Of Graphics

Two programs by Scarborough Systems, Inc., make use of the computer's ability to create dynamic graphic images in a manner easy enough for a child to accomplish.

Picturewriter, by Dr. George Brackett, is an educational drawing program for children from 4 to 14 years of age, which Scarborough markets for the Apple computers at $39.95 suggested retail price. Its origins, says Brackett, are in work he did teaching children about the Logo programming language. When he asked a little girl one day what she wanted to draw, she suggested a rainbow—not the easiest of images to produce via Logo.

"So I began to think about what kind of program I would like to have that would make it easy for children to draw a rainbow," he says. "And it was pretty clear that it had to have a pointing device, like a joystick, rather than a keyboard. I also felt it had to have fairly extensive editing capabilities."

As children use *Picturewriter*, they can learn about spatial and color relationships, the development of geometric patterns, and the basics of computer programming at the same time that they're creating colorful pictures. A selection of preprogrammed works allow the child to alter the designs and colors as well.

Patterns For Apple, Commodore, And IBM

Another program, which Scarborough demonstrated at the recent Softcon show, is *Patternmaker*, a drawing and pattern-creating program for children six years and older and for adults. The package is scheduled to be available in May for Apple computers and by August for Commodore 64 and IBM machines, at a suggested price of $39.95.

"Its educational value is that it gets children comfortable with symmetry, rotations, transformations, inversions, and so on," says Scarborough President Francis Pandolfi. "It makes it easy for them to use those concepts to make beautiful patterns. Symmetry is a very important concept in many areas of science, not to mention art. And the program's manual brings the child through all areas of art in which symmetry has been important."

As you'll quickly find out when studying what's available for microcomputer graphics, the products are coming fast and furious. Softron, Inc., for example, makes a *Home Decorator* program ($34.95 for Commodore 64; other versions planned soon) that teaches about colors, furniture layout, and decorating theory for your home and office. The package even lets you select carpeting, paint walls, and move furniture.

The Age Of Discovery

Other programs, like Access Software's *Spritemaster* ($34.95 for Commodore 64) and Avant-Garde's *StarSprite* series (for Apple computers), show you how to produce sprites for use in multicolor animation.

There are literally hundreds of other graphics programs currently available. And there are numerous books and magazines which will teach you how to create your own colorful graphics.

"I think more and more people are discovering the graphics capabilities of computers," says Mark Pelczarski. "In the last couple of years, we've learned how to make software more easily understood. And with computers like the Commodore and the Atari really hitting the mass market, a lot of people who never would have dreamed of having a computer five years ago are learning about all the capabilities." ©

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

A scientifically proven way to develop an awesome memory.

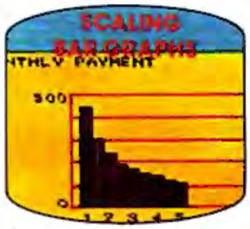
You are trapped in a five-story, 125-room structure made entirely of ice. Find the exit before you freeze!

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

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

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

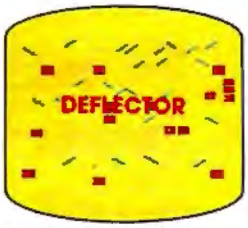
Computerize car maintenance to improve auto performance, economy and resale value.



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

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

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



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

A fun way to dramatically increase typing speed and accuracy.

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

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

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

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

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

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

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

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

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

CALL TOLL-FREE
1-800-334-0868

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

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

SUBSCRIBE NOW AND SAVE UP TO 40% ON COMPUTE!

Yes! Start my subscription to COMPUTE! for:

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

☐ Payment enclosed ☐ Bill me

Charge my ☐ Visa ☐ MasterCard ☐ Am.Ex.

Account No. ____________ / ____ EXP DATE

Return the coupon or postpaid card today!

Name ____________

Address ____________

City ____________ State ______ Zip ______

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

Light Pens And Graphics Tablets:

New Ways To Communicate With Your Computer

Kathy Yakal, Editorial Assistant

You don't have to be an artist to use them. Or a three-year-old. Or a professional architect, engineer, or fabric designer. Light pens and graphics tablets, along with the software that drives them, are making computing easier for young children and adding new dimensions in graphics for everyone.

Whether or not we admit it, we've probably all responded to our computer's SYNTAX ERROR message by typing "Syntax error? I typed it right!"

We may have jabbed a finger at the monitor to show our word processing program precisely which block of text we wanted moved and where we wanted it placed. Or maybe drawn a picture of a spaceship and held it up to the screen to illustrate exactly what we wanted displayed there after typing in a machine language game for the better part of a weekend.

The computer never seems to understand.

Interacting with a microcomputer can sometimes prove exasperating. You still have to talk to a computer in a language it understands. They don't yet respond to written or spoken English.

Animation Station, a graphics tablet designed by Suncom, offers a number of colors and textures for drawings like this.

ONE TOUGH SPELLER.

Time was, Billy would do almost anything to duck his spelling homework.

But since Dad brought home *Spellicopter*™ by DesignWare,™ Billy has become a spelling ace. As well as an ace chopper pilot.

Each week, in addition to the words already in the game, Billy types new spelling words and new sentences into the game. Then he takes command of his chopper and flies a mission through crowded skies and mountainous terrain to recover the words, letter by letter. And always in the right order.

So by the time Billy gets back to the base, he's one proud pilot. And one tough speller.

DESIGNWARE MAKES LEARNING COME ALIVE.

All DesignWare spelling programs combine computer game fun with sound educational principles to help improve your youngster's spelling skills. That's why *Spellicopter,* for example, is consistently on the nation's best seller lists. You'll also want to keep an eye out for new math and science games.

All DesignWare programs run on these computers with disk drive: Apple,® Atari,® Commodore 64,™ IBM PC and IBM PC Jr. See your local software retailer or call DesignWare at (800) 572-7767 (in California 415-546-1866) for our free software catalog. You'll be delighted with the way your kids will learn with DesignWare.

Commodore 64 is a trademark of Commodore Electronics Ltd. IBM PC and IBM PC Jr. are registered trademarks of International Business Machines, Inc. Apple is a registered trademark of Apple Computer, Inc.

DesignWare™

LEARNING COMES ALIVE.

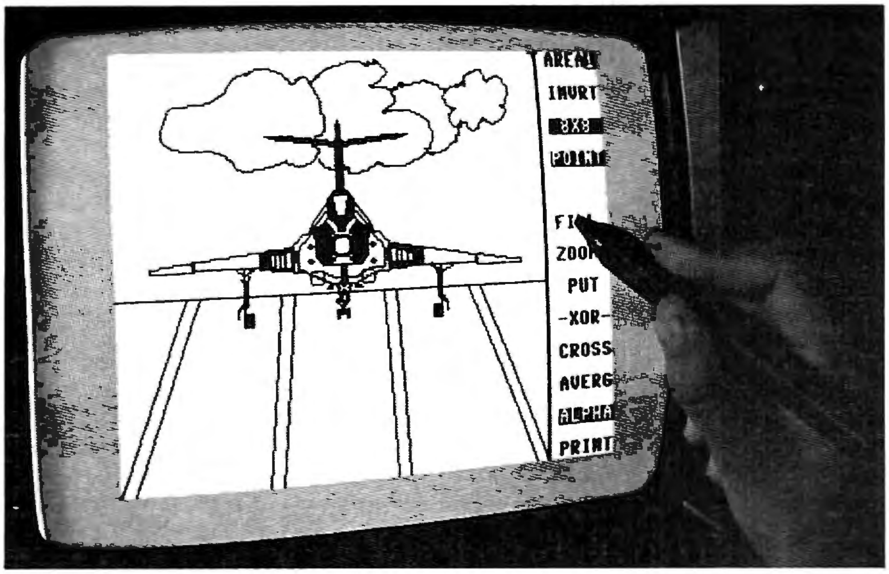

100

A Pointer To The Future

But graphics tablets and light pens bring us a step closer to easy communication by allowing information entry to bypass the keyboard. Like mice and joysticks and the keyboard itself, light pens and graphics tablets are input devices, peripherals through which you interact with your computer.

These pens and tablets do basically two things: draw and point. You can use them to select user options in menu-driven software and to create graphics.

A light pen is like a magic wand. It resembles a regular pen in size and shape, and has a cord that plugs into the computer. When you point it at the screen and activate it (either by pressing a switch on the pen itself or a key on the keyboard), it responds to whatever software you're running.



This picture was created by using the draw and fill features of Flexidraw (Inkwell Systems).

Touch-Sensitive Input

A graphics tablet looks a little like an Etch-A-Sketch, though each manufacturer's is a little different. Instead of pointing at the computer screen, you touch the surface of the tablet with your finger or a stylus.

Though light pens and graphics tablets are fairly new on the home computer scene, they've been used as design tools with larger computers for years. The technology is not brand-new.

Where's The Software?

The Edumate Light Pen, from Futurehouse, Inc., comes with introductory software that demonstrates the pen's features. If that's where it ended, the skeptics' claims that these tools are just gimmicks might be true.

But Futurehouse, along with other light pen manufacturers and software publishers, is beginning to design software that can be used with a light pen. "The potential applications are enormous," says Byrne Elliot, president of Inkwell Systems, another pen manufacturer.

"Not just being able to point at a menu option you might want. They'd be great with even things like spreadsheets and word processors. Instead of learning a lot of control commands that can be very frustrating, you can move text and figures around quickly and easily."

Doing What Comes Naturally

If you've ever tried using spray paint to letter a sign or illustrate a big banner, you have an idea of what drawing with a light pen feels like. The stream of color sometimes comes out faster than you can control it.

Drawing on a graphics tablet is not quite so novel a technique to master. It's like drawing on a piece of paper with a pencil, or drawing pictures on a steamy windowpane with your finger.

Using these pens or tablets is, however, intuitive, to a degree. "Among the first skills that everyone learns when they're young is drawing or writing," says Howard Leventhal, president of Suncom, manufacturers of Animation Station. "There are no other input devices that let someone do that in such a friendly way."

This may be why these new input devices are being so highly touted as educational tools. "There's a strong motivation for people to buy for educational purposes," says Leventhal.

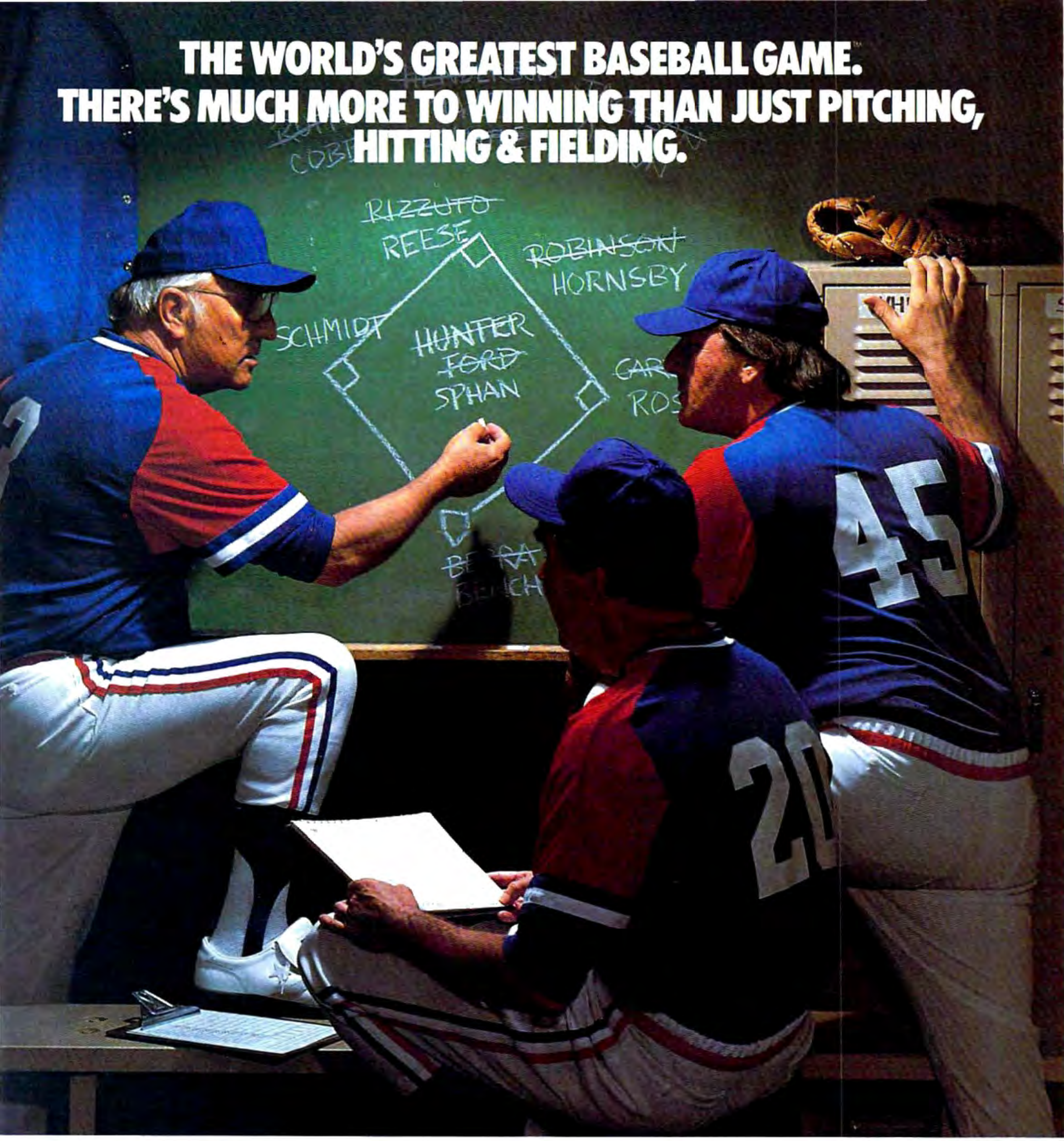
Like Being A Kid Again

It's not hard to see why graphics tablets and light pens can open the world of computing for children. These peripherals don't require the hand-eye coordination or knowledge of letters and numbers that a keyboard does.

"Light pens are superb for education. They really expedite the learning process," says Elliot. "They're a good way to get around the intimidation of the keyboard. To respond to software using a keyboard, you have to type in a series of letters and numbers, then return. The light pen is generally 10–15 times faster than that."

Bob Ranson, president of Chalkboard, agrees. "Graphics tablets allow the preliterate child to use a computer without having to deal with a keyboard," he says. "There are lots of two- and

THE WORLD'S GREATEST BASEBALL GAME.™ THERE'S MUCH MORE TO WINNING THAN JUST PITCHING, HITTING & FIELDING.

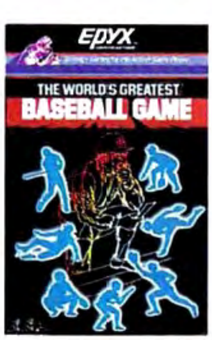


Real baseball is more than just hitting, pitching and fielding. It's also your favorite major league teams, the great stars of today and the Allstars of yesteryear. It's statistics and coaching, and it's managing your own game strategy. With the World's Greatest Baseball Game, you have it all. Pick your major league line-up using the actual player and team stats. Then watch the action unfold against an opponent or the computer. Two modes let you choose between managing and controlling your team or managing only. The World's Greatest Baseball Game—everything you could ever want except the hot dogs and peanuts.

One or two players; joystick controlled.

EPYX™ COMPUTER SOFTWARE

Strategy Games for the Action-Game Player



The Inside Story:

How Graphics Tablets And Light Pens Work

Ottis R. Cowper, Technical Editor

Many programmers find graphics tablets and light pens among the most mysterious of peripherals, but the principles of both are really fairly simple.

The graphics tablet is similar in operation to the more familiar game controller paddles. A paddle consists of a *variable resistor*, a device which can vary the amount of electric current passing through it. For example, rotating the paddle all the way clockwise would allow full current to flow; turning it completely counterclockwise would cut off all current. A computer which accepts paddles must have circuitry which can read the varying current and provide a numeric reading which is proportional to the current, and hence to the position of the paddle knob. This is usually zero when the paddle is rotated all the way in one direction, 255 when the paddle is turned completely in the opposite direction. Joysticks for the Apple, Color Computer, and IBM function in a similar manner, with one resistor on the horizontal axis and another on the vertical.

Graphics Tablet Positions

In a graphics tablet, thin sheets of a special film are used in place of the variable resistors. When you press down on the film, a current flows, with the amount of resistance depending on where on the film you press. For example, if the film is set up to register horizontally from left to right, pressing on the left edge is equivalent to turning the paddle for minimum resistance, while pressing on the right edge is like turning the paddle for maximum resistance.

The working area of the tablet consists of two sheets of this film, one arranged to register horizontally and one arranged for vertical measurements. The same circuitry used to read paddles (or Apple and IBM joysticks) can be used to read the tablet—the computer interprets each of the sheets as a paddle. What would normally be one paddle reading is the horizontal position of the point on the pad being pressed, while the other reading gives the vertical location.

One significance of graphics tablet design is that you should be able to substitute paddles (or the joystick for Apple or IBM) in programs which call for the tablet. Conversely, you might experiment with using the tablet in programs which call for paddles, although the tablet isn't likely to replace paddles for playing *Pong* or *Breakout*.

Holes Of Light

To understand how a light pen works, you must first understand how screen images are created. The chief element of any video display device, television or monitor, is a cathode ray tube (CRT), a sealed glass funnel with an electron "gun" in the narrow end and a specially coated screen across the wide end. The gun shoots electrons at the screen, leaving tiny bullet holes of light where the electrons strike the dark screen.

The shots are not random; they are carefully targeted by powerful electromagnets, the big coils of wire around the throat of the CRT if you've ever looked inside a TV or monitor. Starting at the upper left corner of the screen, the gun is swept across at a constant speed. Shots are fired at the spots that need to be lit up to form part of an image. When the gun has swept all the way across to the right edge, firing is halted while it is aimed at the left edge again, slightly lower than on the first pass. Thus, the spray of

PUZZLEPANIC.™
KEN USTON THINKS HE CAN DRIVE YOU CRAZY.

So you think there's no puzzle too tough for you and no video game you can't beat. Welcome to PuzzlePanic—The computer game that's sure to have you crying "uncle." Designed by Ken Uston, blackjack and arcade game player extraordinaire, PuzzlePanic takes you through 49 increasingly difficult screens based on seven different games of action, logic, strategy and challenge. Compared to PuzzlePanic, Rubik's Cube™ is child's play. So put on your thinking hat, grab your joystick, get ready for the contest of your life, and let Ken Uston drive you crazy.

One player; joystick controlled.

EPYX™ COMPUTER SOFTWARE

Strategy Games for the Action-Game Player shots forms rows across the screen. Several rows are required to form a character. For example, alphanumeric characters for many computers are eight lines tall. To see this, type some spaces in inverse video. If you look closely, you'll see that the reverse space character is a stack of thin, closely spaced lines rather than a solid block.

The drawing process must be repeated over and over because the bullet holes of light glow for only a fraction of a second before fading away. In most computers, the screen is redrawn every 1/60 second. If you had a very fast stopwatch that you started when the gun began firing at the upper left corner, you could read the elapsed time on the watch when the spray of electrons reached any particular point on the screen and, from this reading, determine how far you were from the starting position at the upper left.

This is the secret of light pen operation. In its simplest form, the pen is a plastic cylinder housing a phototransistor, a light-activated switch. (The phototransistor is what you see behind the lens at the end of the pen.) When the pen is held to the screen, the beam of electrons which light up the screen triggers the phototransistor, causing it to signal the computer that the beam has been detected. The computer must then check its video stopwatch to see how much time has elapsed since it started drawing the screen. It can then compute where on the screen the pen is being held.

If the screen is being drawn many times a second, the pen will detect the spray of electrons each time the screen is drawn. Since the spots of light are so small, the pen may be triggered at a slightly different point each time. The readings you get from simple pens can thus be somewhat unsteady, especially for the horizontal location of the pen. Programs written for simple pens usually require that you touch a key on the keyboard to tell the computer when you want the reading to be accepted.

More sophisticated (and hence more expensive) pens have additional circuitry which allows them to latch after triggering so that the readings do not change every time the screen is drawn. This means the readings will be much more stable, and that you will not need to use the keyboard. A switch in the pen tells it when to hold the current reading. In some, the switch is built into the nose of the pen so that you latch the reading by simply pressing the pen against the screen.

If you want to use a light pen or a graphics tablet with your own programs, keep in mind that they won't draw on the screen for you. Like a joystick or a set of paddles, the pen or pad provides only numeric readings. It's up to you to write the software which will decipher the input from the pen or tablet and then accomplish something in your program.

three-year-olds using them."

Beyond that, graphics tablets and light pens can attract adults as well. "People seem to enjoy being able to sit down and draw," says Ranson. "Graphics tablets meet a fundamental human need—the need to express oneself."

Light Pen Elbow?

The naysayers to these new communication tools complain about physical inconveniences. Your arm can get tired from holding the light pen. The cord can get tangled and caught under things. It can be confusing to look back and forth from the tablet to the screen.

"There is a spatial problem when you're writing or drawing on the screen," says Ranson. "But I don't think it's major." Elliot argues that "Once familiarity sets in, you don't look at the tablet."

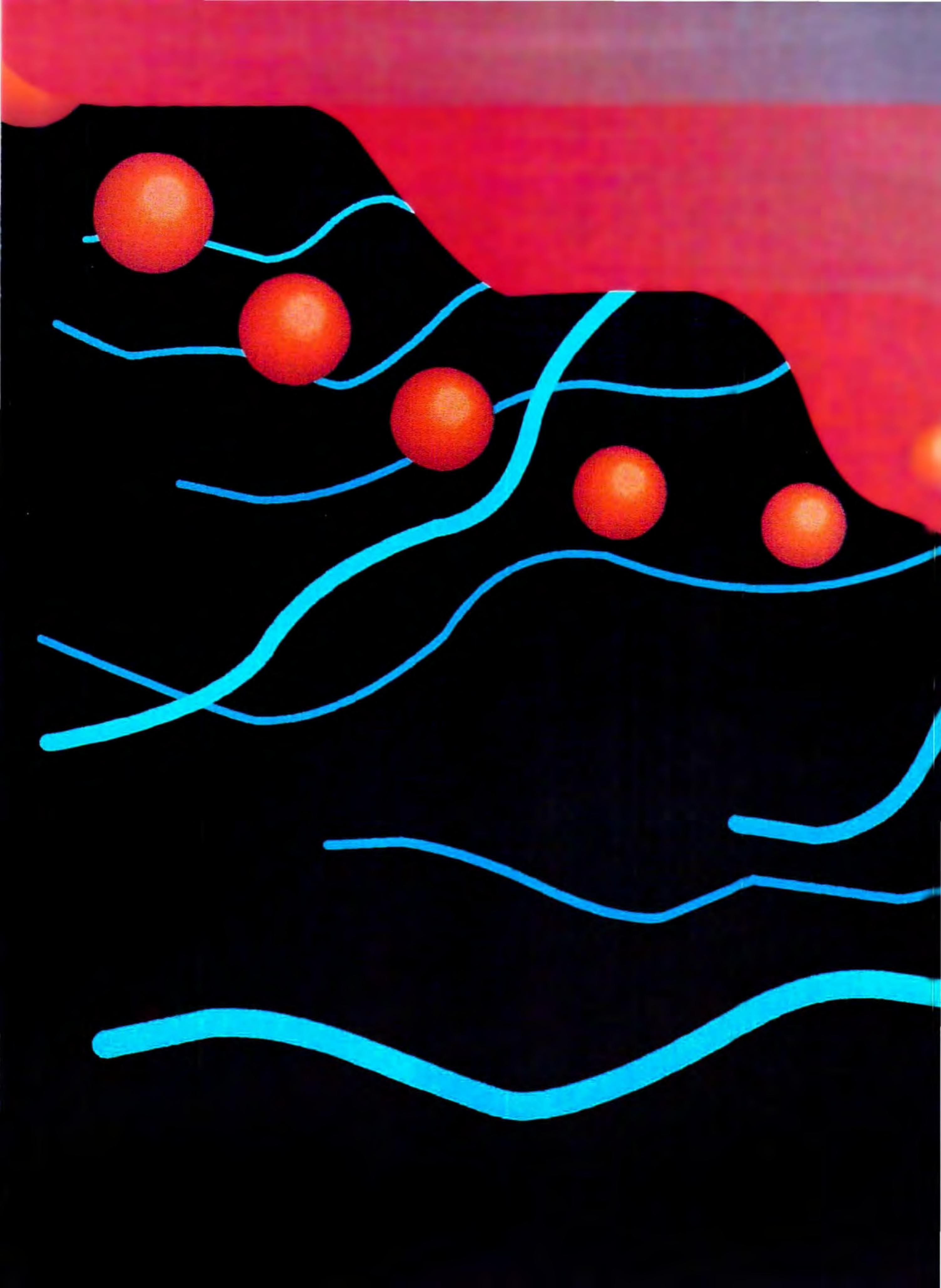
A more serious accusation is that they're kids' stuff, that you can't do more than draw pictures of trees and bunnies and houses.

"Granted, they're great for kids," says Byrne Elliot. "I know a lot of kids use Flexidraw for things like making valentines. But I also know of women who use it to design wallpaper and make dress patterns, and professionals who draw up plans for the interiors and exteriors of buildings with it. Graphics is becoming a lot more important to different kinds of people."

Input For The Future

Will a new input device come along that will make pens and tablets or, for that matter, keyboards obsolete? "Voice recognition is not as simple as some people think," says Chalkboard's Ranson. "Say you're a writer and want to sit down and dictate your work. I defy you to read everything you write all day. You'll lose your voice.

"Until we've reached the ultimate, there will be a lot more people exploring how to get into the computer other than QWERTY. We've been existing with joysticks and keyboards for a long time now. There will always be room for more than one input device." ©



A Portrait Of The Computer Artist

Realtime Dreaming

With Mike Newman

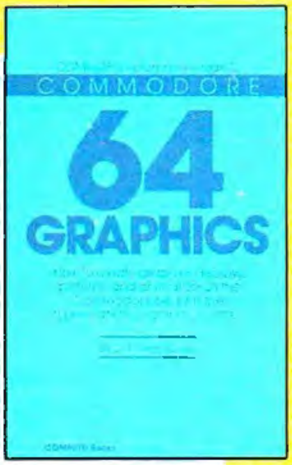
Selby Bateman
Assistant Editor, Features

"Realtime dreaming" is Mike Newman's description of his computer art. He spoke with us recently about his development as a computer artist and the future of personal computer art.

Newman is quickly becoming one of the recognized masters of computer art. His work has been exhibited worldwide. Many of his computer paintings were featured in Joseph Deken's recent book Computer Images: State of the Art.

Newman, 29, is supervisor of Creative Services for the DICOMED Corporation of Minneapolis, an international leader in precision computer graphics. What started four years ago as a part-time experiment with DICOMED has since blossomed into a full-time commitment to computer art. His paintings were created on a $130,000 state-of-the-art computer design station.

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.

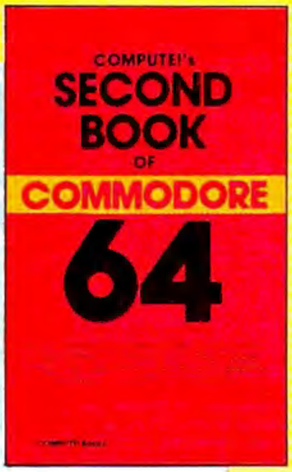
$12.95
ISBN 0-942386-29-9

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

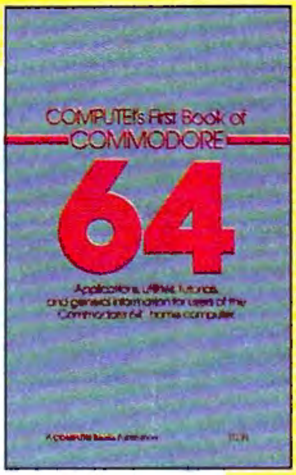


COMPUTE!'s Second Book of Commodore 64

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

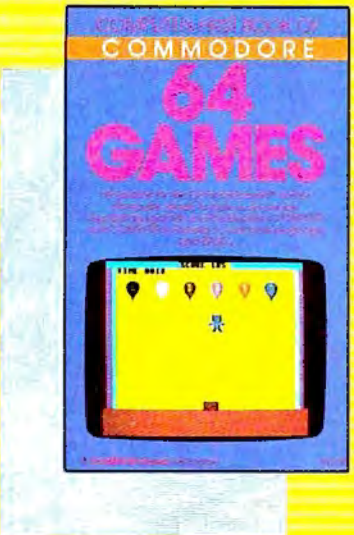


COMPUTE!'s First Book Of Commodore 64

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

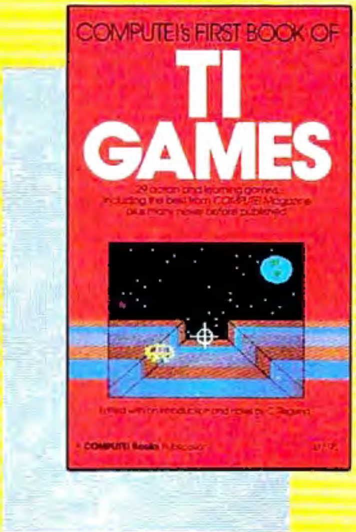


COMPUTE!'s First Book Of Commodore 64 Games

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

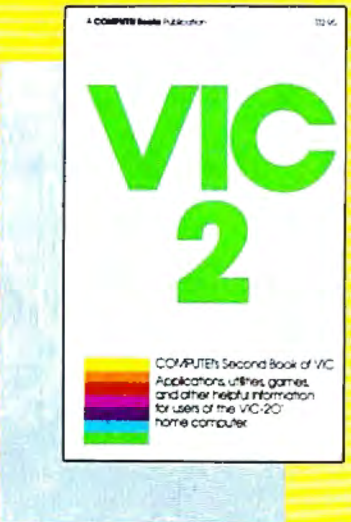


COMPUTE!'s First Book Of TI Games

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

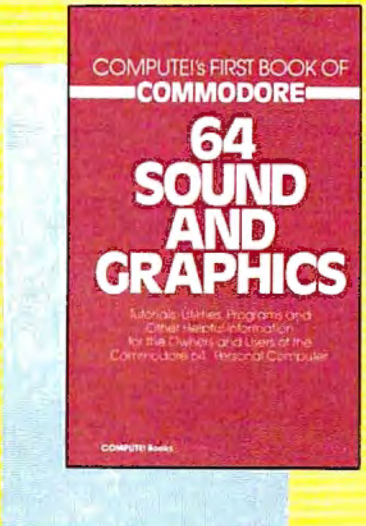


COMPUTE!'s Second Book Of VIC

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.
One of the ABC Publishing Companies
Post Office Box 5406, Greensboro, North Carolina 27403

picking these icons, or pictograms. For instance, if you want to examine color and work with the color menu, you'd go to a magnifying glass icon that's perched over a picture of a rainbow. And that means to examine color.

The design station has 640 pixels across on the screen and 480 down. We write everything down on the disk in very high resolution—8000 lines. We can zoom into an area. We don't blow up the frame buffer as on other systems where you get great big pixels, big squares. We read it off a disk and redraw it so that we can actually address the resolution that the film recorder [which turns the image into a color slide] is going to see. So, we have the addressability of that high resolution, which is unique.

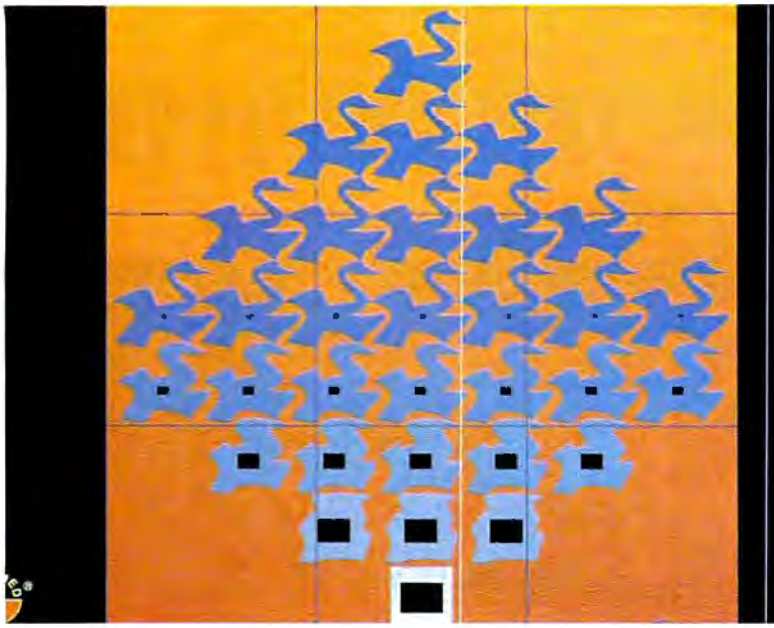
Then everything goes directly onto the disk. We have over 16 million colors that we can blend and use on the system. One of the things that we found out early on, and one of the things I was adamant about, was that we don't have to consciously dump the data to disk. We have a continuous disk update system. But you can back yourself up in case you're going in a direction that you don't like and you want to get back to a previous state.

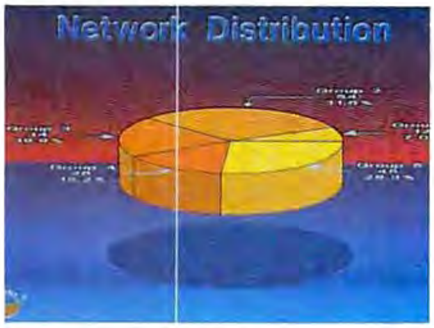
As an artist, you're generally intensely working on something and the last thing in the world you want to do is to remember to save this or that.

COMPUTE!: What's the attraction of computer art?

Newman: The first thing that attracted me was that it took only about ten minutes to understand that this was just another tool, and that you could put a computer and art together.

Computer artist Mike Newman inspects his own digitized image.





Newman created the commercial graphics bar chart and surrounding artwork (far left) by using an Apple IIe and the DICOMED D38 and Imaginator design stations. "Geese," (top) an example of computer interpolation in which Escher-style geese and a photographic slide are transformed. The demands of commercial graphics (bottom) have helped to stretch the limits of computer art.

Some people think of the stereotype of a mathematically oriented artist, of which there are very, very few. There are some who do write their own programs and do artwork, but they are in a completely different ballpark. They are artists and scientists at the same time.

I figured if I could do this, then anybody could. It also attracted me that this was not a threat, because the computer wasn't going to do anything without me. It wasn't going to do anything terrific without somebody who knew about art.

COMPUTE!: What made you reach that conclusion?

Newman: When I saw the work that some of the programmers were doing. They weren't doing terrific work [artistically], although one of the programmers does really great graphics because he also likes art.

It became clear very quickly what the benefits were: I could make a piece of art and experiment with it, begin to do things with it, and *see* that instead of just think about it. In conventional graphics you say "I wonder what this would look like if it were smaller, or turned a little bit?" Whatever the changes—color, position, rotation, duplication—instead of thinking about it, with the computer you can try it. You just do it.

It allows the artist to do real-time dreaming, giving you a much stronger sense of design. It's the same thing with color. The best thing I ever did was to take up watercolors because I had to understand what colors were doing when they were on top of one another, when they were mixed together. And the computer just enhances all that.

Now I blend colors in the same way using the computer, but I can see the artwork. After I have the artwork done, I can begin to play with colors and with shapes. It's like working on a painting and the paint never dries. You can still work with it, but it's more permanent than paint because it's digital on a magnetic medium. So, the permanency is neat, but the flexibility is just remarkable.

COMPUTE!: How do you answer critics who say that computer art is not a genuine art form?

Newman: I think that's a real misconception. People get the opinion that computer art is something done by a computer. In fact I'll read that occasionally: "This art was made by computer." Well, that whole concept is wrong. This art was made by a human *using* a computer. You don't say "This photograph was made by a camera." You usually give credit to the photographer, and it's the same thing with a computer. Not only that, but the person who wrote the program for the computer did a lot of creative programming. There's creativity there, too.

If you look at the wide variety of computer artwork, you

AT LAST, A LETTER-QUALITY PRINTER FOR TIGHTWADS.

And we don't mean a bargain-basement special, either.

Not by a long shot. The Abati LQ-20 is a top-quality printer, all the way. Offering quiet, bidirectional operation, an 18 characters per second printing speed, single sheet or optional continuous form tractor feed, and a full one-year warranty. All at a price even Ebenezer Scrooge would love.

And you'll love it, too.

Especially when you consider that the Abati LQ-20 is compatible with IBM, Apple, or any other microcomputer you might own.

Plus, it's incredibly easy to configure for word processors and spreadsheets, as well as most other software. And it's available in either serial or parallel formats.

At $479,* it's easy to see why anyone looking for a great buy in letter-quality printers is looking at the Abati LQ-20. So should you. After all, you don't have to be a tightwad to buy the Abati LQ-20. Just act like one. Call 1-800-447-4700.

*Parallel model.

"Ed's First String Art."

can see that it's not just taking a picture and doing image processing, like distorting a picture. It's starting from nothing, a blank screen. Instead of a piece of paper, you have a video screen. And you work with shapes and colors and light and textures and all of the conventional things that we know about art. And you put these things together.

It has human emotional feelings built-in, just as every art form does. That's what makes it art. My work looks different from somebody else's work. That's because emotions are involved, and that's what computers don't have. Without the artist, it just sits there.

COMPUTE!: What influences have contributed to your computer artwork?

Newman: I take in as much information from as many sources as possible. I like to think that my visual artwork is influenced by music as much as it is by other artists. I don't draw boundaries between dance and literary art and visual art. To me, it's all-encompassing. So I may have a visual depiction of a song, or music may have a certain effect on me that will give me a different sense of color for a particular design.

I am, however, influenced by other artists. I'd say the first computer artist—who was a computer artist without a computer—was M.C. Escher. This gets back to saying that computer art is not necessarily art made by computers. It's art made by humans. If you look at Escher's work, it was made by the "computer" that he carried around with him. To me, he is the first computer artist. The difference is that he didn't have a computer.

Also, I'm very influenced by design technology—the revolutionary. Buckminster Fuller, although he wasn't an artist in the conventional sense, had a lot of influence on my work. And a host of a thousand musicians and other artists. [Laughs] In an unconscious way, I'm affected by everything that I see—Andy Warhol, James Wyeth—I enjoy everything.

COMPUTE!: What advice would you give those who want to get started in computer art?

Newman: The first thing you want to do to be a computer artist is to be a good artist. You can learn the computer part, but it's hard to develop artistically. Whether you do it in art school or on your own, develop the artistic talent first.

I was not willing to become only a fine artist, because I was afraid that I would wind up being a starving artist, and that's not what I wanted in life. Some

Now your Commodore 64 can write letters like I.B.M.

Our **Super-Text**® Professional Word Processor is now available for the Commodore 64. And that's great news for Commodore owners. Because now you can have all of the advanced features I.B.M. and Apple owners have enjoyed for years. All of the features you'll ever need.

You'll find **Super-Text**® makes all of your writing easier. It's indispensable for school work, home record keeping, business correspondence, or just plain letter writing. It's ideal for both students and businessmen.

Best of all, you don't have to be a computer whiz to use **Super-Text**®. Our **On-Screen Instructions** and simple commands allow you to quickly create professional looking correspondence. An **Exclusive 80 Column Display** and **On-Screen Formatting** shows how your document will appear before it's printed. And **Super-Text**® is the best Commodore word processor that provides 80 column display without costly additional hardware.

Super-Text®. It's the best way to make your Commodore write like an Apple. Even an I.B.M.

Super-Text® is available for Atari 48k*, Apple II and Apple IIe, Commodore 64 and IBM PC.

FOR COMMODORE 64 WITH 1541 DISK DRIVE. $99.00

*80 column display & on-screen formatting not available on Atari.

Muse offers you more.

PROFESSIONAL WORD PROCESSOR $99.00

AWARD WINNING ARCADE/ ADVENTURE GAME $29.95

CHALLENGING ADVENTURE/ STRATEGY GAME $29.95

24 EXCITING GAMES PLUS A SECRET SCREEN $29.95

MUSE® SOFTWARE
347 N. Charles Street
Baltimore, Maryland 21201
(301) 659-7212

Apple, Atari, Commodore and IBM are registered trademarks of Apple Computer Corporation, Atari Inc., Commodore Business Machines and International Business Machines Corporation respectively.

Newman's "Metamorphosis" is another form of computer interpolation. Caterpillar becomes butterfly.

people feel so committed to the work they're doing that they'll take that. Those are conventional decisions you make about art, and they need to be made.

There are some schools in the United States and Canada now that are beginning to have computer graphics programs. They will give you a good overview of the types of systems that are out there, and also give hands-on experience on the equipment so that they know what computer graphics is all about.

COMPUTE!: What are the limitations in computer art?

Newman: There's no medium that does everything. The more painterly aspects of art are hard to simulate. In order to get the high resolution we have, we're based on what we call graphic primitive shapes. You tell the computer you want to make a line, and it knows you want to make a line. You tell it you want to make a perfectly round shape, and it expects you to tell it if you want a full round shape, how big it's going to be, and where it's going to start and stop.

There are other systems that work on a property of more painterly aspects, and these systems are called paint systems. They're more like what you see on personal computers these days. That is, you say you want to make a brush that is *this* fat and you want it to be *this* color, for instance.

The only problem is that you're just concerned with which little lights [pixels] are on and off, and it's hard to translate that into high resolution. You can't just take a display, even if it's a high-resolution display, and, say, double it and expect it to look better.

I do feel, however, that at some point this won't be a problem. I still consider this medium to be in its very beginning stages. We're just coming out of the basement now. All I know is that as an artist who has access to computers, I have a lot to look forward to. And I expect many great things to occur. ©

3-D Plotting

Tim R. Colvin

How many times have you seen beautiful three-dimensional graphics in the ads for video monitors and printers? Now, with these easy-to-use programs, you can create three-dimensional images of your own. Versions are included for the Commodore 64, Atari, Apple, IBM PC and PCjr.

These two programs, "Rectan" and "Spheri," will plot three-dimensional figures using information which you provide.

You don't really need to delve into the mathematics which produce the images. You can just fiddle with the examples given to produce many effective displays. Let's look at some graphic examples. First type in each program and SAVE it to tape or disk.

Then LOAD Rectan. To have Rectan draw a hyperbolic paraboloid, or "saddle function" (it resembles a riding saddle), replace line 790 with:

```
790 Z=X*X/4-Y*Y/9
```

and give the following inputs:

-2,2,-3,3,25,25,45

For another interesting design, use:

```
790 Z=-1/(X*X+Y*Y+.5)
```

and give the following inputs:

-1,1,-1,1,20,20,45

The program will print SCREEN SCALING IN PROGRESS. The program is scaling the image to fit on the screen, which can require a lot of time. The rule is: The more complicated the description of the surface, the longer this step takes.

The Plotting Begins

When the previous step is completed, the screen will clear and turn cyan. The high-resolution plotting now begins. When the plot is finished, the color of the top left corner of the screen will change color. The program is locked in a loop so you can look at your creation. When you have finished looking at the display hold down RUN/STOP and hit RESTORE.

A Spheri Demonstration

To see a torus (doughnut shape), type NEW to clear memory. Then LOAD Spheri, replace lines 820–840 with:

```
820 XT=(4+C1)*C2
830 YT=(4+C1)*S2
840 ZT=S1
```

and give the following inputs:

0,360,0,360,25,25,45

For a sphere, use:

```
820 XT=C1*C2
830 YT=C1*S2
840 ZT=S1
```

and give the following inputs:

0,360,0,180,15,15,45

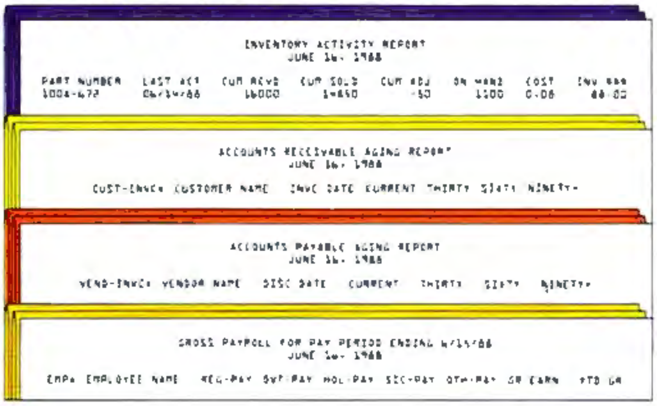
An Illusion Of Depth

These programs use *rectangular* and *spherical coordinate systems* to create an illusion of depth in the screen image. You're probably familiar with the X-Y coordinate system used to specify the location of a point on a flat surface. For example, in Figure 1 the point is located five units over on the X axis

WHO'S GOT 7 NEW BUSINESS SYSTEMS FOR ME? TIMEWORKS.

Here's a series of seven easy-to-use Management Information Reports flexibly designed to accommodate all small and medium size business accounting requirements for Commodore 64* computers.

Inventory Management / Sales Analysis Management / Accounts Receivable Management and Invoicing / Accounts Payable Management and Checkwriting / Payroll Management / Cash Flow Management / General Ledger



General Overview

Each system includes:

A Menu-driven program, sophisticated enough to provide complete Management and Product information, yet requires no prior computer or accounting knowledge to operate.

A unique method of creating your own, unlimited array of reports — easily and quickly. YOU select the information you want, and YOU determine the sequence of the report column headings.

A program which can be used by itself (stand-alone), or can be interfaced, one at a time, with other TIMEWORKS MANAGEMENT INFORMATION programs into a fully integrated accounting system.

A Manual written in easy-to-understand, people-friendly English, abundantly illustrated to provide further clarity and eliminate guesswork.

Password Protection, to prevent unauthorized access to confidential data.

Suggested Retail List at $59.95 each.

Customer Support Plan

Timeworks Telephone Consulting Service is available to all users at no charge to support your installation and ongoing operations.

For further details, contact your local Dealer or Timeworks, Inc. Choose from a host of Commodore 64 programs. Now at your favorite dealer. Or, contact Timeworks, P.O. Box 321, Deerfield, IL 60015. 312-291-9200.

TIMEWORKS

SOFTWARE WITH SUBSTANCE — FOR EVERYONE.

and six units up on the Y axis. The point is said to be at location 5,6.

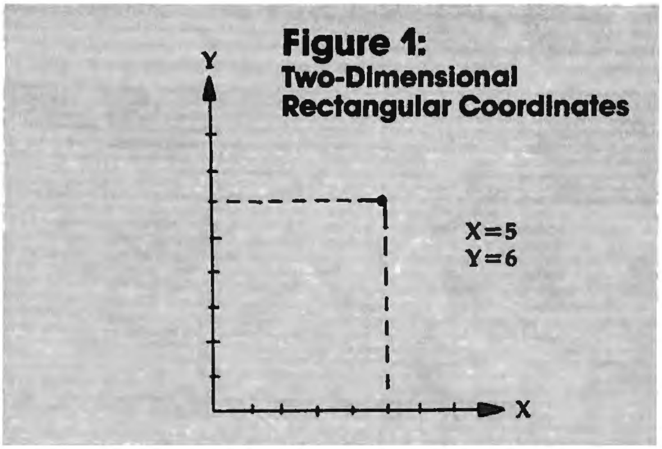


Figure 1: Two-Dimensional Rectangular Coordinates

This simple system works well for specifying the location of a point in a two-dimensional design on a flat surface, but for 3-D plotting you need a third coordinate.

Several coordinate systems are commonly used to plot three-dimensional surfaces. The particular coordinate system you should use depends on the shape you want to draw. Any system can be used, but if you choose the right system, you can simplify your calculations considerably.

A Simple Solution

The easiest system to understand is just an extension of the rectangular (X-Y) coordinates you are already familiar with. All you need to add is a third coordinate (Z) for the third dimension. For example, the point in Figure 2, below, is located five units out on the X axis, six units over on the Y axis, and four units up on the Z axis. The point is said to be at location 5,6,4.

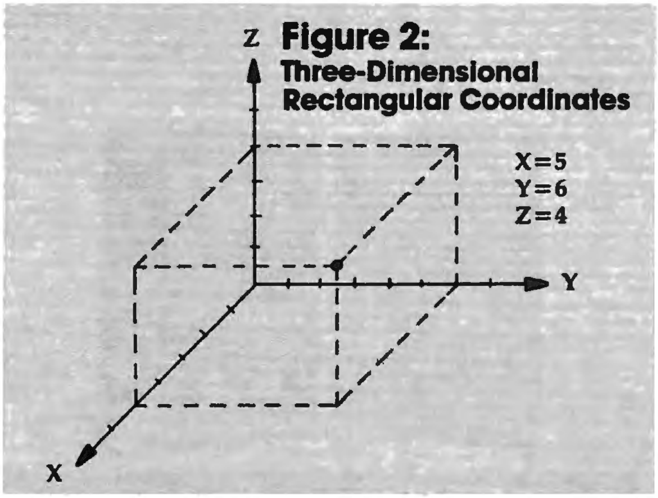


Figure 2: Three-Dimensional Rectangular Coordinates

A System For The Stars

On the other hand, if the design you wish to draw is roughly the shape of a sphere, you should use *spherical coordinates*. In that system, a point is described by two angles and a distance from the origin. For example, astronomers use spherical coordinates to describe the position of a star relative to the earth. The *azimuthal angle* of the star, designated by the Greek letter theta (θ), is the direction you must face to view the star. If north is taken to be zero degrees, then a star that lies due east has an azimuthal angle of 90 degrees. The *elevation angle*, designated by the Greek letter phi (ϕ), specifies how much you must tilt your head back to look directly at the star. If the horizon is taken to be zero degrees, a star that is directly overhead has an elevation angle of 90 degrees. Finally, the *radial distance*, designated by the letter r, is the distance between the earth and the star.

Using spherical coordinates, the point shown in Figure 2 has an azimuthal angle of 50.2 degrees, an elevation angle of 33.7 degrees, and a radial distance of 8.77 units, as shown in Figure 3.

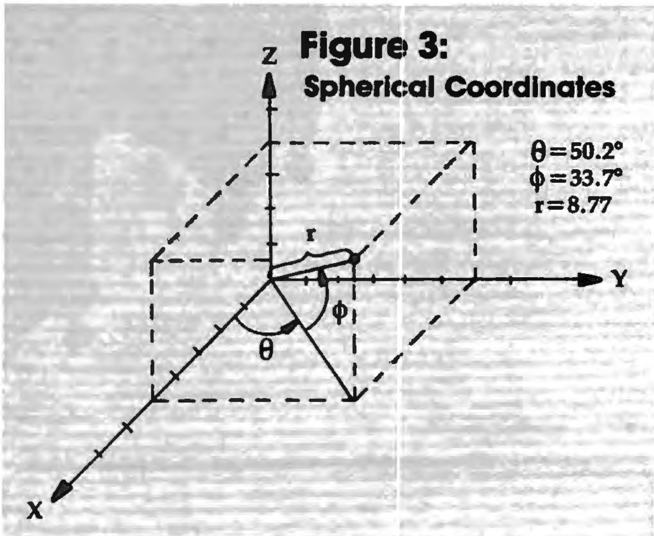


Figure 3: Spherical Coordinates

Despite the fine graphics they produce, these programs have a couple of limitations. Screen pixels are taller than they are wide, which makes spheres look slightly less round than they should. Also, we see the surface as if it were transparent and contour lines were drawn on it. A more advanced program (such as those available commercially) would remove lines that we couldn't see if the surface were not transparent.

Program 1: Rectan—64 Version

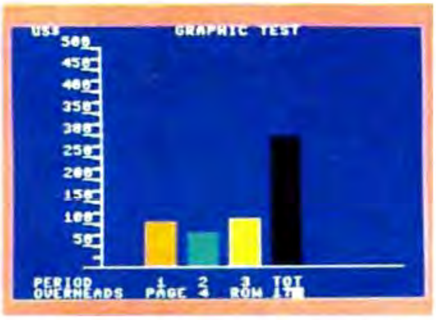
Refer to the "Automatic Proofreader" article before typing this program in.

```
100 REM * THREE-DIMENSIONAL SURFACES *
                                  :rem 253
110 REM * IN RECTANGULAR COORDINATES *
                                  :rem 212
130 PRINT CHR$(147)                :rem 15
140 PRINT "LOWER X LIMIT ";:INPUT A1
                                   :rem 61
150 PRINT "UPPER X LIMIT ";:INPUT B1
                                   :rem 66
160 PRINT "LOWER Y LIMIT ";:INPUT A2
                                   :rem 65
```

Handic – A Complete Line of Products to Keep Your Commodore 64 Busy – Everyday!

Handle your home budget, stock portfolio, loans and mortgages with Calc Result

Calc Result Easy is a simple-to-use spreadsheet program for the Commodore 64. It includes 254 lines × 64 columns, built-in graphics, and flexible printout formats. Plug-in cartridge... just plug it in and its ready. Perfect for cash flow analysis, personal net worth, IRA analysis, travel expenses, credit card expenditures, gas and electricity bills, etc.



Calc Result Easy $49.95

Calc Result Advanced gives you 32 pages of interrelated information. The three-dimensional feature allows you to consolidate calculations in summary format. Calc Result Advanced comes on plug-in cartridge and disk. Disk drive required.

Calc Result Advanced $99.95

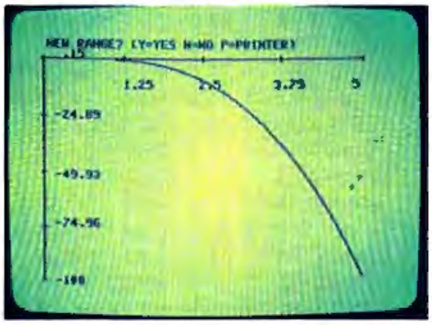
A complete database for the home

Addresses, telephone numbers, appointments, birthdays, or records–whatever you want to remember–put it on DIARY, an electronic notebook for home use. DIARY comes on a plug-in cartridge. Its easy to use and easy to learn, giving you the flexibility to design a personal calendar or address book.

Diary $29.95

Turn statistical information into graphic format

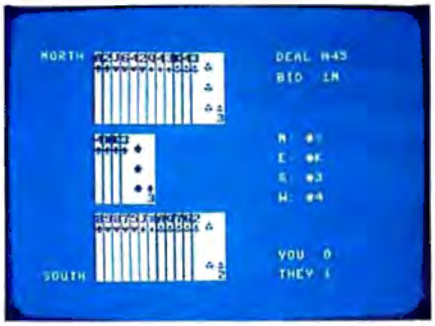
GRAF 64 converts mathematical functions into graphical analysis on the Commodore 64. An ideal program for studying math. Define a function, set the limits of an axis, plot a graph and display the extreme points, intersection values, etc.



Graf 64 $29.95

Develop your bridge skills

Whether you're an experienced bridge player or a beginner, polish your skills or learn the game with BRIDGE 64. Play North-South, then switch to East-West in the same deal, the return to that deal again and test your skill with a different strategy.



Bridge $39.95

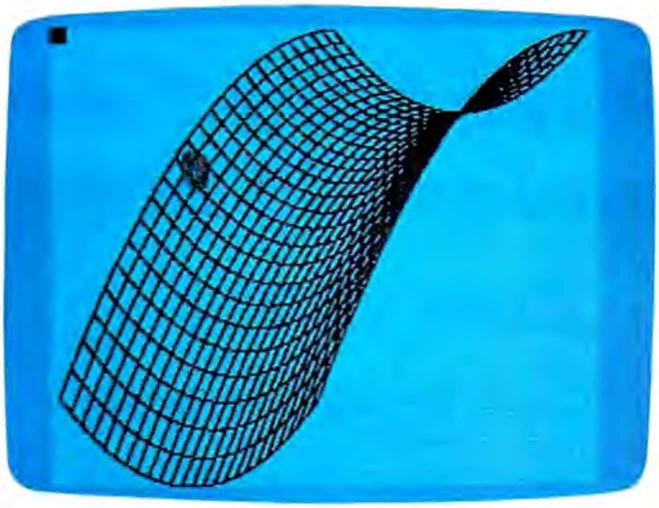
Handic–for the broadest range of Commodore products

As the largest independent developer of Commodore software and accessories, Handic's broad range of business, education and recreation products are designed exclusively for the Commodore user who demands quality and reliability.

For more information and a catalogue of our products, see your nearest Commodore dealer, or call us direct.

handic software

Handic Software, Inc.
Fellowship Business Center
520 Fellowship Road, B 206
Mount Laurel, NJ 08054
Phone (609)663-0660



The hyperbolic paraboloid resembles a saddle or a trough curving downward.

```
170 PRINT "UPPER Y LIMIT ";:INPUT B2
                                      :rem 70
180 PRINT "SLICES IN X ";:INPUT N:rem 111
190 PRINT "SLICES IN Y ";:INPUT M:rem 112
200 PRINT "OBSERVATION ANGLE ";:INPUT Q
                                     :rem 108
210 PRINT "SCREEN SCALING IN PROGRESS"
                                      :rem 49
220 Q=Q*.0174532925                  :rem 209
230 CS=COS(Q)                        :rem 239
240 SI=SIN(Q)                        :rem 251
250 H1=(B1-A1)/319:H2=(B2-A2)/(N-1)
                                     :rem 254
260 H3=(B1-A1)/(M-1):H4=(B2-A2)/319:rem 2
270 M1=99999999:M2=M1:N1=-M1:N2=N1
                                     :rem 167
280 FOR Y=A2 TO B2 STEP H2            :rem 87
290 FOR X=A1 TO B1 STEP H1            :rem 84
300 GOSUB 610                        :rem 170
310 NEXT X                            :rem 43
320 NEXT Y                            :rem 45
330 FOR X=A1 TO B1 STEP H3            :rem 81
340 FOR Y=A2 TO B2 STEP H4            :rem 86
350 GOSUB 610                        :rem 175
360 NEXT Y                            :rem 49
370 NEXT X                            :rem 49
380 D=8192:POKE 53272,PEEK(53272)OR8
                                     :rem 218
390 POKE 53265,PEEK(53265)OR32       :rem 125
400 FOR I=D TO D+7999:POKE I,0:NEXT I
                                       :rem 9
410 FOR I=1024 TO 2023:POKE I,3:NEXT I
                                       :rem 6
420 T1=(N1-M1)/2                      :rem 52
430 T2=(N2-M2)/2                      :rem 56
440 W=T1/T2                          :rem 102
450 IF W<1.60606061 THEN 480         :rem 126
460 XS=159:ZS=159/W                  :rem 106
470 GOTO 490                         :rem 113
480 XS=99*W:ZS=99                     :rem 13
490 FOR Y=A2 TO B2 STEP H2            :rem 90
500 FOR X=A1 TO B1 STEP H1            :rem 78
510 GOSUB 690                        :rem 181
520 NEXT X                            :rem 46
530 NEXT Y                            :rem 48
540 FOR X=A1 TO B1 STEP H3            :rem 84
550 FOR Y=A2 TO B2 STEP H4            :rem 89
560 GOSUB 690                        :rem 186
570 NEXT Y                            :rem 52
580 NEXT X                            :rem 52
590 POKE 1024,16                      :rem 39
600 GOTO 600                         :rem 101
610 GOSUB 790                        :rem 183
620 XT=X-Y*CS                         :rem 31
630 ZT=Z-Y*SI                         :rem 42
640 IF XT>N1 THEN N1=XT               :rem 41
650 IF XT<M1 THEN M1=XT               :rem 38
660 IF ZT>N2 THEN N2=ZT               :rem 49
670 IF ZT<M2 THEN M2=ZT               :rem 46
680 RETURN                           :rem 126
690 GOSUB 790                        :rem 191
700 XT=160+INT(XS*(X-Y*CS-N1+T1)/T1)
                                      :rem 82
710 ZT=100-INT(ZS*(Z-Y*SI-N2+T2)/T2)
                                      :rem 94
720 RO=INT(ZT/8)                     :rem 200
730 CH=INT(XT/8)                     :rem 177
740 LN=(ZT)AND7                      :rem 123
750 BI=7-((XT)AND7)                   :rem 32
760 BY=D+320*RO++8*CH+LN              :rem 76
770 POKE BY,PEEK(BY)OR(2↑BI)         :rem 178
780 RETURN                           :rem 127
790 Z=X*X/4-Y*Y/9                    :rem 229
800 RETURN                           :rem 120
```

Program 2: Spheri—64 Version

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

```
100 REM * THREE-DIMENSIONAL SURFACES *
                                     :rem 253
110 REM *{2 SPACES}IN SPHERICAL COORDINAT
    ES{2 SPACES}*                     :rem 55
130 PRINT CHR$(147)                   :rem 15
140 PRINT "LOWER THETA LIMIT ";:INPUT A1
                                      :rem 91
150 PRINT "UPPER THETA LIMIT ";:INPUT B1
                                      :rem 96
160 PRINT "LOWER PHI LIMIT ";:INPUT A2
                                     :rem 201
170 PRINT "UPPER PHI LIMIT ";:INPUT B2
                                     :rem 206
180 PRINT "SLICES IN THETA ";:INPUT N
                                     :rem 141
190 PRINT "SLICES IN PHI ";:INPUT M
                                     :rem 248
200 PRINT "OBSERVATION ANGLE ";:INPUT Q
                                     :rem 108
210 PRINT "SCREEN SCALING IN PROGRESS"
                                      :rem 49
220 U=.0174532925                     :rem 90
230 Q=Q*U                            :rem 243
240 CS=COS(Q)                        :rem 240
250 SI=SIN(Q)                        :rem 252
260 H1=(B1-A1)/319:H2=(B2-A2)/(N-1)
                                     :rem 255
270 H3=(B1-A1)/(M-1):H4=(B2-A2)/319:rem 3
280 M1=99999999:M2=M1:N1=-M1:N2=N1
                                     :rem 168
290 FOR Y=A2 TO B2 STEP H2            :rem 88
300 FOR X=A1 TO B1 STEP H1            :rem 76
310 GOSUB 620                        :rem 172
320 NEXT X                            :rem 44
330 NEXT Y                            :rem 46
340 FOR X=A1 TO B1 STEP H3            :rem 82
350 FOR Y=A2 TO B2 STEP H4            :rem 87
360 GOSUB 620                        :rem 177
370 NEXT Y                            :rem 50
380 NEXT X                            :rem 50
```


Flak
For Apple,
Atari & Commodore 64
THE ULTIMATE FLIGHT EXPERIENCE
OVER TWO DOZEN GAME SCREENS!

The Mathematics Of 3-D Plotting

"Rectan" plots surfaces using rectangular coordinates (x,y,z). The values for x and y are specified; the value of z is then given by z=f(x,y) for some function f.

To use Rectan, specify the function f(x,y) in line 790. For example, z=x*x/4–y*y/9 defines a hyperbolic paraboloid.

"Spheri" plots surfaces using spherical coordinates. This method describes a point on the surface using three parameters: radial distance from the origin, r; azimuthal angle, theta (θ); and elevation angle, phi (ϕ).

To use Spheri, specify x,y, and z (called XT,YT, and ZT in lines 820–840) as functions of r, theta, and phi in lines 820–840.

Parameters And Slices

Both programs are structured the same. You specify parameter ranges. In Rectan these are for x and y; in Spheri, for θ and ϕ.

Next enter the number of *slices* for the parameters. Each slice corresponds to a contour line on the surface. A contour line is where one of the parameters is held constant.

Finally, you specify an observation angle. This is the angle which allows you to see a three-dimensional surface on a two-dimensional video screen. The most commonly used angle is 45 degrees.

If you'd like any technical information, or if you have a particular surface in mind but don't know how to write an equation for it, please write to:

Tim R. Colvin
1414 San Remo Dr.
Pacific Palisades, CA 90272

```
390 D=8192:POKE 53272,PEEK(53272)OR8
                                      :rem 219
400 POKE 53265,PEEK(53265)OR32       :rem 117
410 FOR I=D TO D+7999:POKE I,0:NEXT I
                                      :rem 10
420 FOR I=1024 TO 2023:POKE I,3:NEXT I
                                      :rem 7
430 T1=(N1-M1)/2                      :rem 53
440 T2=(N2-M2)/2                      :rem 57
450 W=T1/T2                           :rem 103
460 IF W<1.60606061 THEN 490          :rem 128
470 XS=159:ZS=159/W                   :rem 107
480 GOTO 500                          :rem 106
490 XS=99*W:ZS=99                     :rem 14
500 FOR Y=A2 TO B2 STEP H2            :rem 82
510 FOR X=A1 TO B1 STEP H1            :rem 79
520 GOSUB 700                         :rem 174
530 NEXT X                            :rem 47
540 NEXT Y                            :rem 49
550 FOR X=A1 TO B1 STEP H3            :rem 85
560 FOR Y=A2 TO B2 STEP H4            :rem 90
570 GOSUB 700                         :rem 179
580 NEXT Y                            :rem 53
590 NEXT X                            :rem 53
600 POKE 1024,16                      :rem 31
610 GOTO 610                          :rem 103
620 GOSUB 800                         :rem 176
630 XT=XT-YT*CS                       :rem 200
640 ZT=ZT-YT*SI                       :rem 211
650 IF XT>N1 THEN N1=XT               :rem 42
660 IF XT<M1 THEN M1=XT               :rem 39
670 IF ZT>N2 THEN N2=ZT               :rem 50
680 IF ZT<M2 THEN M2=ZT               :rem 47
690 RETURN                            :rem 127
700 GOSUB 800                         :rem 175
710 XT=160+INT(XS*(XT-YT*CS-N1+T1)/T1)
                                      :rem 251
720 ZT=100-INT(ZS*(ZT-YT*SI-N2+T2)/T2)
                                      :rem 7
730 RO=INT(ZT/8)                      :rem 201
740 CH=INT(XT/8)                      :rem 178
750 LN=(ZT)AND7                       :rem 124
760 BI=7-((XT)AND7)                   :rem 33
770 BY=D+320*RO+8*CH+LN               :rem 34
780 POKE BY,PEEK(BY)OR(2↑BI)          :rem 179
790 RETURN                            :rem 128
800 XA=X*U:C1=COS(XA):S1=SIN(XA)      :rem 206
810 YA=Y*U:C2=COS(YA):S2=SIN(YA)      :rem 213
820 XT=(4+C1)*C2                      :rem 70
830 YT=(4+C1)*S2                      :rem 88
840 ZT=S1                             :rem 11
850 RETURN                            :rem 125
```

Program 3: Rectan—Atari Version

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

```
BF 130 GRAPHICS 0
OP 140 ? "LOWER X LIMIT";:INPUT A1
PE 150 ? "UPPER X LIMIT";:INPUT B1
PD 160 ? "LOWER Y LIMIT";:INPUT A2
PI 170 ? "UPPER Y LIMIT";:INPUT B2
CB 180 ? "SLICES IN X";:INPUT N
CC 190 ? "SLICES IN Y";:INPUT M
BO 200 ? "OBSERVATION ANGLE";:INPUT Q
OD 210 ? "SCREEN SCALING IN PROGRESS"
IO 215 U=0.0174532925
PC 220 Q=Q*U
OP 230 CS=COS(Q)
PL 240 SI=SIN(Q)
PO 250 H1=(B1-A1)/319:H2=(B2-A2)/(N-1)
AC 260 H3=(B1-A1)/(M-1):H4=(B2-A2)/319
KH 270 M1=99999999:M2=M1:N1=-M1:N2=N1
FH 280 FOR Y=A2 TO B2 STEP H2
FE 290 FOR X=A1 TO B1 STEP H1
KK 300 GOSUB 610
CL 310 NEXT X
CN 320 NEXT Y
FB 330 FOR X=A1 TO B1 STEP H3
FG 340 FOR Y=A2 TO B2 STEP H4
KP 350 GOSUB 610
DB 360 NEXT Y
DB 370 NEXT X
CE 380 GRAPHICS 8
PB 390 SETCOLOR 2,0,0
OL 400 SETCOLOR 4,0,0
CI 410 SETCOLOR 1,9,15
EK 415 COLOR 1
OE 420 T1=(N1-M1)/2
```

you want to stay ahead of the personal computing ʼolution ...

ʍelcome to *COMPUTE!'s PC & PCjr!*

is exciting new magazine from *COMPUTE!* takes ı inside the PC and the incredible new PCjr to bring ı inside information you'll find nowhere else.

You'll discover how to get the most computer power your money. At home. At school. At work. With easy-run programs. Challenging ɔjects for advanced users. ınd new sound and graphics ɔlications. Plus some of the ɔst exciting computer ıctions outside of the top-ːret research labs!

We'll help you decide what buy. With independent ıluations of hardware, ftware and peripherals. mprehensive reviews of ʍ products as they're intro-ced. Hard-nosed evalua-ns of each machine's engths and weaknesses.

COMPUTE!'s PC & PCjr ıl keep you on the leading ge of personal computing e no other publication can. re are some of the features u can look forward to:

lcome to the PCjr! A com-te introduction to IBM's west personal computer, d a fascinating peek at the 88 microprocessor – the y-but-talented brain of the jr.

signing PCjr — The Inside ry. PCjr's creators tell how ey designed the new com-ter – from original planning to trade-offs to ultimate success.

Telecomputing with Your IBM. How to link up with distant computers over ordinary phone lines, access information services, even do office work at home with your own machine!

Music and Graphics. How to play songs and create sound effects with your PC or PCjr. Programs to generate impressive hi-res graphics and computer animation.

From the publishers of COMPUTE!

Announcing the magazine that takes you inside the IBM PC and the PCjr.

Your First Hour with an IBM. How to avoid those opening night jitters and get your computer up and running fast!

Financial Analysis. Ready-to-run programs to help you make intelligent investments.

PC vs. PCjr. Which one is right for you? *COMPUTE!'s PC & PCjr* helps you decide!

PLUS: Programming the function keys to suit *you*. Speeding up BASIC without resorting to machine language. Tape, disk, or hard disk? Buyer's guides. How to take advantage of free public-domain software. Computing activities for the whole family. Reviews. Games. Educational programs for children. And much, much more!

Subscribe right now and you can enjoy special Charter Subscriber Savings on *COMPUTE!'s PC & PCjr* – just $24 for the first 12 big issues. That's 33% off the cover price!

To start receiving *COMPUTE!'s PC & PCjr,* just mail the postpaid card in this issue or the coupon below today.

OR CALL TOLL FREE 1-800-334-0868

HARTER SAVINGS OFFER

[ ] ENTER my Charter Subscription to *COMPUTE!'s PC & PCjr.* I pay just $24 for the first 12 issues—a 33% saving off the cover price. What's more, I reserve the right to cancel at any time for a full pro-rata refund.

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

NAME ▸
ADDRESS ▸
CITY ▸
STATE/ZIP ▸

[ ] Payment enclosed
[ ] Bill me
Charge
[ ] VISA [ ] MasterCard
[ ] American Express

Acct. No. ____________

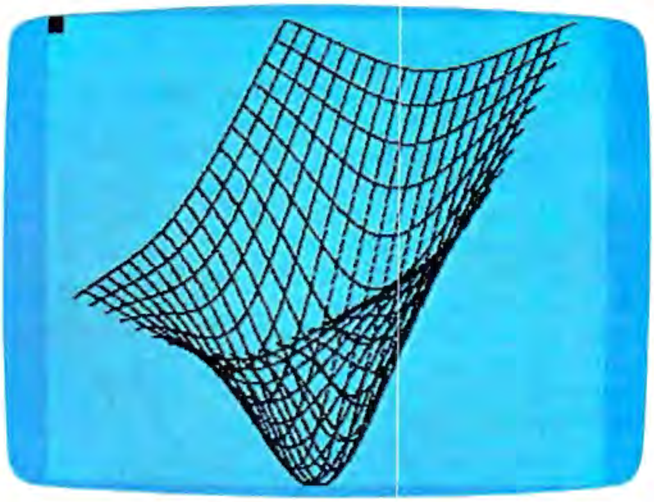
Exp. Date __/__/__

```
DI 430  T2=(N2-M2)/2
GG 440  W=T1/T2
HO 450  IF W<1.60606061 THEN 480
GK 460  XS=159:ZS=159/W
HB 470  GOTO 490
AJ 480  XS=79*W:ZS=79
FK 490  FOR Y=A2 TO B2 STEP H2
EO 500  FOR X=A1 TO B1 STEP H1
LF 510  GOSUB 690
CO 520  NEXT X
DA 530  NEXT Y
FE 540  FOR X=A1 TO B1 STEP H3
FJ 550  FOR Y=A2 TO B2 STEP H4
LK 560  GOSUB 690
DE 570  NEXT Y
DE 580  NEXT X
HF 590  END
LH 610  GOSUB 790
BP 620  XT=X-Y*CS
CK 630  ZT=Z-Y*SI
CJ 640  IF XT>N1 THEN N1=XT
CG 650  IF XT<M1 THEN M1=XT
DB 660  IF ZT>N2 THEN N2=ZT
CO 670  IF ZT<M2 THEN M2=ZT
HO 680  RETURN
LP 690  GOSUB 790
FC 700  XT=160+INT(XS*(X-Y*CS-N1+T1)/T1
        )
GF 710  ZT=80-INT(ZS*(Z-Y*SI-N2+T2)/T2)
FO 720  PLOT XT,ZT
HK 730  RETURN
LE 790  Z=-1/(X*X+Y*Y+0.5)
HI 800  RETURN
```

Program 4: Spheri—Atari Version

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

```
BF 130  GRAPHICS 0
AN 140  ? "LOWER THETA LIMIT";:INPUT A1
BC 150  ? "UPPER THETA LIMIT";:INPUT B1
HL 160  ? "LOWER PHI LIMIT";:INPUT A2
IA 170  ? "UPPER PHI LIMIT";:INPUT B2
DP 180  ? "SLICES IN THETA";:INPUT N
KK 190  ? "SLICES IN PHI";:INPUT M
BO 200  ? "OBSERVATION ANGLE";:INPUT Q
OD 210  ? "SCREEN SCALING IN PROGRESS"
IO 215  U=0.0174532925
PC 220  Q=Q*U
OP 230  CS=COS(Q)
PL 240  SI=SIN(Q)
PD 250  H1=(B1-A1)/319:H2=(B2-A2)/(N-1)
AC 260  H3=(B1-A1)/(M-1):H4=(B2-A2)/319
KH 270  M1=99999999:M2=M1:N1=-M1:N2=N1
FH 280  FOR Y=A2 TO B2 STEP H2
FE 290  FOR X=A1 TO B1 STEP H1
KK 300  GOSUB 610
CL 310  NEXT X
CN 320  NEXT Y
FB 330  FOR X=A1 TO B1 STEP H3
FG 340  FOR Y=A2 TO B2 STEP H4
KP 350  GOSUB 610
DB 360  NEXT Y
DB 370  NEXT X
CE 380  GRAPHICS 8
PB 390  SETCOLOR 2,0,0
OL 400  SETCOLOR 4,0,0
CI 410  SETCOLOR 1,9,15
EK 415  COLOR 1
DE 420  T1=(N1-M1)/2
DI 430  T2=(N2-M2)/2
GG 440  W=T1/T2
HO 450  IF W<1.60606061 THEN 480
GK 460  XS=159:ZS=159/W
```



The "Rectan" program was used to create this "fish net."

```
HB 470  GOTO 490
AJ 480  XS=79*W:ZS=79
FK 490  FOR Y=A2 TO B2 STEP H2
EO 500  FOR X=A1 TO B1 STEP H1
LF 510  GOSUB 690
CO 520  NEXT X
DA 530  NEXT Y
FE 540  FOR X=A1 TO B1 STEP H3
FJ 550  FOR Y=A2 TO B2 STEP H4
LK 560  GOSUB 690
DE 570  NEXT Y
DE 580  NEXT X
HF 590  END
LH 610  GOSUB 790
MH 620  XT=XT-YT*CS
NC 630  ZT=ZT-YT*SI
CJ 640  IF XT>N1 THEN N1=XT
CG 650  IF XT<M1 THEN M1=XT
DB 660  IF ZT>N2 THEN N2=ZT
CO 670  IF ZT<M2 THEN M2=ZT
HO 680  RETURN
LP 690  GOSUB 790
PK 700  XT=160+INT(XS*(XT-YT*CS-N1+T1)/
        T1)
NN 710  ZT=80-INT(ZS*(ZT-YT*SI-N2+T2)/T
        2)
FO 720  PLOT XT,ZT
HK 730  RETURN
NG 790  XA=X*U:C1=COS(XA):S1=SIN(XA)
NE 800  YA=Y*U:C2=COS(YA):S2=SIN(YA)
EG 820  XT=(4+C1)*C2
FI 830  YT=(4+C1)*S2
AL 840  ZT=S1
HN 850  RETURN
```

Program 5: Rectan—PC/PCjr Version

```
100 SCREEN 0,0,0:CLS
140 INPUT "Lower X limit ";A1
150 INPUT "Upper X limit ";B1
160 INPUT "Lower Y limit ";A2
170 INPUT "Upper Y limit ";B2
180 INPUT "Slices in X ";N
190 INPUT "Slices in Y ";M
200 INPUT "Observation angle ";Q
210 PRINT "Screen scaling in progress"
220 U=.0174532925#:Q=Q*U
230 CS=COS(Q)
240 SI=SIN(Q)
250 H1=(B1-A1)/639:H2=(B2-A2)/(N-1)
```

Jump on 10 monsters, 64 screens and $10,000 with Pogo Joe.™

A Mutated Wonderwhisk whisks by. The Spinning Top almost topples him!

Close. But Pogo Joe bounces back. Bouncing from cylinder to cylinder, screen to screen, Pogo Joe racks up point after point.

You guide him from cylinder to cylinder, changing the color on top of each. Change the top of each cylinder on a screen, then you're on to the next.

The more screens you complete, the nastier the monsters you face, and the faster they attack.

Press the fire button! Jump two cylinders to safety. Hop into a transport tube, and then whoosh! Pogo Joe appears across the screen. Jump on an escaping monster. Blam! It's gone in a flash! Only to reappear out of thin air.

Keep bouncing Joe to original music on realistic 3-dimensional cylinders. All the characters in this rollicking game are also 3-dimensional and fully animated. The graphics almost jump off the screen, leaving the arcades behind.

What's ahead with *Pogo Joe*™ is $10,000. Simply tell us what magic word appears after *Pogo Joe's* tenth screen. If your name is drawn from among the correct answers you'll win $10,000!

No purchase is necessary. You'll find entry forms at any store that sells Screenplay™ games.

But if you don't win you can't lose. *Pogo Joe*™ is so much fun you'll jump for joy no matter what.

screenplay™

Box 3558, Chapel Hill, NC 27514 800-334-5470

Pogo Joe in 48-64K on the Atari and Commodore 64. See your local software dealer.

```
260 H3=(B1-A1)/(M-1):H4=(B2-A2)/639
270 M1=99999999#:M2=M1:N1=-M1:N2=N1
280 FOR Y=A2 TO B2 STEP H2
290 FOR X=A1 TO B1 STEP H1
300 GOSUB 610
310 NEXT X
320 NEXT Y
330 FOR X=A1 TO B1 STEP H3
340 FOR Y=A2 TO B2 STEP H4
350 GOSUB 610
360 NEXT Y
370 NEXT X
380 SCREEN 2,1
420 T1=(N1-M1)/2
430 T2=(N2-M2)/2
440 W=T1/T2
450 IF W<3.21212121# THEN 480
460 XS=319:ZS=319/W
470 GOTO 490
480 XS=199*W:ZS=99
490 FOR Y=A2 TO B2 STEP H2
500 FOR X=A1 TO B1 STEP H1
510 GOSUB 690
520 NEXT X
530 NEXT Y
540 FOR X=A1 TO B1 STEP H3
550 FOR Y=A2 TO B2 STEP H4
560 GOSUB 690
570 NEXT Y
580 NEXT X
590 GOTO 590
610 GOSUB 790
620 XT=X-Y*CS
630 ZT=Z-Y*SI
640 IF XT>N1 THEN N1=XT
650 IF XT<M1 THEN M1=XT
660 IF ZT>N2 THEN N2=ZT
670 IF ZT<M2 THEN M2=ZT
680 RETURN
690 GOSUB 790
700 XT=320+INT(XS*(X-Y*CS-N1+T1)/T1)
710 ZT=100-INT(ZS*(Z-Y*SI-N2+T2)/T2)
720 PSET (XT,ZT)
730 RETURN
790 Z=X+Y
800 RETURN
```

Program 6: Spheri—PC/PCjr Version

```
100 SCREEN 0,0,0:CLS
110 KEY OFF
140 INPUT "Lower Theta limit ";A1
150 INPUT "Upper Theta limit ";B1
160 INPUT "Lower Phi limit ";A2
170 INPUT "Upper Phi limit ";B2
180 INPUT "Slices in Theta ";N
190 INPUT "Slices in Phi ";M
200 INPUT "Observation angle ";Q
210 PRINT "Screen scaling in progress"
220 U=.0174532925#:Q=Q*U
230 CS=COS(Q)
240 SI=SIN(Q)
250 H1=(B1-A1)/639:H2=(B2-A2)/(N-1)
260 H3=(B1-A1)/(M-1):H4=(B2-A2)/639
270 M1=99999999#:M2=M1:N1=-M1:N2=N1
280 FOR Y=A2 TO B2 STEP H2
290 FOR X=A1 TO B1 STEP H1
300 GOSUB 610
310 NEXT X
320 NEXT Y
330 FOR X=A1 TO B1 STEP H3
340 FOR Y=A2 TO B2 STEP H4
350 GOSUB 610
360 NEXT Y
370 NEXT X
380 SCREEN 2,1
420 T1=(N1-M1)/2
430 T2=(N2-M2)/2
440 W=T1/T2
450 IF W<3.21212121# THEN 480
460 XS=319:ZS=319/W
470 GOTO 490
480 XS=199*W:ZS=99
490 FOR Y=A2 TO B2 STEP H2
500 FOR X=A1 TO B1 STEP H1
510 GOSUB 690
520 NEXT X
530 NEXT Y
540 FOR X=A1 TO B1 STEP H3
550 FOR Y=A2 TO B2 STEP H4
560 GOSUB 690
570 NEXT Y
580 NEXT X
590 GOTO 590
610 GOSUB 790
620 XT=XT-YT*CS
630 ZT=ZT-YT*SI
640 IF XT>N1 THEN N1=XT
650 IF XT<M1 THEN M1=XT
660 IF ZT>N2 THEN N2=ZT
670 IF ZT<M2 THEN M2=ZT
680 RETURN
690 GOSUB 790
700 XT=320+INT(XS*(XT-YT*CS-N1+T1)/T1)
710 ZT=100-INT(ZS*(ZT-YT*SI-N2+T2)/T2)
720 PSET (XT,ZT)
730 RETURN
790 REM The function
800 XA=X*U:C1=COS(XA):S1=SIN(XA)
810 YA=Y*U:C2=COS(YA):S2=SIN(YA)
820 XT=(4+C1)*C2
830 YT=(4+C1)*S2
840 ZT=S1
850 RETURN
```

Program 7: Rectan—Apple Version

```
100  HCOLOR= 3
130  HOME
140  INPUT "LOWER X LIMIT:";A1
150  INPUT "UPPER X LIMIT:";B1
160  INPUT "LOWER Y LIMIT:";A2
170  INPUT "UPPER Y LIMIT:";B2
180  INPUT "SLICES IN X:";N
190  INPUT "SLICES IN Y:";M
200  INPUT "OBSERVATION ANGLE:";Q
210  PRINT "SCREEN SCALING IN PROGRESS"
215 U = .0174532925
220 Q = Q * U
230 CS =  COS (Q)
240 SI =  SIN (Q)
250 H1 = (B1 - A1) / 279:H2 = (B2 - A2) / (N
     - 1)
260 H3 = (B1 - A1) / (M - 1):H4 = (B2 - A2) /
     279
270 M1 = 9999999:M2 = M1:N1 =  - M1:N2 = N1
280  FOR Y = A2 TO B2 STEP H2
290  FOR X = A1 TO B1 STEP H1
300  GOSUB 610
310  NEXT
320  NEXT
```

NOW FOR YOUR ATARI,* APPLE* OR COMMODORE 64!*

Super Bunny

Ardy the Aardvark

Cosmic Tunnels

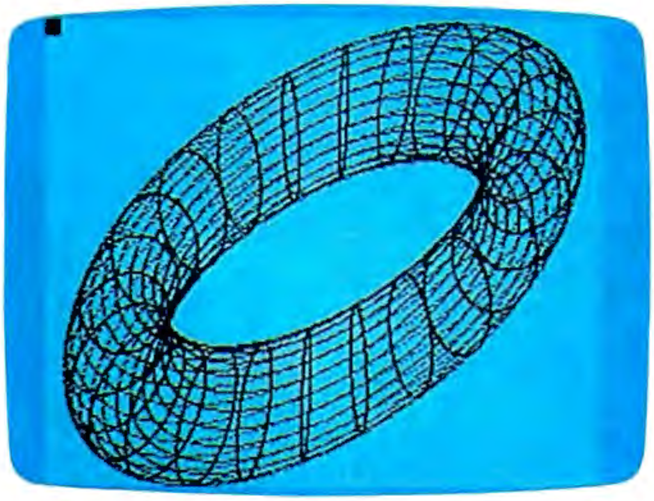
Mr. Robot

Go ahead. Take one for a spin. But be prepared. These games will challenge you like no others. Incredible arcade animation. Never-ending action. And magical music. Your eyes won't believe your ears. Are you up to it? Are you ready to test yourself against the best that DATAMOST has to offer? Go ahead. Take one for a spin.

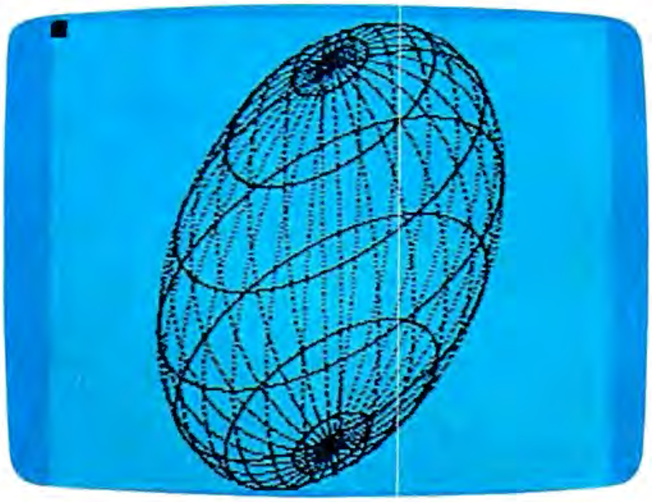
DATAMOST™ INC.

The most out of our minds.™

Datamost, Inc., 20660 Nordhoff St., Chatsworth, CA 91311, (213) 709-1202
*Atari is a trademark of Atari Computer. *Apple is a trademark of Apple Computer.
*Commodore 64 is a trademark of Commodore Business Machines, Inc.



"3-D Plotting" can create spectacular pictures such as this torus or "doughnut."



This sphere was drawn using the "Spheri" program.

```
330  FOR X = A1 TO B1 STEP H3
340  FOR Y = A2 TO B2 STEP H4
350  GOSUB 610
360  NEXT
370  NEXT
380  HGR2
420 T1 = (N1 - M1) / 2
430 T2 = (N2 - M2) / 2
440 W = T1 / T2
450  IF W < 1.46333333 THEN 480
460 XS = 139:ZS = 139 / W
470  GOTO 490
480 XS = 95 * W:ZS = 95
490  FOR Y = A2 TO B2 STEP H2
500  FOR X = A1 TO B1 STEP H1
510  GOSUB 690
520  NEXT
530  NEXT
540  FOR X = A1 TO B1 STEP H3
550  FOR Y = A2 TO B2 STEP H4
560  GOSUB 690
570  NEXT
580  NEXT
600  END
610  GOSUB 790
620 XT = X - Y * CS
630 ZT = Z - Y * SI
640  IF XT > N1 THEN N1 = XT
650  IF XT < M1 THEN M1 = XT
660  IF ZT > N2 THEN N2 = ZT
670  IF ZT < M2 THEN M2 = ZT
680  RETURN
690  GOSUB 790
700 XT = 140 +  INT (XS * (X - Y * CS - N1
     + T1) / T1)
710 ZT = 96 -  INT (ZS * (Z - Y * SI - N2 +
     T2) / T2)
720  HPLOT XT,ZT
780  RETURN
790 Z =  - 1 / (X * X + Y * Y + .5)
800  RETURN
```

Program 8: Spheri—Apple Version

```
100  HCOLOR= 3
130  HOME
140  INPUT "LOWER THETA LIMIT:";A1
150  INPUT "UPPER THETA LIMIT:";B1
160  INPUT "LOWER PHI LIMIT:";A2
170  INPUT "UPPER PHI LIMIT:";B2
180  INPUT "SLICES IN THETA:";N
190  INPUT "SLICES IN PHI:";M
200  INPUT "OBSERVATION ANGLE:";Q
210  PRINT "SCREEN SCALING IN PROGRESS"
215 U = .0174532925
220 Q = Q * U
230 CS =  COS (Q)
240 SI =  SIN (Q)
250 H1 = (B1 - A1) / 279:H2 = (B2 - A2) / (N
     - 1)
260 H3 = (B1 - A1) / (M - 1):H4 = (B2 - A2) /
     279
270 M1 = 99999999:M2 = M1:N1 =  - M1:N2 = N1

280  FOR Y = A2 TO B2 STEP H2
290  FOR X = A1 TO B1 STEP H1
300  GOSUB 610
310  NEXT
320  NEXT
330  FOR X = A1 TO B1 STEP H3
340  FOR Y = A2 TO B2 STEP H4
350  GOSUB 610
360  NEXT
370  NEXT
380  HGR2
420 T1 = (N1 - M1) / 2
430 T2 = (N2 - M2) / 2
440 W = T1 / T2
450  IF W < 1.46333333 THEN 480
460 XS = 139:ZS = 139 / W
470  GOTO 490
480 XS = 95 * W:ZS = 95
490  FOR Y = A2 TO B2 STEP H2
500  FOR X = A1 TO B1 STEP H1
510  GOSUB 690
520  NEXT
530  NEXT
540  FOR X = A1 TO B1 STEP H3
550  FOR Y = A2 TO B2 STEP H4
560  GOSUB 690
570  NEXT
580  NEXT
600  END
610  GOSUB 790
620 XT = XT - YT * CS
630 ZT = ZT - YT * SI
640  IF XT > N1 THEN N1 = XT
650  IF XT < M1 THEN M1 = XT
```

Save New York!™

It was as peaceful a day as New York ever gets, when suddenly the sky went dark and a monstrous droning noise filled the air. Hordes of grotesque aliens were swooping down from all sides, biting into the Big Apple as if they hadn't eaten for days. They were laying eggs, too. Horrible slimy things that got down into the subway tunnels and began clawing their way up. If anyone was going to save the city, it would have to be me. I leapt into my rocket and began blasting away. I thought I stood a fighting chance, but fuel's running low… another wave of invaders on the horizon… signing off…

SAVE NEW YORK.™ For the Commodore 64.

CREATIVE SOFTWARE

```
660  IF ZT > N2 THEN N2 = ZT
670  IF ZT < M2 THEN M2 = ZT
680  RETURN
690  GOSUB 790
700 XT = 140 +  INT (XS * (XT - YT * CS - N
   1  + T1) / T1)
710 ZT = 96 -  INT (ZS * (ZT - YT * SI - N2
   + T2) / T2)
715  IF XT < 0 THEN XT = 0
716  IF XT > 279 THEN XT = 279
720  HPLOT XT,ZT
780  RETURN
790 XA = X * U:C1 =  COS (XA):S1 =  SIN (XA
   )
800 YA = Y * U:C2 =  COS (YA):S2 =  SIN (YA
   )
820 XT = (4 + C1) * C2
830 YT = (4 + C1) * S2
840 ZT = S1
850  RETURN
```

©

PICTURE PERFECT
For Atari And Commodore 64

Coy V Ison

Construct screen art on the Atari with a joystick and basic shapes formed by redefining characters. You also can save a picture to disk or tape for later viewing. The Commodore 64 version, called "Hi-Res Graphics Editor," employs sprites to transport and transform even the most intricate designs.

"Picture Perfect" is not a game that pits you against the computer, but instead is a way to create pictures, patterns, and designs by using the computer and your imagination.

Type in the listing and SAVE a copy, making sure that line 1520 is exactly as shown. When you run the program, you will be prompted for the filename to be used later when saving or loading your picture file. Tape users should enter C: for the filename. Disk users can enter any legal filename, but it must be prefixed with D:. Once you have selected a valid filename, a picture of a castle will be displayed. Press START, and two rows of nine shapes will appear at the bottom of the screen, below the drawing area.

Touch the OPTION key to see two new rows of shapes, and touch OPTION again to toggle back to the first two rows of shapes. These are redefined characters, to be used in your drawings.

Choose A Shape

A question mark will blink on top of the shape to indicate your position. Using a joystick plugged into port 1, you can move across the two rows of redesigned shapes. To pick up one of the shapes, stop on top of it and touch the joystick button. The question mark will then move to the upper right corner of the drawing screen.

You can now place the redesigned shape anywhere on the screen by touching the joystick button. When you want another redesigned shape, touch the SELECT key. This places your cursor on the two rows of shapes so you can pick up another design.

To erase a shape, move the question mark on top of it and touch the space bar. Should you want to erase a large portion of a picture, touch the E key. A red E will replace the question mark on the screen. By holding down on the joystick button and moving the red E, you will be able to quickly erase a large portion of the screen. To stop erasing, simply press the E key again. If you want to erase the whole screen, touch the CLEAR key.

Storing A Picture

To store a picture on tape, first place a tape in the recorder or your disk in the drive and press PLAY and RECORD, then touch the S key on the keyboard. The program will save the picture on tape for you. When using tape, be sure that you press PLAY and RECORD before you touch the S key. No RETURN is necessary and the saving will start immediately.

To save a picture to disk, first insert the disk in the drive and close the door. Then touch the S key.

Loading Your Picture

If you have a picture already stored on a tape or disk and want to load it into the program, you need to have Picture Perfect in the computer. Place your tape into the recorder (or the disk into the disk drive), press PLAY (for cassette) then touch the L key. When the picture is loaded, it

LAST NIGHT, COMPUSERVE TURNED THIS COMPUTER INTO A TRAVEL AGENT FOR JENNIE, A STOCK ANALYST FOR RALPH, AND NOW, IT'S SENDING HERBIE TO ANOTHER GALAXY.

NO MATTER WHICH COMPUTER YOU OWN, WE'LL HELP YOU GET THE MOST OUT OF IT.

If you've got places to go, CompuServe can save you time and money getting there. Just access the Official Airline Guide Electronic Edition—for current flight schedules and fares. Make reservations through our on-line travel service. Even charter a yacht through "Worldwide Exchange."

If your money's in the market, CompuServe offers a wealth of prestigious financial data bases. Access Value Line, or Standard and Poor's. Get the latest information on 40,000 stocks, bonds or commodities. Then, consult experts like IDS or Heinold Commodities. All on line with CompuServe.

Or if, like Herbie, intergalactic gamesmanship is your thing, enjoy the best in fantasy, adventure, and space games. Like MegaWars, the ultimate computer conflict.

To get all this and more, you'll need a computer, a modem and CompuServe. CompuServe connects with almost any personal computer, terminal, or communicating word processor. To receive an illustrated guide to CompuServe and learn how you can subscribe, contact or call:

CompuServe

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

800-848-8199

In Ohio call 614-457-0802

An H&R Block Company

will be displayed. Once again, be sure that your tape is ready and that you have the PLAY button pressed before you touch the L key.

If you don't want to type in the program, I will make copies (for the Atari only) on cassette, if you send the usual $3, a cassette, and a stamped, self-addressed mailer to:

Coy Ison
605 Fifth Ave.
Conway, AR 72032

Program 1: Atari Picture Perfect

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

```
FN 10 GOSUB 2000:DIM SC$(380)
JD 20 GOSUB 1480:GOSUB 970:GOSUB 1260
      :GOTO 130
NC 30 FOR I=0 TO 100 STEP 20
EB 40 SOUND 0,100-I,10,8
ON 50 NEXT I
EG 60 RETURN
HC 70 REM CLEAR SCREEN
FN 80 FOR DO=1 TO 19
EM 90 GOSUB 30
NO 100 POSITION 1,DO:? #6;"
       {18 SPACES}":REM (18 SPACES)
GE 110 NEXT DO
HD 120 RETURN
OG 130 REM JOYSTICK ROUTE FOR RE-SHAP
       ES
DP 140 FOR T=1 TO 100:NEXT T
LH 150 POV=2:PDO=21
PI 160 S=STICK(0):POKE 764,255
PJ 170 DX=(S=7)-(S=11)
CL 180 DY=(S=13)-(S=14)
GJ 190 IF DX<>0 OR DY<>0 THEN GOSUB 3
       0
BE 200 POV=POV+DX+DX:PDO=PDO+DY+DY
EP 210 IF POV<2 THEN POV=18
FC 220 IF POV>18 THEN POV=2
FK 230 IF PDO<21 THEN PDO=23
FN 240 IF PDO>23 THEN PDO=21
MA 250 LOCATE POV,PDO,A
FB 260 POSITION POV,PDO:? #6;"?"
BF 270 FOR T=1 TO 30:NEXT T
LD 280 POSITION POV,PDO:? #6;CHR$(A)
GK 290 IF STRIG(0)=0 THEN 370
JD 300 IF PEEK(53279)=3 THEN GOSUB 13
       70:GOTO 130
KI 310 IF PEEK(53279)=6 THEN DP=1:GOS
       UB 970:GOTO 130
NK 320 IF PEEK(764)=255 THEN 160
KP 330 IF PEEK(764)=62 THEN 780
HC 340 IF PEEK(764)=0 THEN 630
AB 350 IF PEEK(764)=54 THEN GOSUB 70:
       GOTO 130
GJ 360 GOTO 160
FC 370 REM JOYSTICK ROUTE FOR SCREEN
       DRAW
CG 380 FOR T=1 TO 99:NEXT T
FF 390 OV=18:DO=1:POKE 77,0
PF 400 S=STICK(0):POKE 764,255
PG 410 DX=(S=7)-(S=11)
CI 420 DY=(S=13)-(S=14)
GG 430 IF DX<>0 OR DY<>0 THEN GOSUB 3
       0
EL 440 OV=OV+DX:DO=DO+DY
LE 450 IF OV<1 THEN OV=18
LH 460 IF OV>18 THEN OV=1
JD 470 IF DO<1 THEN DO=19
JG 480 IF DO>19 THEN DO=1
DE 490 LOCATE OV,DO,Q
PO 500 POSITION OV,DO:? #6;"?"
GE 505 POSITION POV,PDO:? #6;"  "
BA 510 FOR T=1 TO 10:NEXT T
CA 520 POSITION OV,DO:? #6;CHR$(Q)
LF 525 POSITION POV,PDO:? #6;CHR$(A)
EB 530 IF STRIG(0)=0 THEN POSITION OV
       ,DO:? #6:CHR$(A)
JJ 540 IF PEEK(53279)=3 THEN GOSUB 13
       70:GOTO 130
NO 550 IF PEEK(53279)=5 THEN 130
NN 560 IF PEEK(764)=255 THEN 400
OD 565 IF PEEK(764)=42 THEN 1235
LF 570 IF PEEK(764)=62 THEN 780
DP 580 IF PEEK(764)=33 THEN POSITION
       OV,DO:? #6;"  "
HJ 590 IF PEEK(764)=0 THEN 630
PP 600 IF PEEK(764)=54 THEN GOSUB 70:
       GOTO 130
NL 610 IF PEEK(53279)=5 THEN 130
GF 620 GOTO 400
LP 630 REM LOADING DATA TAPE
NF 640 SC$=""
EC 650 POSITION 1,20:? #6;"LOADING DA
       TA TAPE"
JO 660 FN=1
FC 670 OPEN #4,4,0,FILE$
EC 680 GET #4,A
BN 690 IF A=63 THEN CLOSE #4:GOTO 720
LI 700 SC$(LEN(SC$)+1)=CHR$(A)
GP 710 GOTO 680
JH 720 FOR LP=1 TO 19
AH 730 POSITION 1,LP:? #6;SC$(FN,FN+1
       7)
JE 740 FN=FN+18
HH 750 NEXT LP
ND 760 POSITION 1,20:? #6;"███████████
       ███████"
GL 770 GOTO 130
IP 780 REM SAVING DATA TAPE
NL 790 SC$=""
CY 800 POSITION 2,20:? #6;"SAVING DAT
       A TAPE"
IO 810 FOR DO=1 TO 19
KA 820 FOR OV=1 TO 18
GF 830 LOCATE OV,DO,ZZ:SC$(LEN(SC$)+1
       )=CHR$(ZZ)
AF 840 POSITION OV,DO:? #6:"?"
IJ 850 POSITION OV,DO:? #6;CHR$(ZZ)
IC 860 NEXT OV
HB 870 NEXT DO
FJ 880 OPEN #4,8,0,FILE$
BP 890 FOR LP=1 TO LEN(SC$)
KM 900 PUT #4,ASC(SC$(LP,LP))
HF 910 NEXT LP
IA 920 PUT #4,63
GJ 930 CLOSE #4
DD 940 POSITION 2,20:? #6;"███████████
       ███████"
GL 950 GOTO 130
GN 960 REM DRAW CASTLE
```

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

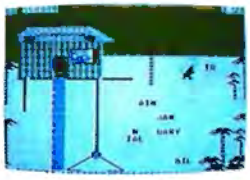
For help with spelling and vocabulary, choose *Monkeybuilder*™. It encourages the child to combine word pieces correctly to form building blocks, and make a tree house for Marc.

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.

```
PE 970  POSITION 1,1:? #6;"{18 SPACES}"
KO 980  POSITION 1,2:? #6;"{11 SPACES}\
        {3 SPACES}\   "
HD 990  POSITION 1,3:? #6;"{5 SPACES}*
        {7 SPACES}\{4 SPACES}"
BI 1000 POSITION 1,4:? #6;"
        {18 SPACES}"
BK 1010 POSITION 1,5:? #6;"
        {18 SPACES}"
LF 1020 POSITION 1,6:? #6;"
        {5 SPACES}) #$  ){7 SPACES}"
EM 1030 POSITION 1,7:? #6;"   #$ C CC
        C #${4 SPACES}"
HI 1040 POSITION 1,8:? #6;"99CC9CCCCC
        C9CC9999"
LI 1050 POSITION 1,9:? #6;"   CC C+CC+
        C CC{4 SPACES}"
CN 1060 POSITION 1,10:? #6;"   CC)CCCC
        CC)CC{4 SPACES}"
HY 1070 POSITION 1,11:? #6;"   +CCCCCC
        CCCC+ #$  "
IH 1080 POSITION 1,12:? #6;"   CCCCCCC
        CCCCC ,,  "
OH 1090 POSITION 1,13:? #6;"   CC CCCC
        CC CC{4 SPACES}"
JO 1100 POSITION 1,14:? #6;"   C+ CC^^
        CC +C))   "
OH 1110 POSITION 1,15:? #6;"   CC CC^^
        CC CC;;)  "
EN 1120 POSITION 1,16:? #6;" )CC CC^^
        CC CC;;;)"
HL 1130 POSITION 1,17:? #6;") ■■■■■■■
        ■■■■■ :;  "
FC 1140 POSITION 1,18:? #6;"
        {18 SPACES}"
FE 1150 POSITION 1,19:? #6;"
        {18 SPACES}"
KD 1160 IF DP=1 THEN DP=0:RETURN
BE 1170 GOSUB 1280
AF 1180 POSITION 3,1:? #6;"picture pe
        rfect"
HA 1190 POSITION 4,22:? #6;"PRESS STA
        RT"
DP 1200 FOR T=1 TO 30:NEXT T
EJ 1210 POSITION 4,22:? #6;"press sta
        rt"
EB 1220 FOR T=1 TO 30:NEXT T
DK 1230 IF PEEK(53279)=6 THEN POSITIO
        N 2,22:? #6;"{13 SPACES}":GOSU
        B 70:RETURN
ML 1231 GOTO 1190
GJ 1232 REM E KEY ROUTE
HJ 1235 EOV=OV:EDO=DO
CN 1236 S=STICK(0):POKE 764,255
PK 1237 DX=(S=7)-(S=11):DY=(S=13)-(S=
        14)
JF 1238 EOV=EOV+DX:EDO=EDO+DY
HE 1239 IF EOV<1 THEN EOV=18
GO 1240 IF EOV>18 THEN EOV=1
EK 1241 IF EDO<1 THEN EDO=19
EH 1242 IF EDO>19 THEN EDO=1
CN 1243 LOCATE EOV,EDO,EL
GE 1244 POSITION EOV,EDO:? #6;"E"
EH 1245 FOR T=1 TO 20:NEXT T
CA 1246 POSITION EOV,EDO:? #6;CHR$(EL
        )
LH 1247 IF STRIG(0)=0 THEN POSITION E
        OV,EDO:? #6;" "
NM 1248 IF PEEK(764)=42 THEN 140
NF 1249 GOTO 1236
AG 1259 REM SET UP SCREEN
AA 1260 MN=1:CC=1
BG 1270 GOSUB 1410
MI 1280 FOR LP=0 TO 19
CL 1290 POSITION LP,0:? #6;"■"
FF 1300 POSITION LP,20:? #6;"■"
KA 1310 NEXT LP
ME 1320 FOR LP=1 TO 19
CG 1330 POSITION 0,LP:? #6;"■"
GP 1340 POSITION 19,LP:? #6;"■"
KE 1350 NEXT LP
KY 1360 RETURN
JO 1370 REM FLIP SHAPES
FD 1380 CC=CC+1:IF CC>2 THEN CC=1
OE 1390 ON CC GOTO 1410,1440
ML 1400 GOTO 1390
FJ 1410 POSITION 2,21:? #6;"! # $ ^ C
         ' ( ) &"
MN 1420 POSITION 2,23:? #6;"■ + , - ]
         \ : ; *"
KI 1430 RETURN
KG 1440 REM SECOND SET OF SHAPES
DC 1450 POSITION 2,21:? #6;"0 1 2 3 [
         > % . /"
ME 1460 POSITION 2,23:? #6;"E 8 9 < =
         6 5 7 4"
FM 1470 RETURN
BF 1480 REM FAST DUMP
OC 1490 GRAPHICS 1+16
PP 1500 POSITION 4,10:? #6;"ONE MOMEN
        T"
KI 1510 DIM E$(50):RAMTOP=PEEK(106)-8
        :POKE 106,RAMTOP:CHBAS=RAMTOP
        :ADDR=CHBAS*256:PAGE=4
CP 1520 FOR I=1 TO 41:READ UM:E$(I,I)
        =CHR$(UM):NEXT I:A=USR(ADR(E$
        ),ADDR,PAGE)
EF 1521 DATA 104,104,133,207,104,133
EO 1522 DATA 206,104,104,133,212,169
ON 1523 DATA 0,133,204,169,224,133
IJ 1524 DATA 205,162,1,160,0,177
FG 1525 DATA 204,145,206,200,208,249
FB 1526 DATA 230,205,230,207,232,228
PN 1527 DATA 212,208,240,96,0
MD 1530 FOR LP=1 TO 33
AE 1540 READ CHAR
BB 1550 POS=ADDR+(CHAR*8)
FG 1560 FOR X=0 TO 7:READ A:POKE (POS
        +X),A:NEXT X
KI 1570 NEXT LP
FF 1580 POSITION 4,10:? #6;"
        {10 SPACES}"
FK 1590 DATA 1,255,255,255,255,255,25
        5,255,255
HI 1600 DATA 3,1,3,7,15,31,63,127,255
EJ 1610 DATA 4,128,192,224,240,248,25
        2,254,255
BE 1620 DATA 5,24,24,24,255,255,24,24
        ,24
LG 1630 DATA 6,60,36,60,36,60,36,60,3
        6
KM 1640 DATA 7,0,255,255,0,0,255,255,
        0
GC 1650 DATA 8,222,222,0,123,123,0,22
        2,222
IC 1660 DATA 9,24,24,60,60,126,126,25
        5,255
FH 1670 DATA 10,66,165,90,60,60,90,16
        5,66
IL 1680 DATA 11,255,129,129,129,129,1
        29,129,255
```

Strengthen your hand with Superbase 64 T.M.

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:

Superbase 64 T.M.

Precision Software

Precision Software (USA), Inc.
Suite 11D
1675 York Avenue
NEW YORK
N.Y. 10128
(212) 410 3418

OFFICIALLY APPROVED BY COMMODORE

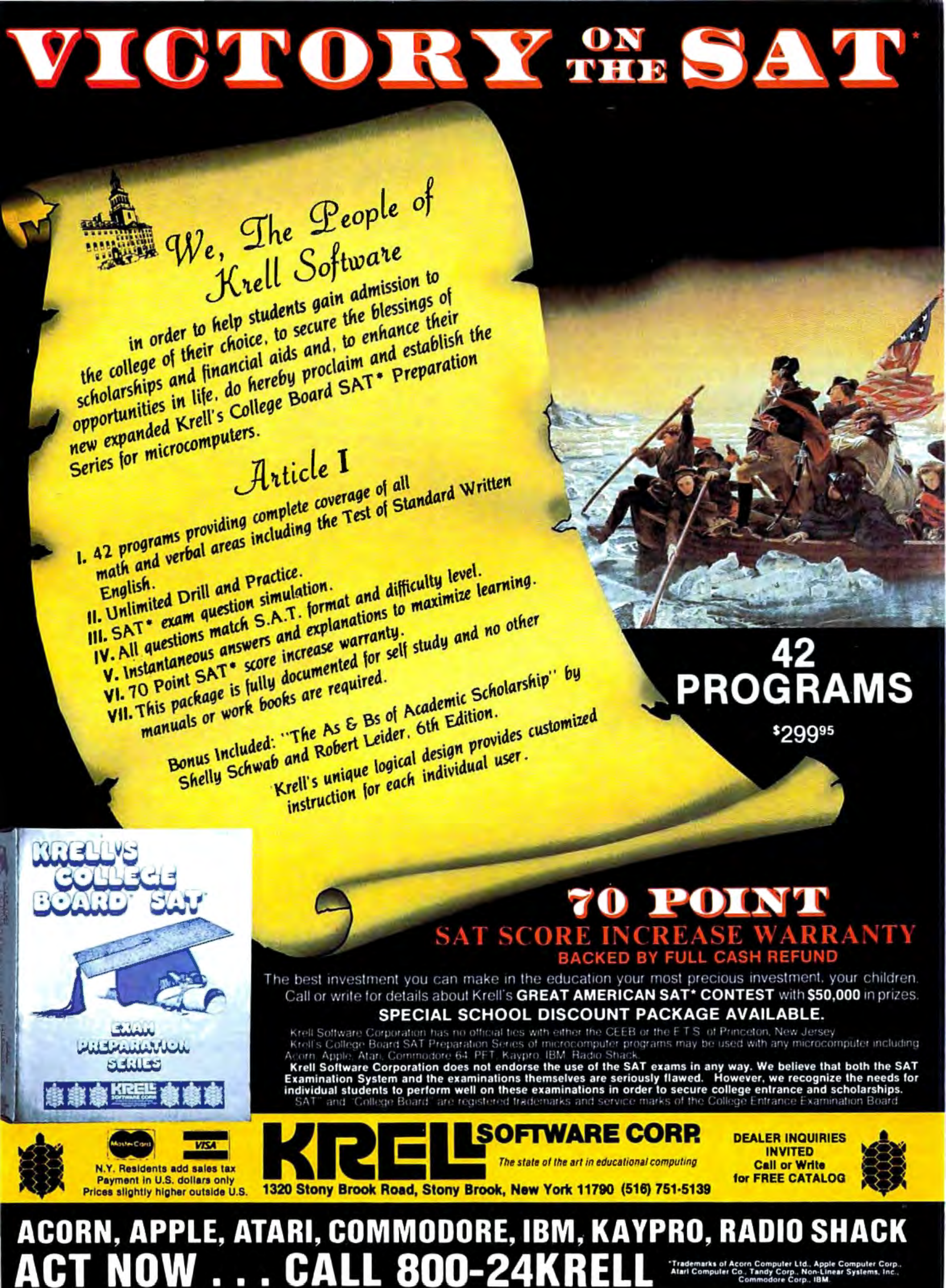
commodore

```
AG 1690 DATA 12,215,0,190,0,221,0,60,
        129
KI 1700 DATA 13,24,60,126,255,255,126
        ,60,24
NI 1710 DATA 14,24,24,24,24,24,24,24,
        24
AC 1720 DATA 15,0,0,0,255,255,0,0,0
JD 1730 DATA 26,255,255,255,255,255,2
        55,249,249
ED 1740 DATA 27,0,0,0,0,34,170,85,35
IH 1750 REM SECOND DATA
BN 1760 DATA 16,60,126,255,255,255,25
        5,126,60
CI 1770 DATA 17,60,255,126,219,126,36
        ,255,189
PN 1780 DATA 18,60,90,126,165,24,255,
        189,189
BE 1790 DATA 19,189,189,60,60,102,102
        ,102,231
FD 1800 DATA 20,24,24,24,248,248,0,0,
        0
KF 1810 DATA 21,0,0,0,248,248,24,24,2
        4
DD 1820 DATA 22,0,0,0,31,31,24,24,24
DF 1830 DATA 23,24,24,24,31,31,0,0,0
KH 1840 DATA 24,24,24,24,255,255,0,0,
        0
KJ 1850 DATA 25,0,0,0,255,255,24,24,2
        4
NP 1860 DATA 28,24,24,24,31,31,24,24,
        24
FF 1870 DATA 29,24,24,24,248,248,24,2
        4,24
II 1880 DATA 30,1,3,6,12,24,48,96,192
PO 1890 DATA 59,128,192,96,48,24,12,6
        ,3
AI 1900 DATA 60,0,0,0,36,90,129,0,0
NK 1910 DATA 61,24,28,30,31,31,30,28,
        24
MC 1920 DATA 62,170,85,170,85,170,85,
        170,85
CL 1930 POKE 756,CHBAS
KO 1940 RETURN
FC 2000 DIM FILE$(15):GRAPHICS 0
DI 2010 TRAP 2060:PRINT "{CLEAR}
        {DOWN}INPUT FILENAME"
JL 2020 PRINT "{DOWN}CASSETTE USERS E
        NTER C:"
OJ 2030 PRINT "{DOWN}DISK USERS ENTER
         FILENAME WITH D:"
JK 2040 INPUT FILE$
LA 2050 IF FILE$(1,2)="C:" OR FILE$(1
        ,2)="D:" THEN TRAP 40000:RETU
        RN
CG 2060 TRAP 2060:PRINT "{BELL}{DOWN}
        ERROR IN FILENAME!":FOR UM=1
        TO 200:NEXT UM:GOTO 2010
```

Program 2:
Machine Language For Hi-Res Graphics Editor

(Use MLX to enter this program.)

```
49152 :032,107,198,169,015,141,150
49158 :226,206,032,013,198,169,082
49164 :128,133,044,141,130,002,078
49170 :169,000,141,000,128,169,113
49176 :200,141,000,208,141,254,200
49182 :206,169,003,141,021,208,010
49188 :169,033,141,212,205,169,197
49194 :000,141,016,208,141,255,035
49200 :206,169,100,141,001,208,105
49206 :141,003,208,173,024,208,043
49212 :041,240,009,008,141,024,011
49218 :208,173,017,208,009,032,201
49224 :141,017,208,169,000,141,236
49230 :238,002,032,182,200,032,252
49236 :107,192,032,004,194,032,133
49242 :186,197,032,239,197,032,205
49248 :186,199,032,008,201,173,127
49254 :238,002,240,230,096,169,053
49260 :032,141,248,007,169,001,194
49266 :141,039,208,238,040,208,220
49272 :173,227,205,201,003,208,113
49278 :018,169,076,141,198,205,165
49284 :169,248,141,197,205,169,237
49290 :014,141,241,002,076,160,004
49296 :192,169,063,141,198,205,088
49302 :169,228,141,197,205,169,235
49308 :025,141,241,002,173,212,182
49314 :205,141,249,007,173,000,169
49320 :220,041,015,141,253,206,020
49326 :056,169,015,237,253,206,086
49332 :141,252,206,160,000,200,115
49338 :204,252,206,208,250,152,178
49344 :010,168,185,204,192,072,255
49350 :185,203,192,072,096,002,180
49356 :194,214,193,218,193,002,194
49362 :194,226,193,230,193,237,203
49368 :193,002,194,222,193,251,247
49374 :193,244,193,002,194,169,193
49380 :050,205,001,208,176,012,112
49386 :173,001,208,056,173,001,078
49392 :208,233,001,141,001,208,008
49398 :096,173,197,205,205,001,099
49404 :208,144,012,173,001,208,230
49410 :024,173,001,208,105,001,002
49416 :141,001,208,096,056,173,171
49422 :254,206,237,198,205,141,231
49428 :253,206,173,255,206,233,066
49434 :001,013,253,206,144,014,145
49440 :173,198,205,141,254,206,185
49446 :169,001,141,255,206,076,118
49452 :063,193,024,173,254,206,189
49458 :105,001,141,254,206,173,162
49464 :255,206,105,000,141,255,250
49470 :206,056,173,254,206,233,166
49476 :000,141,253,206,173,255,072
49482 :206,233,001,013,253,206,218
49488 :144,015,173,016,208,009,133
49494 :001,141,016,208,173,254,111
49500 :206,141,000,208,096,173,148
49506 :016,208,041,254,141,016,006
49512 :208,173,254,206,141,000,062
49518 :208,096,056,173,254,206,079
49524 :237,241,002,141,253,206,172
49530 :173,255,206,233,000,013,234
49536 :253,206,176,017,056,173,241
49542 :241,002,233,001,141,254,238
49548 :206,169,000,141,255,206,093
49554 :076,166,193,056,173,254,040
49560 :206,233,001,141,254,206,169
49566 :173,255,206,233,000,141,142
49572 :255,206,056,173,254,206,034
49578 :233,000,141,253,206,173,152
49584 :255,206,233,001,013,253,113
49590 :206,144,015,173,016,208,176
49596 :009,001,141,016,208,173,224
49602 :254,206,141,000,208,096,075
49608 :173,016,208,041,254,141,009
49614 :016,208,173,254,206,141,180
49620 :000,208,096,032,227,192,199
```

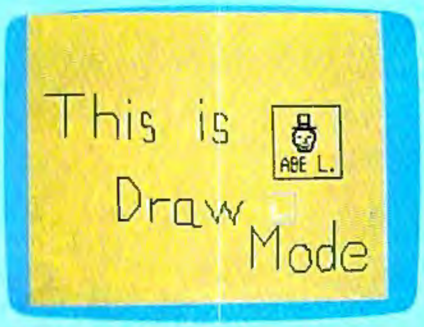


VICTORY ON THE SAT*
We, The People of Krell Software
in order to help students gain admission to the college of their choice, to secure the blessings of scholarships and financial aids and, to enhance their opportunities in life, do hereby proclaim and establish the new expanded Krell's College Board SAT* Preparation Series for microcomputers.
Article I
I. 42 programs providing complete coverage of all math and verbal areas including the Test of Standard Written English.
II. Unlimited Drill and Practice.
III. SAT* exam question simulation.
IV. All questions match S.A.T. format and difficulty level.
V. Instantaneous answers and explanations to maximize learning.
VI. 70 Point SAT* score increase warranty.
VII. This package is fully documented for self study and no other manuals or work books are required.
Bonus Included: "The As & Bs of Academic Scholarship" by Shelly Schwab and Robert Leider, 6th Edition.
Krell's unique logical design provides customized instruction for each individual user.
42 PROGRAMS
$299.95
KRELL'S COLLEGE BOARD SAT
EXAM PREPARATION SERIES
KRELL
70 POINT
SAT SCORE INCREASE WARRANTY
BACKED BY FULL CASH REFUND
The best investment you can make in the education your most precious investment, your children.
Call or write for details about Krell's GREAT AMERICAN SAT* CONTEST with $50,000 in prizes.
SPECIAL SCHOOL DISCOUNT PACKAGE AVAILABLE.
Krell Software Corporation has no official ties with either the CEEB or the E.T.S. of Princeton, New Jersey.
Krell's College Board SAT Preparation Series of microcomputer programs may be used with any microcomputer including Acorn, Apple, Atari, Commodore 64, PET, Kaypro, IBM, Radio Shack.
Krell Software Corporation does not endorse the use of the SAT exams in any way. We believe that both the SAT Examination System and the examinations themselves are seriously flawed. However, we recognize the needs for individual students to perform well on these examinations in order to secure college entrance and scholarships.
"SAT" and "College Board" are registered trademarks and service marks of the College Entrance Examination Board.
MasterCard
VISA
N.Y. Residents add sales tax
Payment in U.S. dollars only
Prices slightly higher outside U.S.
KRELL SOFTWARE CORP.
The state of the art in educational computing
1320 Stony Brook Road, Stony Brook, New York 11790 (516) 751-5139
DEALER INQUIRIES INVITED
Call or Write for FREE CATALOG
ACORN, APPLE, ATARI, COMMODORE, IBM, KAYPRO, RADIO SHACK
ACT NOW . . . CALL 800-24KRELL
*Trademarks of Acorn Computer Ltd., Apple Computer Corp., Atari Computer Co., Tandy Corp., Non-Linear Systems, Inc., Commodore Corp., IBM.

64 Hi-Res Graphics Editor

49152--51553

Gregg Peele, Assistant Programming Supervisor



Just as a word processor allows you to expand your writing skills by giving you power to manipulate text freely, "Hi-Res Graphics Editor" allows you to easily draw, erase, and edit images on the 64's hi-res screen. Once you have finished your drawing, you can even send the results to your 1525 printer.

The Editor expands on the graphics techniques in "Picture Perfect," using the sprite capability of your 64 to create and modify intricate designs on the screen. Parts of pictures can be "imprinted" onto a sprite and "planted" on another area of the screen. You can then enlarge the sprite to full-screen size and edit it more precisely.

Type It In With MLX

Hi-Res Graphics Editor is in two parts. First you must type in Program 2 using the MLX program elsewhere in this issue. After saving Program 2 to disk or tape, reset your machine by turning it off.

Now type in Program 3, the BASIC part of Hi-Res Graphics Editor. SAVE it to disk or tape.

To run the program, first LOAD the file created by MLX with this format:

```
LOAD "your filename",8,1   for disk
LOAD "your filename",1,1   for tape
```

Now enter this line and press RETURN:

```
POKE 642,128:POKE 44,128:POKE 32768,0:NEW
```

This moves BASIC to a safe place in memory—leaving plenty of room for hi-res screens. You must type this line each time before you LOAD Program 3.

Next, LOAD the BASIC program—Program 3. Type RUN, press RETURN, and you are in the Editor.

Set The Joystick Speed

The first prompt in Hi-Res Graphics Editor is for joystick speed. Enter a number from 1 to 10 (10 is fastest). The lower the number, the more control you have over drawing. You can experiment with these numbers to find the best speed for your purposes.

Next, the screen clears and a rectangle appears in the center. This is the sprite cursor. Press the letter D and the box will change into an arrow. You are now in Draw Mode. With a joystick in port 2, you can move this arrow around the screen.

Pressing the fire button draws on the screen. If what you have drawn is invisible, press B to change the background color and F to change the foreground color. Repeat each of these keys to step through the sequence of all possible colors.

Erasing With The Arrow

If you wish to erase what you have drawn, engage the SHIFT LOCK key on the keyboard. Then hold down the fire button and use the joystick to point the arrow at any pixel you want to erase. To start over with a clean slate, just press the f1 key. This clears the screen.

Sprite Mode can be accessed by pressing the A (Add), S (Stamp), C (Copy), or E (Erase) key. Let's explore the most interesting of these, hitting the letter C.

Using the joystick, move the rectangle around the screen until it's superimposed on part of your original drawing. (If you have cleared the screen, you can return to draw mode by pressing D.) Press the fire button, and the contents of the screen "under" the sprite will be copied onto the sprite.

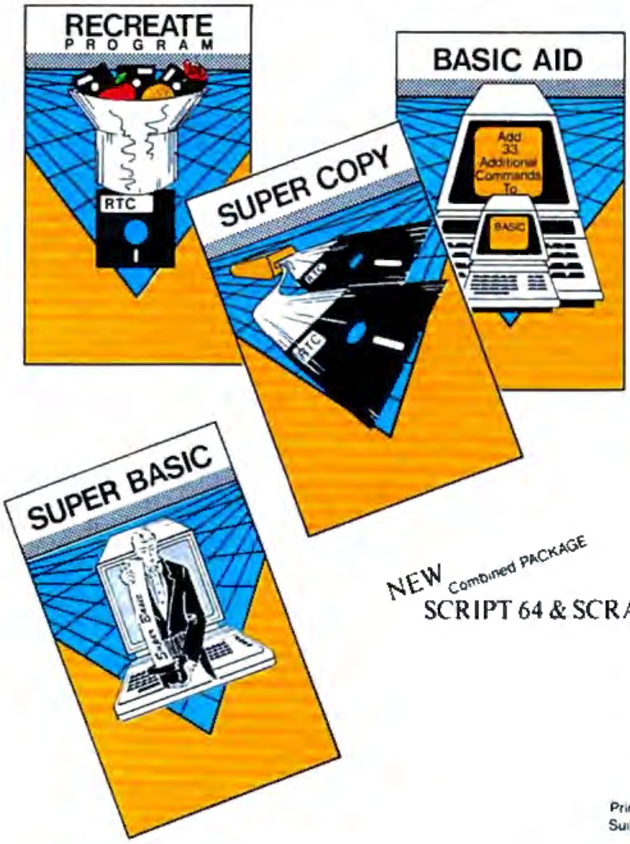
You can enter Add Mode at any time by pressing A. In this mode, you can move your sprite around the screen and "plant" the image anywhere you like. (You *add* the image of the sprite to the images already on the screen.) If you hold the button down while you move the sprite, the sprite's image becomes a wide "brush," which you can use for calligraphy and to create other interesting effects.

A Graphic Stamp

Stamp Mode replaces the contents of the screen with the contents of the sprite. And if you make a mistake in your drawing, use E, Erase Mode. This mode transforms the sprite cursor into a giant eraser which clears any pixels it passes over.

Developing a mind for the Future.

RTC



SUPER BASIC $39[95] U.S.*

- Gives you 3 different versions of Commodore Basic Programing Language Version 4 PLUS!
- A Built in Machine Language Monitor!
- Disk & File Maintenance Commands
- Data Handling Commands
- Graphics Plus Basic
- Compatible with Commmodore's "B" Series & Much Much More!

RECREATE PROGRAM $39[95] U.S.*

RTC's Answers to Program Recreation
Converts Printer's File to SCRIPT 64's Files

SUPER COPY $29[95] U.S.*

Super Fast Disk Copies on a 1541
Copy Entire Disk in 7 Minutes or Less
Copy Selected Files
Complete Pattern Matching
Full Prompts

BASIC AID $39[95] U.S.*

Your Aid to Writing Programs
Allows Scrolling Through Programs
Adds 33 more Commands to the Basic Language
Has Find, Change, Merge, Move Commands
Convert Hex, Binary and Decimal Numbers and More!

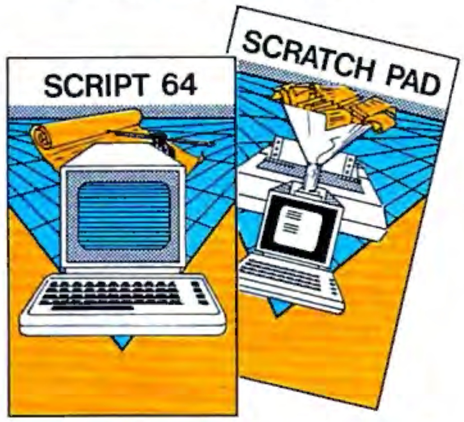
Color 80 $49[95] U.S.*

Produces 80 Columns With Color!
Needs No Extra Hardware
Use With Terminal Programs
Uses No Basic Memory in the 64
Develop Your Own Programs

NEW Combined PACKAGE

SCRIPT 64 & SCRATCH PAD 64 $105[00] U.S.*

Script 64
Word Processor in French and English
80 Columns
Global Search and Replace
User Created Dictionary
Spelling Check
Scratch Pad 64
The Database/Mail List in One!
Merges with Script 64 Word Processor
Print out Labels, Envelopes, Mail List & More!
Suitable with both Single and Dual Disk Drives
Fully C64 Link Compatible



*SUGGESTED RETAIL PRICE F.O.B. in U.S. FUNDS Toronto, CANADA

C64 LINK©

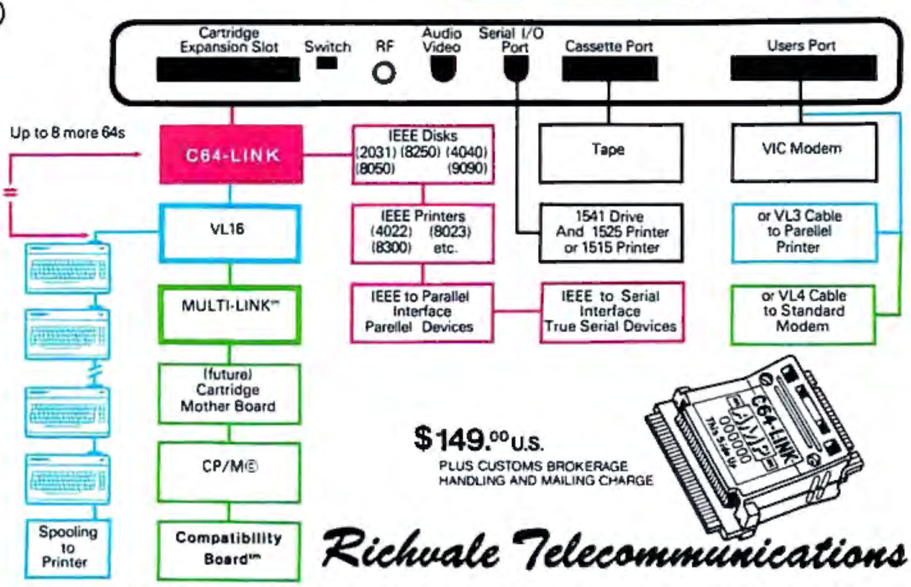
The Smart 64

Give These Expanded Capabilities to Your 64 and VIC 20

- The ability to transfer data from any type of device to another (IEEE, Serial, Parallel)
- BASIC 4.0 which allows you to run more PET BASIC programs and gives you extended disk and I/O commands.
- The ability to have several 64s on line together - sharing common IEEE devices such as disks or printers with Spooling Capability.
- Built-in machine language monitor.
- A built-in terminal or modem program which allows the system to communicate through a modem to many bulletin board systems and other computer mainframes.
- Compatability with CP/M.

Contact your local Commodore dealer or RTC.

Payments by VISA, MASTERCARD or BANK TRANSFER.
Mail orders also by certified cheque, etc.



$149.[00] U.S.

PLUS CUSTOMS BROKERAGE HANDLING AND MAILING CHARGE

Richvale Telecommunications

10610 Bayview Avenue (Bayview Plaza) Richmond Hill, Ontario, Canada L4C 3N8 (416) 884-4165

A Sprite Editor

You can create your own sprites by enlarging the sprite to full screen proportions. Hold down the f7 key briefly. The screen will clear and an enlarged image of the sprite will appear in the upper left corner of the screen. To edit this sprite, press the fire button of the joystick as you move the cursor in this area. Erasing is simple. Just engage the SHIFT/LOCK key, and instead of drawing to the sprite image, you will erase parts of the sprite. The f1 key clears the sprite, just as it cleared the screen in hi-res mode.

If you want to save or load a hi-res screen, you must do it from this sprite definition mode. Hold the CTRL key while you press L for LOAD, and a series of prompts will then appear for loading from disk or tape. Likewise, holding CTRL and S allows you to save to disk or tape.

Anytime you wish to return to hi-res mode, simply hold f7 down for a moment. You can then use the sprite definition you have just created to produce intricate pictures on the hi-res screen.

Two Graphics Screens

The Editor contains a feature which allows you to have two full screens of graphics in memory at one time. Press T to toggle between them. When you first try this function, the screen will fill with garbage if nothing has been created on the alternate screen. (There is undefined data in this area.)

Clear the screen (using the f1 key) to start with a new palette. Draw a new design on this screen, and press T (toggle) to return to the old screen. Pressing T again takes you back to your second creation, and so on.

Printing Your Creation

Since an image created on a computer screen will last only as long as the power is on, a hi-res screen dump is included. Just press the letter P, and your 1525 printer (or 1525 compatible printer) will print the contents (minus the sprite cursors) of the screen.

Here's a summary of the commands in the Hi-Res Graphics Editor:

D	Draw Mode
SHIFT LOCK on	Erase draw (in sprite definition mode, erase parts of sprite)
A	Add Mode; overlay sprite with screen
C	Copy screen to sprite
S	Stamp Mode; replace what is onscreen with sprite image
E	Erase under sprite
F	Sequence through foreground colors
B	Sequence through background colors
T	Toggle between screens
f1	Clear screen (hi-res and sprite definition modes)
f7	Change from hi-res to sprite definition and vice versa
CTRL-L	Load screen from disk or tape; available only from sprite definition mode
CTRL-S	Save screen from disk or tape; available only from sprite definition mode
P	Produce printout on 1525 printer

```
49626 :096,032,247,192,096,032,145
49632 :012,193,096,032,112,193,094
49638 :096,032,227,192,032,112,153
49644 :193,096,032,247,192,032,004
49650 :112,193,096,032,247,192,090
49656 :032,012,193,096,032,227,072
49662 :192,032,012,193,096,096,107
49668 :173,001,208,141,003,208,226
49674 :173,000,208,141,002,208,230
49680 :173,016,208,041,001,240,183
49686 :011,169,002,013,016,208,185
49692 :141,016,208,076,042,194,193
49698 :169,253,045,016,208,141,098
49704 :016,208,056,173,254,206,185
49710 :233,024,141,250,206,173,049
49716 :255,206,233,000,141,251,114
49722 :206,165,197,201,013,240,056
49728 :023,201,010,240,030,201,001
49734 :014,240,046,201,018,240,061
49740 :053,201,020,240,079,201,102
49746 :003,240,025,076,168,194,020
49752 :169,000,141,227,205,032,094
49758 :138,194,076,168,194,169,009
49764 :001,141,227,205,032,138,076
49770 :194,076,168,194,032,138,140
49776 :194,076,180,199,076,168,237
49782 :194,169,002,141,227,205,032
49788 :032,138,194,076,168,194,158
49794 :169,003,141,227,205,076,183
49800 :168,194,169,172,141,000,212
49806 :208,141,254,206,169,000,096
49812 :141,016,208,141,255,206,091
49818 :169,124,141,001,208,096,125
49824 :169,004,141,227,205,032,170
49830 :138,194,173,227,205,201,024
49836 :003,208,016,169,034,141,231
49842 :212,205,173,021,208,041,014
49848 :254,141,021,208,076,204,064
49854 :194,169,033,141,212,205,120
49860 :173,021,208,009,003,141,239
49866 :021,208,056,173,001,208,101
49872 :233,050,141,248,206,173,235
49878 :000,220,041,016,208,017,204
49884 :169,000,141,224,206,162,098
49890 :000,173,227,205,201,004,012
49896 :208,006,076,243,194,076,011
49902 :018,196,076,125,195,173,253
49908 :250,206,141,218,205,173,157
49914 :251,206,141,219,205,169,161
49920 :128,141,216,205,169,000,091
```

```
49926 :168,170,141,214,205,142,022
49932 :222,205,140,221,205,032,013
49938 :022,196,174,222,205,172,241
49944 :221,205,173,224,205,045,073
49950 :206,207,240,012,173,216,060
49956 :205,025,000,008,153,000,171
49962 :008,076,057,195,173,216,255
49968 :205,073,255,057,000,008,134
49974 :153,000,008,078,216,205,202
49980 :208,006,169,128,141,216,160
49986 :205,200,024,173,250,206,100
49992 :105,001,141,250,206,173,180
49998 :251,206,105,000,141,251,008
50004 :206,232,224,024,208,177,131
50010 :162,000,173,218,205,141,221
50016 :250,206,173,219,205,141,010
50022 :251,206,238,248,206,162,133
50028 :000,238,214,205,173,214,128
50034 :205,201,021,144,148,169,234
50040 :001,141,227,205,096,169,191
50046 :128,141,226,206,172,224,199
50052 :206,185,000,008,045,226,034
50058 :206,240,008,169,001,141,135
50064 :228,206,076,157,195,169,151
50070 :000,141,228,206,076,157,190
50076 :195,173,227,205,201,003,136
50082 :208,039,173,141,002,208,165
50088 :008,169,001,141,228,206,153
50094 :076,182,195,169,000,141,169
50100 :228,206,024,173,250,206,243
50106 :105,011,141,250,206,173,048
50112 :251,206,105,000,141,251,122
50118 :206,032,022,196,096,142,124
50124 :216,206,032,022,196,174,026
50130 :216,206,024,173,250,206,005
50136 :105,001,141,250,206,173,068
50142 :251,206,105,000,141,251,152
50148 :206,110,226,206,208,152,056
50154 :238,224,206,232,224,003,081
50160 :240,003,076,125,195,162,017
50166 :000,238,248,206,056,173,143
50172 :250,206,233,024,141,250,076
50178 :206,173,251,206,233,000,047
50184 :141,251,206,172,224,206,184
50190 :192,063,144,001,096,076,074
50196 :125,195,173,250,206,141,086
50202 :250,207,173,251,206,141,230
50208 :251,207,173,248,206,141,234
50214 :248,207,169,000,141,249,028
50220 :207,173,250,207,141,212,210
50226 :207,173,251,207,141,213,218
50232 :207,173,248,207,141,214,222
50238 :207,173,249,207,141,215,230
50244 :207,173,215,207,074,141,061
50250 :217,207,173,214,207,106,174
50256 :141,216,207,173,217,207,217
50262 :074,141,217,207,173,216,090
50268 :207,106,141,216,207,173,118
50274 :217,207,074,141,217,207,137
50280 :173,216,207,106,141,216,139
50286 :207,173,213,207,074,141,101
50292 :219,207,173,212,207,106,216
50298 :141,218,207,173,219,207,007
50304 :074,141,219,207,173,218,136
50310 :207,106,141,218,207,173,162
50316 :219,207,074,141,219,207,183
50322 :173,218,207,106,141,218,185
50328 :207,173,214,207,041,007,233
50334 :141,220,207,173,216,207,042
50340 :010,046,217,207,010,046,188
50346 :217,207,010,141,210,207,138
50352 :046,217,207,173,217,207,219
50358 :141,211,207,173,210,207,051
50364 :010,046,217,207,010,046,212
50370 :217,207,109,210,207,141,005
50376 :216,207,173,211,207,109,043
50382 :217,207,141,217,207,173,088
50388 :216,207,010,046,217,207,091
50394 :010,046,217,207,010,046,242
50400 :217,207,141,216,207,173,105
50406 :218,207,010,046,219,207,113
50412 :010,046,219,207,010,046,006
50418 :219,207,141,218,207,024,234
50424 :173,216,207,109,218,207,098
50430 :141,208,207,173,217,207,127
50436 :109,219,207,141,209,207,072
50442 :024,173,220,207,109,208,183
50448 :207,141,208,207,169,000,180
50454 :109,209,207,141,209,207,080
50460 :024,169,032,109,209,207,010
50466 :141,209,207,173,208,207,155
50472 :133,251,173,209,207,133,122
50478 :252,173,212,207,041,007,170
50484 :141,225,207,056,169,007,089
50490 :237,225,207,141,225,207,020
50496 :169,000,141,206,207,056,075
50502 :173,225,207,046,206,207,110
50508 :206,225,207,016,245,160,111
50514 :000,173,227,205,201,005,125
50520 :240,090,201,002,240,064,157
50526 :201,004,208,003,076,180,254
50532 :197,173,228,206,240,010,130
50538 :177,251,013,206,207,145,081
50544 :251,076,180,197,173,227,192
50550 :205,201,001,240,018,173,188
50556 :206,207,073,255,141,206,188
50562 :207,177,251,045,206,207,199
50568 :145,251,076,180,197,177,138
50574 :251,045,206,207,240,032,099
50580 :177,251,013,206,207,145,123
50586 :251,076,180,197,177,251,006
50592 :045,206,207,240,015,173,022
50598 :206,207,073,255,141,206,230
50604 :207,177,251,045,206,207,241
50610 :145,251,177,251,141,224,087
50616 :205,096,165,197,201,004,028
50622 :208,046,169,000,133,170,148
50628 :169,032,133,171,160,000,093
50634 :152,145,170,056,165,170,036
50640 :233,255,141,212,206,165,140
50646 :171,233,063,013,212,206,088
50652 :240,016,024,165,170,105,172
50658 :001,133,170,165,171,105,203
50664 :000,133,171,076,200,197,241
50670 :096,165,197,170,201,028,071
50676 :208,008,169,015,141,212,229
50682 :206,076,010,198,201,021,194
50688 :208,104,169,240,141,212,050
50694 :206,076,034,198,238,214,204
50700 :206,173,214,206,045,212,044
50706 :206,201,015,208,035,173,088
50712 :214,206,041,240,141,214,056
50718 :206,076,058,198,024,173,253
50724 :214,206,105,016,141,214,164
50730 :206,045,212,206,201,240,128
50736 :208,008,173,214,206,041,130
50742 :015,141,214,206,169,000,031
50748 :133,170,169,004,133,171,072
50754 :173,214,206,160,000,145,196
50760 :170,056,165,170,233,231,073
50766 :141,212,206,165,171,233,182
50772 :007,013,212,206,176,016,202
```

```
50778 :024,165,170,105,001,133,176
50784 :170,165,171,105,000,133,072
50790 :171,076,066,198,096,160,101
50796 :128,185,119,198,153,064,187
50802 :008,136,016,247,096,255,104
50808 :255,255,192,000,003,192,249
50814 :000,003,192,000,003,192,004
50820 :000,003,192,000,003,192,010
50826 :000,003,192,000,003,192,016
50832 :000,003,192,000,003,192,022
50838 :000,003,192,000,003,192,028
50844 :000,003,192,000,003,192,034
50850 :000,003,192,000,003,192,040
50856 :000,003,192,000,003,192,046
50862 :000,003,192,000,003,255,115
50868 :255,255,000,000,048,000,226
50874 :000,060,000,000,063,000,053
50880 :000,062,000,000,055,000,053
50886 :000,003,128,000,001,192,010
50892 :000,000,224,000,000,000,172
50898 :000,000,000,000,000,000,210
50904 :000,000,000,000,000,000,216
50910 :000,000,000,000,000,000,222
50916 :000,000,000,000,000,000,228
50922 :000,000,000,000,000,000,234
50928 :000,000,000,000,000,000,240
50934 :000,000,000,000,000,000,246
50940 :000,169,012,141,033,208,047
50946 :169,147,032,210,255,169,216
50952 :021,141,024,208,169,027,086
50958 :141,017,208,169,000,141,178
50964 :208,205,133,180,141,207,070
50970 :205,141,206,205,133,195,087
50976 :169,216,133,196,169,004,151
50982 :133,181,162,000,160,000,162
50988 :169,128,141,210,205,140,013
50994 :206,205,172,207,205,185,206
51000 :000,008,140,207,205,172,020
51006 :206,205,045,210,205,240,149
51012 :011,169,001,145,195,169,246
51018 :160,145,180,076,088,199,154
51024 :169,000,145,195,169,160,150
51030 :145,180,024,165,195,105,132
51036 :001,133,195,165,196,105,119
51042 :000,133,196,024,165,180,028
51048 :105,001,133,180,165,181,101
51054 :105,000,133,181,078,210,049
51060 :205,173,210,205,240,003,128
51066 :076,049,199,238,207,205,072
51072 :169,128,141,210,205,232,189
51078 :224,003,144,167,024,165,093
51084 :180,105,016,133,180,165,151
51090 :181,105,000,133,181,024,002
51096 :165,195,105,016,133,195,193
51102 :165,196,105,000,133,196,185
51108 :162,000,238,208,205,173,126
51114 :208,205,201,021,176,003,216
51120 :076,049,199,096,169,001,254
51126 :141,238,002,096,165,197,253
51132 :201,041,240,001,096,169,168
51138 :000,032,189,255,169,004,075
51144 :170,160,255,032,186,255,234
51150 :032,192,255,162,004,032,115
51156 :201,255,176,003,076,220,119
51162 :199,096,169,008,032,210,164
51168 :255,169,013,032,210,255,134
51174 :162,000,169,001,141,204,139
51180 :205,169,000,141,250,206,183
51186 :169,000,141,251,206,169,154
51192 :199,141,248,206,169,005,192
51198 :141,227,205,142,242,002,189
51204 :032,022,196,174,242,002,160
51210 :173,224,205,045,206,207,046
51216 :240,012,173,202,205,013,093
51222 :204,205,141,202,205,076,031
51228 :041,200,173,204,205,073,156
51234 :255,045,202,205,141,202,060
51240 :205,014,204,205,173,204,021
51246 :205,201,128,240,020,024,096
51252 :173,250,206,105,001,141,160
51258 :250,206,173,251,206,105,225
51264 :000,141,251,206,076,001,227
51270 :200,173,202,205,009,128,219
51276 :224,045,144,010,173,202,106
51282 :205,041,031,009,128,141,125
51288 :202,205,168,032,210,255,136
51294 :152,032,210,255,169,001,145
51300 :141,204,205,169,000,141,192
51306 :202,205,056,173,250,206,174
51312 :233,006,141,250,206,173,097
51318 :251,206,233,000,141,251,176
51324 :206,206,248,206,173,248,131
51330 :206,201,255,240,003,076,087
51336 :001,200,224,045,176,031,045
51342 :024,173,250,206,105,007,139
51348 :141,250,206,173,251,206,095
51354 :105,000,141,251,206,232,065
51360 :169,199,141,248,206,169,012
51366 :013,032,210,255,076,001,241
51372 :200,169,013,032,210,255,027
51378 :032,231,255,096,174,240,182
51384 :002,160,255,136,208,253,174
51390 :202,208,248,096,173,167,004
51396 :002,174,168,002,160,001,191
51402 :032,186,255,173,169,002,251
51408 :162,172,160,002,032,189,157
51414 :255,169,000,162,000,160,192
51420 :032,032,213,255,096,173,253
51426 :167,002,174,168,002,160,131
51432 :001,032,186,255,173,169,024
51438 :002,162,172,160,002,032,000
51444 :189,255,169,032,133,254,252
51450 :169,000,133,253,169,253,203
51456 :162,255,160,063,032,216,120
51462 :255,096,165,197,201,022,174
51468 :240,001,096,169,000,133,139
51474 :170,169,032,133,171,169,094
51480 :000,133,180,169,096,133,223
51486 :181,160,000,177,170,141,091
51492 :062,003,177,180,141,064,151
51498 :003,173,062,003,145,180,096
51504 :173,064,003,145,170,024,115
51510 :165,170,105,001,133,170,030
51516 :165,171,105,000,133,171,037
51522 :024,165,180,105,001,133,162
51528 :180,165,181,105,000,133,068
51534 :181,056,165,170,233,255,114
51540 :141,200,205,165,171,233,175
51546 :063,013,200,205,144,193,140
51552 :096,013,013,013,013,013,001
```

Program 3:
BASIC Portion Of Hi-Res Graphics Editor

```
5 INPUT "{CLR}JOYSTICK SPEED (1-10)";JS$
                                    :rem 137
6 IF VAL(JS$)<1OR VAL(JS$)>10 THEN5
                                    :rem 192
7 POKE752,11-VAL(JS$)               :rem 180
8 FOR T= 2048TO2048+64:POKET,0:NEXT
                                     :rem 22
```

```
10 SYS50624                               :rem 97
11 SYS49152                               :rem 102
12 GETA$:IF PEEK(197)<>3THEN12            :rem 199
13 FOR T= 1 TO 300:NEXT                   :rem 188
15 SYS50941                               :rem 104
16 VI=53248:POKEVI+21,1:POKEVI,21:POKEVI+
   16,PEEK(VI+16)OR1:POKEVI+1,100 :rem 51
17 POKE2040,32                            :rem 238
20 SC= 1024:PX=0:PY=0:CN=0:OS=55296:OC=PE
   EK(OS)                                 :rem 24
30 GET A$:IF A$=""THEN CN=CN+1            :rem 65
31 IF PEEK(197)=4 THEN FOR T=2048TO2048+6
   4:POKET,0:NEXT:SYS50941                :rem 196
32 IF PEEK(197)=3THENPOKE198,0:FORT=1TO30
   0:NEXT:GOTO11                          :rem 62
33 IF A$="{L}"THEN GOSUB 300:SYS51394:GOS
   UB400:SYS50941                         :rem 242
34 IF A$="{HOME}"THEN GOSUB300:SYS51425:G
   OSUB400:SYS50941                       :rem 245
40 IF CN= 2 THEN POKE SC,PEEK(SC)OR128:CN
   =0                                     :rem 147
50 IF CN= 1 THEN POKE SC,PEEK(SC)AND127
                                          :rem 140
60 IF(PEEK(56320)AND16)<>0 THEN 65:rem 58
61 IF PEEK(653)THEN POKESC+54272,0:SH=1:G
   OSUB200:GOTO 65                        :rem 246
63 POKESC+54272,1:SH=0:GOSUB 200   :rem 72
65 IF 15-PEEK(56320)=0 THEN 79     :rem 15
66 FL=0:OC=PEEK(SC+54272):OS=SC+54272
                                          :rem 141
70 ON 15-PEEK(56320)AND15GOSUB 80,90,95,1
   00,120,130,140,150,160,170             :rem 163
72 POKESC,(PEEK(SC)OR128)                 :rem 243
75 SC=1024+40*Y+X                         :rem 155
79 GOTO 30                                :rem 12
80 Y =Y+(Y>0):RETURN                      :rem 180
90 Y=Y-(Y<20):RETURN                      :rem 231
95 RETURN                                 :rem 78
100 X=X+(X>0):RETURN                      :rem 218
110 RETURN                                :rem 114
120 Y=Y+(Y>0):X=X+(X>0):RETURN     :rem 72
130 Y=Y-(Y<20):X=X+(X>0):RETURN   :rem 123
140 RETURN                                :rem 117
150 X=X-(X<23):RETURN                     :rem 20
160 Y=Y+(Y>-0):X=X-(X<23):RETURN :rem 174
170 Y=Y-(Y<20):X=X-(X<23):RETURN :rem 180
200 BO=Y*3+INT(X/8)                       :rem 60
210 BT= 2↑(7-(X-INT(X/8)*8)):P=64*PEEK(20
    40)+BO                                :rem 49
220 IF SH=0 THENPOKEP,PEEK(P)ORBT:GOTO230
                                          :rem 10
225 POKEP,PEEK(P)AND(255-BT):SH=0:rem 207
230 RETURN                                :rem 117
300 PRINT"{BLK}{7 RIGHT}{CLR}{RVS}D{OFF}I
    SK OR {RVS}T{OFF}APE"                 :rem 144
301 GET J$:IF J$=""THEN301                :rem 93
302 IF J$<>"D"AND J$<>"T"THEN 301:rem 170
303 INPUT "FILENAME";FI$                  :rem 153
305 IF LEFT$(J$,1)="D"THEN D=8:GOTO310
                                          :rem 70
306 D=1                                   :rem 75
310 FOR T= 684 TO 684+LEN(FI$)-1:POKET,AS
    C(MID$(FI$,T-683,1)):NEXT             :rem 150
320 POKE679,D:POKE680,D:POKE681,LEN(FI$):
    POKE682,172:POKE683,2                 :rem 159
325 RETURN                                :rem 122
400 OPEN15,8,15:INPUT#15,A$,B$,C$,D$:PRIN
    TA$;" ";B$" ";C$;" ";C$;" ";D$:rem 52
405 CLOSE15                               :rem 117
410 FOR T= 1TO 3000 :NEXT :RETURN:rem 55
```

Snertle

Soori Sivakumaran

By making simple selections from a menu, a child can change this arithmetic drill to fit his or her own tutoring needs. Written for the unexpanded VIC, versions also are included for the Commodore 64, Atari, TI-99/4A, Color Computer, Apple, IBM PC, and PCjr.

"Snertle" is designed to help teach children the fundamentals of addition, subtraction, and multiplication. A turtle named Snertle is drawn on the screen to give encouragement and assistance to the player.

An Individual Challenge

Snertle allows children to tailor math problems to fit their individual abilities and weaknesses. Snertle first asks the child to select addition, subtraction, or multiplication problems. If addition or subtraction is selected, the child is then asked to choose the largest and smallest numbers to be used in creating the problems. The largest number that can be chosen is 99 and the smallest number is zero.

If multiplication is chosen, the child can decide to practice a certain "times table," or solve problems created randomly from 0 through the 14 times table.

For example, if the 12 times table is selected, then one number in each question created will always be 12. The other number will be randomly selected from the range 0–14.

If the child chooses to attempt random multiplication problems, he or she must define the range of numbers (within the limits of 0 and 14) from which the problems can be created, similar to the process for random addition or subtraction problems.

Creating The Screen

In Program 1, once the necessary information is entered, the turtle's image is POKEd onto the screen. The two numbers used in the problem are chosen in lines 305, 315, and 1070. The numbers are then displayed on the screen, each digit being four regular characters high and three wide. The large character set is created in a series of subroutines in lines 500–990.

The larger number is always displayed above the smaller number to avoid negative answers to subtraction problems. The appropriate sign for addition, subtraction, or multiplication is drawn on the screen by a subroutine beginning at line 6000. Next, a horizontal line is drawn under the numbers.

Line 394 contains a FOR–NEXT loop that clears the keyboard buffer. This prevents the child from accidentally entering data while the turtle and the problem are being put on the screen.

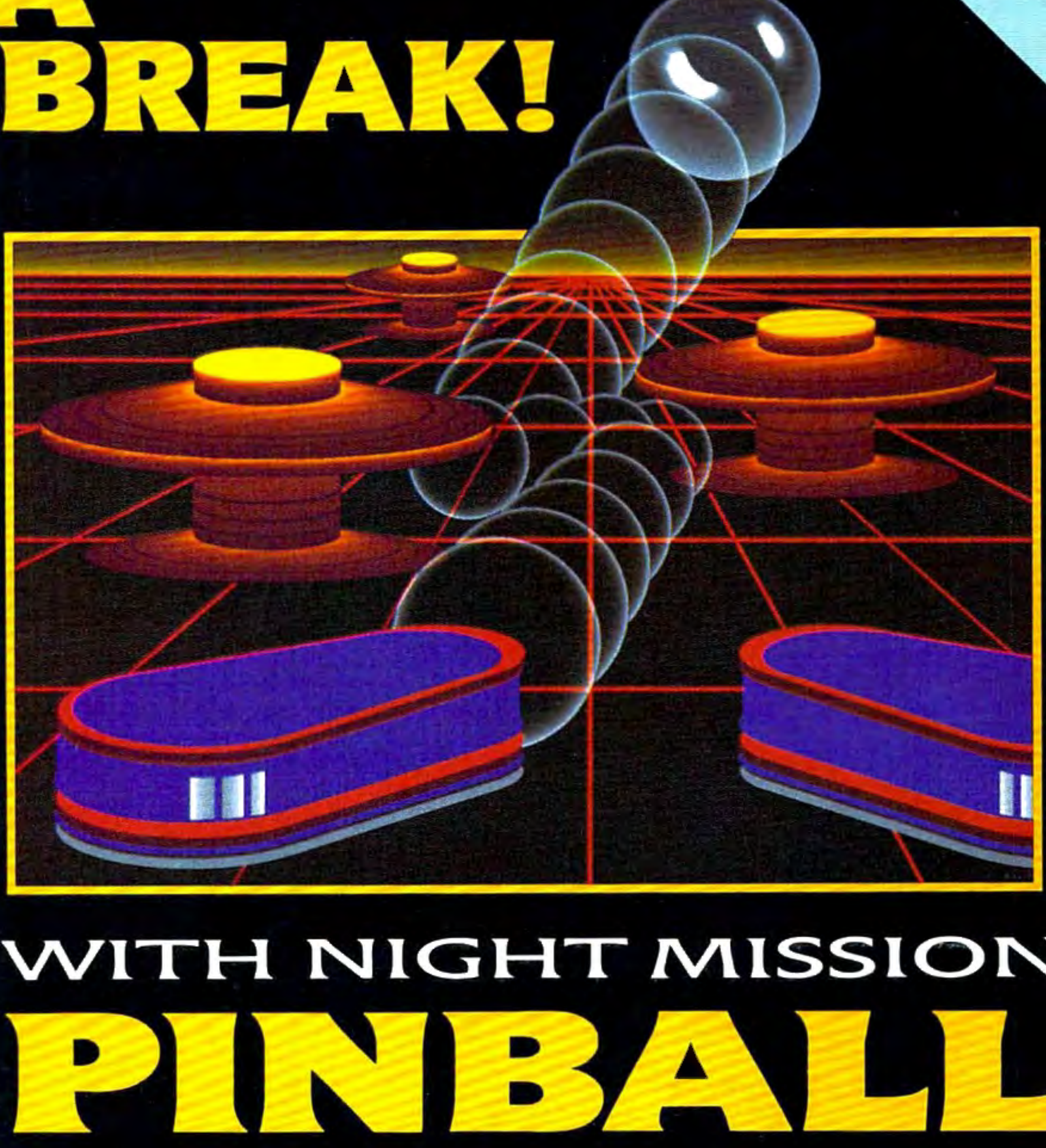
Another FOR–NEXT loop in lines 395–420 enters the user's response to the problem. Because a GET statement is used, the RETURN key does not have to be pressed when entering the response. An arrow will appear at the bottom of the screen to prompt for each digit of the response.

The Turtle Smiles

Once the response is entered, Snertle checks it against the correct answer. If the child's response is correct the turtle will smile, GOOD! will appear on its shell, and a high beep will sound. If the

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

TAKE A BREAK!



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

response is incorrect, Snertle the turtle's head will disappear into his shell and the message TRY AGAIN will appear on his side.

The user will be given a second chance. If the new response is correct, Snertle will poke his head out from his shell. If the answer is again incorrect, the correct answer will be displayed on the screen.

The program will keep producing problems until the X key is pressed in response to a problem. The percentage of correctly answered questions is then calculated in line 410, and displayed. This percentage only includes problems answered correctly on the first attempt. Snertle then returns to the menu where the child may END the program or select more problems.

Program 1 uses all but 84 bytes of the unexpanded VIC's memory.

A subtraction problem—"Snertle" for VIC. Other versions similar.

Program 1: Snertle For VIC

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

```
100 A$=CHR$(147):B$=CHR$(17):C$=CHR$(29):
    D$=CHR$(18):E$=CHR$(146):Y=160:LL=368
    76                              :rem 62
110 PRINTA$SPC(5)B$B$"**SNERTLE**":POKELL
    +2,15                          :rem 181
120 PRINTB$B$B$B$C$C$ D$"SELECT ONE:"E$
                                   :rem 119
130 PRINTB$"1) ADDITION"           :rem 113
140 PRINTB$"2) SUBTRACTION"        :rem 117
150 PRINTB$"3) MULTIPLICATION"      :rem 87
155 PRINTB$"4) END PROGRAM"         :rem 30
160 PRINTB$"(ENTER 1,2,3 OR 4)";:INPUTQ:I
    FQ>4ORQ<0THEN160               :rem 102
185 C=14:IFQ=1ORQ=2THENC=99        :rem 141
187 IFQ=3THEN1000                  :rem 224
188 IFQ=4THENEND                   :rem 248
190 PRINTA$B$B$"ENTER LARGEST VALUE"
                                   :rem 169
200 PRINT"(MIN.:0 MAX.:";C;")";:INPUTR:IF
    R<0ORR>CTHEN200                :rem 142
230 PRINTB$B$"ENTER SMALLEST VALUE"
                                   :rem 146
240 PRINT"(MIN.:0 MAX.:";R;")";:INPUTS:IF
    S<0ORS>RTHEN240                :rem 183
263 PRINTA$B$"PRESS "D$"X"E$" RETURN TO M
    ENU":FORI=1TO750:NEXTI           :rem 6
265 PRINTA$                        :rem 143
270 Z=0:ZZ=0:GOSUB2000              :rem 55
275 GOSUB1100:GOSUB1170:GOSUB1230:GOSUB12
    60                             :rem 102
301 TR=0:ZZ=ZZ+1                   :rem 226
305 L=INT(RND(1)*(R-S+1))+S        :rem 234
310 IFQ=3ANDT=1THEN320              :rem 61
315 K=INT(RND(1)*(R-S+1))+S        :rem 234
320 F$=STR$(K):W=0                 :rem 243
325 IFK<LTHENW=110                  :rem 81
330 GOSUB3000                      :rem 217
335 W=110                          :rem 193
337 IFL>KTHENW=0                   :rem 244
340 F$=STR$(L)                     :rem 248
345 GOSUB3000                      :rem 223
346 ONQGOSUB6000,6000,6004         :rem 185
350 IFQ=1THENM=K+L                  :rem 97
355 IFQ=2ANDK>=LTHENM=K-L           :rem 78
360 IFQ=2ANDK<LTHENM=L-K            :rem 11
365 IFQ=3THENM=K*L                 :rem 104
380 GOSUB740:MM=1:IFM>9THENMM=2    :rem 189
385 IFM>99THENMM=3                 :rem 101
390 GOSUB740                       :rem 183
393 V=0:GOSUB1100                  :rem 222
394 FORI=631TO640:POKEI,0:NEXTI    :rem 180
395 FORJ=0 TO MM-1                 :rem 218
397 POKE8177-(4*J),30               :rem 94
400 GETH$                          :rem 224
405 IFH$=""THEN400                 :rem 216
407 IFH$="X"ANDZZ=1THEN100          :rem 36
410 IFH$="X"THENPRINTA$"PERCENTAGE:";INT(
    Z/(ZZ-1)*100):GOTO120           :rem 10
412 FORO=8164TO8168:POKEO,32:NEXTO
                                   :rem 104
415 P=VAL(H$)                      :rem 199
420 V=V+(P*10↑J):X=8110-(4*J):GOSUB480:NE
    XTJ                             :rem 86
450 IFM=VTHEN470                   :rem 210
451 POKELL,160:FORI=1TO500:NEXTI:POKELL,0
                                    :rem 83
452 FORI=8098TO8186:POKEI,32:NEXTI:rem 96
456 IFTR=1THEN460                   :rem 11
458 TR=1:GOSUB1500:GOSUB770:GOTO393
                                   :rem 159
460 M$=STR$(M)                       :rem 3
461 FORI=1TO22-MM:READA:NEXTI       :rem 96
462 FOROO=1TOMM                    :rem 204
464 P=VAL(MID$(M$,(OO+1),1))       :rem 243
465 READX:GOSUB480:NEXTOO:RESTORE:rem 222
470 GOSUB1230:IFTR=0THENGOSUB2500:GOSUB75
    5:Z=Z+1:GOSUB6500              :rem 154
471 GOSUB2225:GOTO301              :rem 238
480 IFP=0THENGOSUB720               :rem 48
485 ONPGOSUB 500,525,555,585,610,633,660,
    680,700:RETURN                 :rem 254
500 FORI=0TO66STEP22:POKEX+I+1,Y:NEXTI:RE
    TURN                           :rem 211
525 GOSUB990:GOSUB980:POKEX+44,Y:GOSUB970
    :RETURN                        :rem 102
555 GOSUB990:GOSUB980:POKEX+46,Y:GOSUB970
    :RETURN                        :rem 107
585 POKEX,Y:POKEX+22,160           :rem 193
595 FORI=44TO46:POKEI+X,Y:NEXTI      :rem 1
600 POKEX+23,118:POKEX+67,118:RETURN
                                   :rem 172
610 GOSUB990                       :rem 185
```

```
620 POKEX+22,Y:POKEX+23,98:POKEX+24,98:PO
    KEX+46,Y:GOSUB970:RETURN        :rem 95
633 GOSUB990                       :rem 190
640 POKEX+22,Y:POKEX+23,98:POKEX+24,98 ,
                                    :rem 18
645 POKEX+44,Y:POKEX+46,Y:GOSUB970:RETURN
                                   :rem 141
660 GOSUB990                       :rem 190
670 POKEX+24,Y:POKEX+45,Y:POKEX+46,97:POK
    EX+67,Y:RETURN                 :rem 254
680 GOSUB525                       :rem 186
690 POKEX+22,Y:POKEX+46,Y:RETURN    :rem 47
700 GOSUB680:POKEX+44,32:RETURN    :rem 180
720 GOSUB680:POKEX+23,32:RETURN    :rem 179
740 FORI=8080TO8093:POKEI,64:NEXTI:RETURN
                                   :rem 115
755 POKE7753,7:POKE7754,15:POKE7755,15:PO
    KE7756,4:POKE7757,33            :rem 37
760 POKE7753,7:POKE7754,15:POKE7755,15:PO
    KE7756,4:POKE7757,33:RETURN     :rem 59
770 POKE7732,20:POKE7733,18:POKE7734,25
                                   :rem 209
780 POKE7753,1:POKE7754,7:POKE7755,1:POKE
    7756,9:POKE7757,14:POKE7758,33
                                   :rem 147
785 FORI=1TO750:NEXTI:RETURN        :rem 93
960 FORI=ØTO66STEP22:POKE I+X,160:NEXTI:R
    ETURN                          :rem 191
970 FORI=ØTO2:POKEI+66+X,160:NEXTI:RETURN
                                   :rem 125
980 POKEX+22,98:POKEX+23,98:POKEX+24,160:
    RETURN                         :rem 113
990 FORI=ØTO2:POKEX+I,160:NEXTI:RETURN
                                   :rem 232
1000 PRINTA$B$B$SPC(2)"DO YOU WISH TO:"
                                   :rem 212
1010 PRINTB$SPC(3)"1) PRACTICE TIMES"
                                   :rem 138
1015 PRINT"TABLES"                  :rem 83
1020 PRINTB$SPC(3)"2) RANDOM NUMBERS"
                                   :rem 156
1030 PRINT"(ENTER 1 OR 2)";:INPUTT:IFT<ØO
     RT>2THEN1030                  :rem 162
1050 IFT=2THENGOTO190               :rem 26
1060 PRINTA$B$B$SPC(2)"ENTER TIMES TABLE"
                                   :rem 154
1070 PRINTB$SPC(3)"(1-14)";:INPUTK:IFK<1O
     RK>14THEN1070                 :rem 212
1090 S=Ø:R=14:GOTO263              :rem 198
1100 FORI=7702TO7790STEP22          :rem 25
1110 READA:READB                   :rem 184
1120 FORJ=1TOB                      :rem 72
1130 POKE (I+A+J),102               :rem 46
1140 NEXTJ:NEXTI:RESTORE:RETURN    :rem 137
1170 FORI=1TO11                    :rem 108
1180 POKE(7815+I),120               :rem 82
1190 NEXTI                          :rem 83
1200 POKE7793,74                    :rem 99
1210 RETURN                        :rem 164
1230 FORI=1TO1Ø:READA:NEXTI        :rem 193
1232 FORI=7724TO7768STEP 22         :rem 40
1234 FORJ=15TO17                   :rem 169
1235 READA:POKEI+J,A:NEXTJ:NEXTI:RESTORE:
     RETURN                        :rem 185
1260 FORI=1TO2                      :rem 60
1270 POKE7817+I,Y:POKE7821+I,Y:NEXTI
                                   :rem 191
1300 FORI=1TO3                      :rem 56
1310 POKE7839+I,Y                  :rem 200
1320 POKE7843+I,Y                  :rem 196
1330 NEXTI:RETURN                  :rem 105
1500 FORI=7724TO7768STEP 22         :rem 38
1510 FORJ=15TO17:POKEI+J,32:NEXTJ:NEXTI:R
     ETURN                         :rem 253
2000 FORI=384ØØTO38575             :rem 221
2001 POKEI,5:NEXTI                  :rem 94
2003 POKE38482,6:FORI=38576TO389Ø5:POKEI,
     1+Q:NEXTI:RETURN               :rem 38
2225 FORI=7878TO8185:POKEI,32:NEXTI:RETUR
     N                             :rem 174
2500 POKE7785,2Ø2:RETURN           :rem 171
3000 IFLEN(F$)>2THEN3Ø3Ø            :rem 81
3015 P=VAL(MID$(F$,2,1))           :rem 254
3020 X=789Ø+W:GOSUB48Ø              :rem 1Ø
3025 RETURN                        :rem 17Ø
3030 P=VAL(MID$(F$,2,1))           :rem 251
3035 X=7886+W:GOSUB48Ø              :rem 21
3040 P=VAL(MID$(F$,3,1))           :rem 253
3045 X=789Ø+W:GOSUB 48Ø             :rem 17
3050 RETURN                        :rem 168
5000 DATA 6,5,5,7,4,9,3,11,3,11,233,16Ø,1
     6Ø,16Ø,1Ø8,16Ø,16Ø,16Ø,16Ø,81Ø2,81Ø6
     ,811Ø                         :rem 159
6000 POKE8Ø15,Y:POKE8Ø36,Y:POKE8Ø37,Y:POK
     E8Ø38,Y:POKE8Ø59,Y             :rem 76
6002 IFQ=2THENPOKE8Ø15,32:POKE8Ø59,32
                                   :rem 164
6003 RETURN                        :rem 169
6004 POKE8Ø14,Y:POKE8Ø16,Y:POKE8Ø37,Y:POK
     E8Ø58,Y:POKE8Ø6Ø,Y:RETURN      :rem 97
6500 POKELL,2Ø7:FORI=1TO15Ø:NEXTI:POKELL,
     215:FORI=1TO175:NEXTI:POKELL,Ø:RETUR
     N                              :rem 64
```

Subtraction, 64 version of "Snertle." Other versions similar.

Program 2: Snertle For Commodore 64

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

```
90 FOR I=54272 TO 54296:POKEI,Ø:NEXTI
                                    :rem 87
100 A$=CHR$(147):B$=CHR$(17):C$=CHR$(29):
    D$=CHR$(18):E$=CHR$(146):Y=16Ø:rem 33
105 LL=54272:POKELL+5,1:POKELL+6,241:POKE
    LL+24,15                       :rem 118
110 PRINTA$SPC(15)B$B$"**SNERTLE**":POKE5
    3281,1                         :rem 191
120 PRINTTAB(13)B$B$B$B$C$C$ D$"SELECT ON
    E:"E$                            :rem 3
130 PRINTTAB(13)B$"1) ADDITION"   :rem 253
```

```
140 PRINTTAB(13)B$"2) SUBTRACTION" :rem 1
150 PRINTTAB(13)B$"3) MULTIPLICATION"
                                  :rem 227
155 PRINTTAB(13)B$"4) END PROGRAM"
                                  :rem 170
160 PRINT"{HOME}{16 DOWN}"TAB(13)B$"(ENTE
    R 1,2,3 OR 4)";:INPUTQ        :rem 169
170 IFQ>4ORQ<1THEN160              :rem 15
185 C=14:IFQ=1ORQ=2THENC=99       :rem 141
187 IFQ=3THEN1000                 :rem 224
188 IFQ=4THENPRINT"{CLR}":END     :rem 150
190 PRINTA$B$B$TAB(10)"ENTER LARGEST VALU
    E"                             :rem 50
200 PRINT"{HOME}{3 DOWN}"TAB(10)"(MIN.:1
    {SPACE}MAX.:";C;")";:INPUTR:IFR<1ORR>
    CTHEN200                      :rem 163
230 PRINTB$B$TAB(10)"ENTER SMALLEST VALUE
    "                              :rem 27
240 PRINT"{HOME}{8 DOWN}"TAB(10)"(MIN.:0
    {SPACE}MAX.:";R;")";:INPUTS:IFS<0ORS>
    RTHEN240                       :rem 31
263 PRINTA$B$B$B$B$B$B$TAB(8)"PRESS "D$"X
    "E$" RETURN TO MENU":FORI=1TO1500:NEX
    T                              :rem 69
265 PRINTA$                       :rem 143
270 Z=0:ZZ=0:GOSUB2000             :rem 55
275 GOSUB1100:GOSUB1170:GOSUB1230:GOSUB12
    60                            :rem 102
301 TR=0:ZZ=ZZ+1                  :rem 226
305 L=INT(RND(1)*(R-S+1))+S       :rem 234
310 IFQ=3ANDT=1THEN320             :rem 61
315 K=INT(RND(1)*(R-S+1))+S       :rem 234
320 F$=STR$(K):W=0                :rem 243
325 IFK<LANDQ=2 THEN305            :rem 86
330 W=5:GOSUB3000                 :rem 220
337 IFL>KTHENW=0                  :rem 244
340 F$=STR$(L)                    :rem 248
345 W=205:GOSUB3000                :rem 68
346 ONQGOSUB6000,6000,6004        :rem 185
350 IFQ=1THENM=K+L                 :rem 97
355 IFQ=2ANDK>=LTHENM=K-L          :rem 78
360 IFQ=2ANDK<LTHENM=L-K           :rem 11
365 IFQ=3THENM=K*L                :rem 104
380 GOSUB740:MM=1:IFM>9THENMM=2   :rem 189
385 IFM>99THENMM=3                :rem 101
390 GOSUB740                      :rem 183
393 V=0:GOSUB1100                 :rem 222
394 FORI=631TO640:POKEI,0:NEXTI   :rem 180
395 FORJ=0 TO MM-1                :rem 218
397 POKE1802-(4*J),30              :rem 82
400 GETH$                         :rem 224
405 IFH$=""THEN400                :rem 216
407 IFH$="X"ANDZZ=1THEN100         :rem 36
410 IFH$="X"THENPRINTA$B$B$SPC(13)"PERCEN
    TAGE:";INT(Z/(ZZ-1)*100):GOTO120
                                  :rem 113
411 IF H$<>"0"AND VAL(H$)=0 THEN 400
                                   :rem 34
412 FORO=1984TO2023:POKEO,32:NEXTO:rem 91
415 P=VAL(H$)                     :rem 199
420 V=V+(P*10↑J):X=1801-(4*J):GOSUB480:NE
    XTJ                            :rem 86
450 IFM=VTHEN470                  :rem 210
451 GOSUB 6600                    :rem 230
452 FORI=1792TO1943:POKEI,32:NEXTI:rem 84
456 IFTR=1THEN:GOTO460            :rem 126
458 TR=1:GOSUB1500:GOSUB770:GOTO393
                                  :rem 159
460 M$=STR$(M)                      :rem 3
461 FORI=1TO25-MM:READA:NEXTI      :rem 99
462 FOROO=1TOMM                   :rem 204
464 P=VAL(MID$(M$,(OO+1),1))      :rem 243
465 READX:GOSUB480:NEXTOO:RESTORE:rem 222
470 GOSUB1230:IFTR=0THENGOSUB2500:GOSUB75
    5:Z=Z+1:GOSUB6500             :rem 154
471 GOSUB2225:GOTO301             :rem 238
480 IFP=0THENGOSUB720              :rem 48
485 ONPGOSUB 500,525,555,585,610,633,660,
    680,700:RETURN                :rem 254
500 FORI=0TO120STEP40:POKEX+I+1,Y:NEXTI:R
    ETURN                         :rem 250
525 GOSUB990:GOSUB980:POKEX+80,Y:GOSUB970
    :RETURN                       :rem 102
555 GOSUB990:GOSUB980:POKEX+82,Y:GOSUB970
    :RETURN                       :rem 107
585 POKEX,Y:POKEX+40,160          :rem 193
595 FORI=80TO82:POKEI+X,Y:NEXTI     :rem 1
600 FORI=1TO2:POKEX+I,118:POKEX+40+I,118:
    POKEX+120+I,118:RETURN         :rem 97
610 GOSUB990                      :rem 185
620 POKEX+40,Y:POKEX+41,98:POKEX+42,98:PO
    KEX+82,Y:GOSUB970:RETURN       :rem 95
633 GOSUB990                      :rem 190
640 POKEX+40,Y:POKEX+41,98:POKEX+42,98
                                   :rem 18
645 POKEX+80,Y:POKEX+82,Y:GOSUB970:RETURN
                                  :rem 141
660 GOSUB990                      :rem 190
670 POKEX+42,Y:POKEX+81,Y:POKEX+82,97:POK
    EX+121,Y:RETURN                :rem 37
680 GOSUB525                      :rem 186
690 POKEX+40,Y:POKEX+82,Y:RETURN   :rem 47
700 GOSUB680:POKEX+80,32:RETURN   :rem 180
720 GOSUB680:POKEX+41,32:RETURN   :rem 179
740 FORI=1748TO1763:POKEI,64:NEXTI:RETURN
                                  :rem 116
755 POKE1151,7:POKE1152,15:POKE1153,15:PO
    KE1154,4:POKE1155,33          :rem 223
760 POKE1151,7:POKE1152,15:POKE1153,15:PO
    KE1154,4:POKE1155,33:RETURN   :rem 245
770 POKE1112,20:POKE1113,18:POKE1114,25
                                  :rem 167
780 POKE1151,1:POKE1152,7:POKE1153,1:POKE
    1154,9:POKE1155,14:POKE1156,33:rem 63
785 FORI=1TO250:NEXTI:RETURN       :rem 88
960 FORI=0TO120STEP40:POKE I+X,160:NEXTI:
    RETURN                        :rem 230
970 FORI=0TO2:POKEI+120+X,160:NEXTI:RETUR
    N                             :rem 164
980 POKEX+40,98:POKEX+41,98:POKEX+42,160:
    RETURN                        :rem 113
990 FORI=0TO2:POKEX+I,160:NEXTI:RETURN
                                  :rem 232
1000 PRINTA$B$B$SPC(11)"DO YOU WISH TO:"
                                    :rem 4
1010 PRINTB$SPC(11)"1) PRACTICE TIMES TAB
     LES"                         :rem 116
1020 PRINTB$SPC(11)"2) RANDOM NUMBERS"
                                  :rem 203
1030 PRINT"{HOME}{7 DOWN}"B$SPC(11)"(ENTE
     R 1 OR 2)";:INPUTT           :rem 142
1040 IFT<1ORT>2THEN1030           :rem 109
1050 IFT=2THENGOTO190              :rem 26
1060 PRINTA$B$B$SPC(11)"ENTER TIMES TABLE
     "                            :rem 202
1070 PRINT"{HOME}{3 DOWN}"B$SPC(11)"(1-14
     )";:INPUTK:IFK<1ORK>14THEN1070
                                  :rem 141
1090 S=0:R=14:GOTO263             :rem 198
1100 FORI=1064TO1224STEP40          :rem 6
1110 READA:READB                  :rem 184
1120 FORJ=1TOB*2-1                  :rem 2
```

"BEAT TO QUARTERS! RUN OUT THE GUNS, AND CLEAR FOR ACTION!"

These urgent commands of the 18th-century captain are now yours to issue as you re-enact legendary naval engagements from the age of fighting sail.

BROADSIDES™ — a new game from SSI — gives you non-stop naval action, as fast and demanding as the historical battles it re-creates.

In this two-player/solitaire simulation, the ships will be rigged, manned, and armed just like the real ships were, and will handle in the wind just as real ships sailed.

For speed of play, the ARCADE game lets you sail right into action. You set your course, maneuver and fire broadsides as fast as your crew can reload.

For authenticity and historical detail, the TACTICAL game puts you on the quarter-deck. There, you make the decisions real captains had to make: How much sail? What course for best speed? Aim your guns for the rigging or waterline? What range? Load with solid shot or grape shot?

When the ships move within grappling range, your boarding parties and sharpshooters will decide the battle. As you watch the fighting on your screen, you'll give orders for your saber-wielding men to thrust, counter-thrust or hack away.

If you want to design your own ships, **BROADSIDES** lets you build frigates and ships-of-the-line from the keel up, rigging them, manning them, and arming them with everything from 12-pounders to hull-smashing carronades.

The enemy's guns have been rolled out, his sharpshooters are in his tops, and he's almost within range. The battle is about to begin.

Your ships move faster than real-time on the Hi-Res color screen.

The semi-animated boarding phase.

NOW ALSO ON 48K DISK FOR
ATARI®
HOME COMPUTERS

FOR THE APPLE®

BROADSIDES™ ($39.95) comes on 48K disk for the Apple II with Applesoft ROM, II+, IIe, and Apple III.

Designed by Wayne Garris.

All SSI games are covered by a 14-day "satisfaction or your money back" guarantee.

RapidFire
GAMES FROM SSI

If there are no convenient stores near you, VISA & Mastercard holders can order direct by calling **800-227-1617, ext. 335 (toll free).** In California, call 800-772-3545, ext. 335. To order by mail, send your check to: Strategic Simulations Inc, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. Please specify computer format and include $2.00 for shipping & handling. (California residents, add 6½% sales tax.)

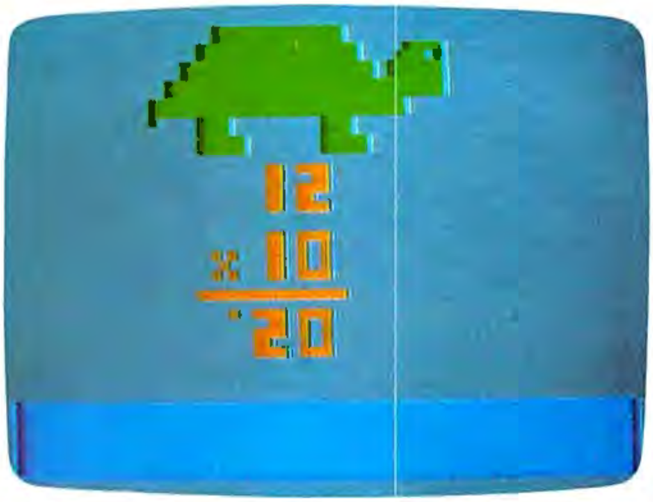
WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

```
1130 POKE (I+A+J),102                :rem 46
1140 NEXTJ:NEXTI:RESTORE:RETURN      :rem 137
1170 FORI=1TO21                      :rem 109
1180 POKE(1267+I),120                 :rem 77
1190 NEXTI                            :rem 83
1200 POKE1227,74                      :rem 85
1210 RETURN                          :rem 164
1230 FORI=1TO10:READA:NEXTI          :rem 193
1232 FORI=1104TO1184STEP 40           :rem 12
1234 FORJ=25TO28                     :rem 172
1235 READA:POKEI+J,A:NEXTJ:NEXTI:RESTORE:
     RETURN                          :rem 185
1260 FORI=1TO3                        :rem 61
1270 POKE1271+I,Y:POKE1280+I,Y:NEXTI
                                     :rem 172
1300 FORI=1TO4                        :rem 57
1310 POKE1311+I,Y                    :rem 179
1320 POKE1320+I,Y                    :rem 180
1330 NEXTI:RETURN                    :rem 105
1500 FORI=1064 TO 1224STEP 40         :rem 10
1510 FORJ=25TO28:POKEI+J,32:NEXTJ:NEXTI:R
     ETURN                             :rem 0
2000 FORI=55296TO55615               :rem 227
2001 POKEI,5:NEXTI                    :rem 94
2003 POKE55442,6:FORI=55616TO56256:POKEI,
     1+Q:NEXTI:RETURN                 :rem 26
2225 FORI=1384TO2023:POKEI,32:NEXTI:RETUR
     N                               :rem 145
2500 POKE1212,202:RETURN             :rem 150
2600 FORI=1TO24:POKELL+I,0:NEXTI:POKELL+5
     ,240:POKELL+6,72:POKEV,72:RETURN
                                     :rem 138
3000 IFLEN(F$)>2THEN3030              :rem 81
3015 P=VAL(MID$(F$,2,1))             :rem 254
3020 X=1396+W:GOSUB480                 :rem 5
3025 RETURN                          :rem 170
3030 P=VAL(MID$(F$,2,1))             :rem 251
3035 X=1392+W:GOSUB480                 :rem 7
3040 P=VAL(MID$(F$,3,1))             :rem 253
3045 X=1396+W:GOSUB 480               :rem 12
3050 RETURN                          :rem 168
5000 DATA 6,5,5,7,4,9,3,11,3,11,233,160,1
     60,160,160,108,160,160          :rem 244
5010 DATA 160,160,160,160,1793,1797,1801
                                     :rem 186
6000 POKE1631,Y:POKE1670,Y:POKE1671,Y:POK
     E1672,Y:POKE1711,Y               :rem 52
6002 IFQ=2THENPOKE1631,32:POKE1711,32
                                     :rem 149
6003 RETURN                          :rem 169
6004 POKE1630,Y:POKE1632,Y:POKE1671,Y:POK
     E1710,Y:POKE1712,Y:RETURN        :rem 73
6500 POKE LL+4,33:POKELL+1,21:POKELL,31:F
     ORI=1TO200:NEXTI:POKELL+1,25:POKELL,
     30                               :rem 79
6510 FORI=1TO600:NEXTI:POKELL+4,32:FORI=1
     TO1000:NEXTI:POKELL+4,8:RETURN
                                      :rem 50
6600 POKE LL+4,33:POKELL+1,10:POKELL,143:
     FORI=1TO1500:NEXTI:POKELL+4,32
                                      :rem 39
6610 FOR I=1TO1000:NEXTI:POKELL+4,8:RETUR
     N                               :rem 111
```

Program 3: Snertle For Atari

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

```
ME 90 DIM F$(4),M$(3):OPEN #1,4,0,"K:"
DL 100 GRAPHICS 17:SETCOLOR 0,12,10
KN 110 POSITION 3,1:? #6;"**SNERTLE**"
```



The final digit is just beginning to appear onscreen, Atari version of "Snertle." Other versions similar.

```
MI 120 POSITION 3,4:? #6;"SELECT ONE:"
JG 130 POSITION 3,7:? #6;"1) ADDITION"
JM 140 POSITION 3,9:? #6;"2) SUBTRACTI
       ON"
KH 150 POSITION 3,11:? #6;"3) MULTIPLI
       CATION"
GM 160 POSITION 3,13:? #6;"4) END PROG
       RAM"
PM 170 POSITION 1,17:? #6;"( ENTER 1,2
       ,3 OR 4)";:GET #1,Q:IF Q<49 OR
       Q>52 THEN 170
DP 185 Q=Q-48:C=14:IF Q=1 OR Q=2 THEN
       C=99
OA 187 IF Q=3 THEN 1000
PI 188 IF Q=4 THEN END
OP 190 GRAPHICS 17:POSITION 0,3:? #6;"
       ENTER LARGEST VALUE"
KM 200 ? #6;"MIN.:1  MAX.:";C;" ";
GL 203 GET #1,R:IF R<48 OR R>57 THEN 2
       03
JA 205 ? #6;R-48;
IL 210 GET #1,RR:IF (RR<48 OR RR>57) A
       ND (RR<>155) THEN 210
GG 211 IF RR=155 THEN 215
KF 212 ? #6;RR-48
OK 215 IF RR=155 THEN RR=R:R=48
GI 220 R=10*(R-48)+RR-48:IF R<1 OR R>C
        THEN PRINT #6:GOTO 200
HM 230 POSITION 0,14:? #6;"ENTER SMALL
       EST VALUE"
LO 240 ? #6;"MIN.:0  MAX.:";R;" ";
HE 242 GET #1,S:IF S<48 OR S>57 THEN 2
       42
JE 244 ? #6;S-48;
JM 250 GET #1,SS:IF (SS<48 OR SS>58) A
       ND (SS<>155) THEN 250
EN 251 IF SS=155 THEN SS=S:S=48:GOTO 2
       53
KL 252 ? #6;SS-48
IG 253 S=10*(S-48)+SS-48:IF S<0 OR S>R
        THEN PRINT #6:GOTO 240
OK 263 GRAPHICS 17:POSITION 2,8:? #6;"
       ENTER X TO RETURN":POSITION 6,1
       0:? #6;"TO MENU":FOR I=1 TO 500
       :NEXT I:? "{CLEAR}"
NP 270 Z=0:ZZ=0:GRAPHICS 5:POKE 752,1
OD 275 GOSUB 1100:GOSUB 1170:GOSUB 123
       0
OC 301 TR=0:ZZ=ZZ+1
OK 305 L=INT(RND(1)*(R-S+1))+S
```

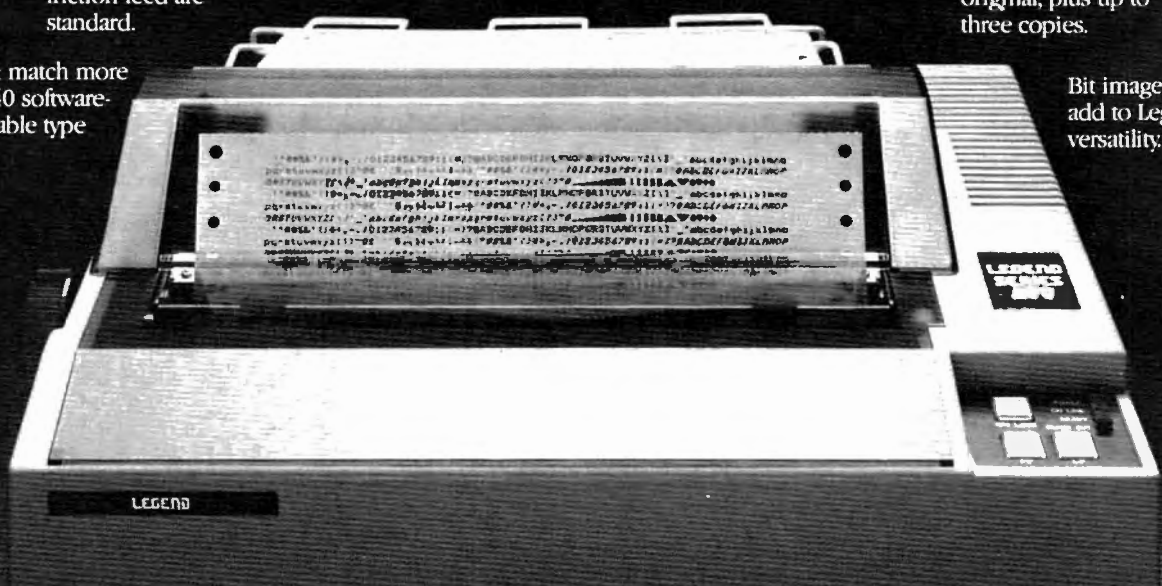
The Making Of A Legend.

Both tractor & friction feed are standard.

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

Prints a crisp original, plus up to three copies.

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

```
HJ 310 IF Q=3 AND T=49 THEN 320
OK 315 K=INT(RND(1)*(R-S+1))+S
CJ 320 F$=STR$(K):W=15
CD 325 IF K<L THEN W=22
NJ 330 GOSUB 3000
JD 335 W=22
CI 337 IF K<L THEN W=15
PI 340 F$=STR$(L)
NP 345 GOSUB 3000
LL 346 ON Q GOSUB 6000,6002,6004
GB 350 IF Q=1 THEN M=K+L
AL 360 IF Q=2 AND L<K THEN M=K-L
AN 362 IF Q=2 AND K<L THEN M=L-K
GI 365 IF Q=3 THEN M=K*L
PH 380 ? "{CLEAR}":GOSUB 740:MM=1:IF M>9 THEN MM=2
GF 385 IF M>99 THEN MM=3
GC 393 V=0
NK 395 FOR J=0 TO MM-1
KM 397 PLOT 40-J*6,30
BG 400 POKE 764,255:GET #1,P
CJ 401 IF (P<>88) AND (P<48 OR P>57) THEN 400
NM 407 IF P=88 AND ZZ=1 THEN 100
DP 408 IF P=88 AND TR=1 THEN ZZ=ZZ+1
IK 410 IF P=88 THEN GRAPHICS 17:SETCOLOR 0,12,10:? #6;"  PERCENTAGE=";INT(Z/(ZZ-1)*100):GOTO 120
EB 415 P=P-48:W=30
HK 417 COLOR 0:PLOT 40-J*6,30:COLOR 3
JH 420 V=V+INT((P*10^J)+0.1):X=40-6*J:GOSUB 480:NEXT J
NC 450 IF M=V THEN 470
AI 451 SOUND 2,200,12,12:FOR I=1 TO 100:NEXT I:SOUND 2,0,0,0
EM 452 COLOR 0:FOR Y=30 TO 35:FOR I=24 TO 42:PLOT I,Y:NEXT I:NEXT Y:COLOR 3
AL 456 IF TR=1 THEN 460
HH 458 TR=1:COLOR 0:GOSUB 1170:COLOR 3:GOSUB 770:GOTO 393
KB 460 M$=STR$(M):IF MM=3 THEN 462
CP 461 FOR I=1 TO 3-MM:READ A:NEXT I
MM 462 FOR OO=1 TO MM
IJ 464 P=VAL(M$(OO,OO))
NO 465 READ X:GOSUB 480:NEXT OO:RESTORE
MC 470 ? "{CLEAR}":COLOR 2:GOSUB 1170:IF TR=0 THEN GOSUB 2500:GOSUB 755:Z=Z+1:GOSUB 6500
FF 471 GOSUB 2225:POKE 198,0:GOTO 301
BK 480 COLOR 1:IF P=0 THEN GOSUB 720
PP 485 ON P GOSUB 500,525,530,555,585,610,633,660,680:RETURN
GD 500 PLOT X,W:DRAWTO X,W+4:PLOT X-1,W:DRAWTO X-1,W+4:RETURN
AO 525 PLOT X,W:DRAWTO X-3,W:PLOT X-1,W+1:PLOT X,W+1:PLOT X,W+2:DRAWTO X-3,W+2
PJ 527 PLOT X-3,W+3:PLOT X-2,W+3:PLOT X-3,W+4:DRAWTO X,W+4:RETURN
LD 530 PLOT X,W:DRAWTO X,W+4:PLOT X-1,W:DRAWTO X-1,W+4:PLOT X-3,W:PLOT X-2,W
LP 540 PLOT X-3,W+2:PLOT X-2,W+2:PLOT X-3,W+4:PLOT X-2,W+4:RETURN
EI 555 PLOT X-3,W:DRAWTO X-3,W+2:PLOT X-1,W+1:DRAWTO X-1,W+4:PLOT X,W+2:PLOT X-2,W+2:RETURN
KO 585 PLOT X-3,W:DRAWTO X,W:PLOT X-3,W+2:DRAWTO X,W+2:PLOT X-3,W+4:DRAWTO X,W+4
FP 590 PLOT X-3,W+1:PLOT X-2,W+1:PLOT X-1,W+3:PLOT X,W+3:RETURN
DD 610 PLOT X-3,W:DRAWTO X-3,W+4:PLOT X-1,W:PLOT X,W:PLOT X-2,W:DRAWTO X-2,W+4:PLOT X-1,W+2:PLOT X-1,W+4
HC 615 PLOT X,W+2:DRAWTO X,W+4:RETURN
GL 633 PLOT X,W:DRAWTO X-3,W:PLOT X,W+1:DRAWTO X-3,W+4:RETURN
OO 660 GOSUB 720:PLOT X-2,W+2:PLOT X-1,W+2:RETURN
IL 680 PLOT X-3,W+4:DRAWTO X,W+4:DRAWTO X,W:DRAWTO X-3,W:DRAWTO X-3,W+2
CE 685 DRAWTO X-1,W+2:RETURN
OD 720 PLOT X,W:DRAWTO X,W+4:DRAWTO X-3,W+4:DRAWTO X-3,W:DRAWTO X,W:RETURN
KL 740 FOR I=24 TO 42:PLOT I,28:NEXT I:RETURN
GH 755 ? "{12 SPACES}GOOD":RETURN
JK 770 ? "{10 SPACES}TRY AGAIN":RETURN
KE 1000 GRAPHICS 17:SETCOLOR 1,12,10:POSITION 2,2:? #6;"DO YOU WISH TO:"
JB 1010 POSITION 2,5:? #6;"1) PRACTICE TIMES":POSITION 2,6:? #6;"TABLES"
FH 1020 POSITION 2,8:? #6;"2) RANDOM NUMBERS"
BM 1030 POSITION 2,10:? #6;"(ENTER 1 OR 2)"
ML 1040 GET #1,T:IF T<49 OR T>50 THEN 1040
BE 1050 IF T=50 THEN 190
EE 1060 POSITION 2,12:? #6;"ENTER TIMES TABLES";
BL 1065 K=0:P=0:? #6;"  (1-14)  ";
MN 1070 GET #1,Z:IF (Z<48 OR Z>57) AND (Z<>155) THEN 1070
MN 1073 IF Z=155 THEN K=ZZ-48:GOTO 1090
MO 1075 ? #6;Z-48;
BB 1080 P=P+1:IF P=1 AND Z<>155 THEN K=(Z-48)*10:ZZ=Z:GOTO 1070
MN 1085 K=K+(Z-48):IF K>14 THEN ? #6:GOTO 1065
MG 1090 S=0:R=14:GOTO 263
GF 1100 COLOR 2:A=40:B=28:FOR I=0 TO 9
ME 1110 IF I/2=INT(I/2) THEN A=A+2:B=B-2
DN 1120 PLOT B,I:DRAWTO A,I:NEXT I
NF 1130 PLOT B,I:RETURN
IH 1170 FOR I=51 TO 55:PLOT I,2:NEXT I:FOR I=50 TO 55:PLOT I,3:NEXT I
CE 1180 FOR I=4 TO 7:FOR J=49 TO 55:PLOT J,I:NEXT J:NEXT I
NF 1190 COLOR 0:PLOT 54,3:RETURN
DI 1230 COLOR 2:Y=24:FOR X=Y TO Y+3:PLOT X,10:DRAWTO X,13:NEXT X
EM 1240 Y=40:FOR X=Y TO Y+3:PLOT X,10:DRAWTO X,13:NEXT X
FK 1250 PLOT 28,12:PLOT 28,13:PLOT 29,12:PLOT 29,13
GN 1260 PLOT 44,12:PLOT 44,13:PLOT 45,12:PLOT 45,13:RETURN
JJ 2225 COLOR 0:FOR Y=15 TO 35:FOR I=24 TO 42:PLOT I,Y:NEXT I:NEXT Y:COLOR 3:RETURN
AE 2500 COLOR 0:PLOT 54,7:PLOT 53,6:COLOR 3:RETURN
FA 3000 IF LEN(F$)>1 THEN 3030
ND 3015 P=VAL(F$(1,1))
BE 3020 X=40:GOSUB 480
```

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

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

The all new, 1984 Indus GT.™

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

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

Or an Apple into a Red Hot Apple.

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

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

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

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

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

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

Parks like a Beetle.
The GT's small, sleek, condensed size makes it easy to park.

And its low price makes it easy to buy. $449 for Atari. $329 for Apple.

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

The drive will be well worth it.

INDUS™

The all-new 1984 Indus GT Disk Drive.
The most advanced, most handsome disk drive in the world.

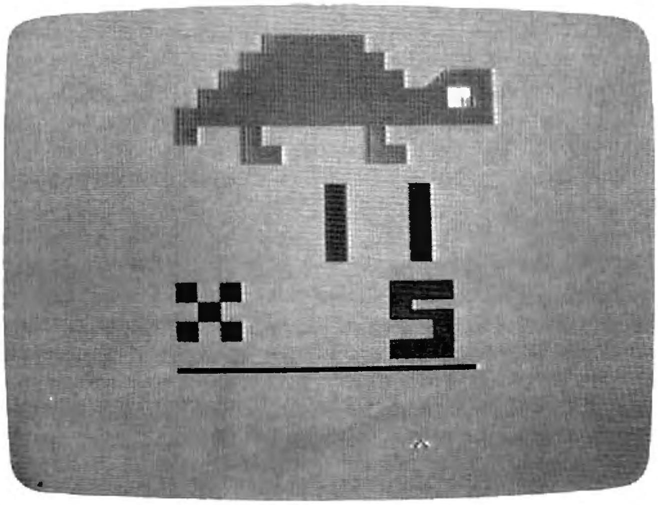
For dealer information, call 1-800-33-INDUS. In California, 1-800-54-INDUS, 213/882-9600.

```
KK 3Ø25  RETURN
NA 3Ø3Ø  P=VAL(F$(1,1))
BN 3Ø35  X=34:GOSUB 48Ø
ND 3Ø4Ø  P=VAL(F$(2,2))
BL 3Ø45  X=4Ø:GOSUB 48Ø
KI 3Ø5Ø  RETURN
JO 6ØØØ  PLOT 27,24:DRAWTO 27,26:PLOT 2
         6,25:DRAWTO 28,25:RETURN
CE 6ØØ2  PLOT 26,25:DRAWTO 28,25:RETURN

PD 6ØØ4  PLOT 26,24:PLOT 28,24:PLOT 27,
         25:PLOT 26,26:PLOT 28,26:RETUR
         N
JO 65ØØ  SOUND 2,15Ø,1Ø,1Ø:FOR I=1 TO 5
         Ø:NEXT I:SOUND 2,125,1Ø,12:FOR
          I=1 TO 5Ø:NEXT I:SOUND 2,Ø,Ø,
         Ø:RETURN
DJ 651Ø  REM SOUND
GP 8ØØØ  DATA 28,34,4Ø
```

Program 4: Snertle For TI-99/4A

```
1ØØ GOTO 15Ø
11Ø FOR I=1 TO LEN(H$)
12Ø CALL HCHAR(ROW,COL+I,ASC(SEG$(H
    $,I,1)))
13Ø NEXT I
14Ø RETURN
15Ø GOSUB 271Ø
16Ø CALL CLEAR
17Ø CALL SCREEN(12)
18Ø PRINT TAB(5);"** S N E R T L E
    **"::::
19Ø PRINT "SELECT ONE:"::
2ØØ PRINT TAB(3);"1) ADDITION"::
21Ø PRINT TAB(3);"2) SUBTRACTION"::
22Ø PRINT TAB(3);"3) MULTIPLICATION
    "::
23Ø PRINT TAB(3);"4) END PROGRAM"::
    ::::
24Ø PRINT "(ENTER 1, 2, 3, OR 4)";
25Ø CALL KEY(Ø,Q,ST)
26Ø IF ST=Ø THEN 25Ø
27Ø Q=Q-48
28Ø IF (Q>4)+(Q<1)THEN 25Ø
29Ø KOL=Q
3ØØ IF Q<>2 THEN 32Ø
31Ø KOL=1Ø
32Ø CALL COLOR(11,KOL+4,1)
```



"Snertle," TI version.

```
33Ø C=14
34Ø IF (Q<>1)*(Q<>2)THEN 36Ø
35Ø C=99
36Ø IF Q=3 THEN 221Ø
37Ø IF Q=4 THEN 31ØØ
38Ø CALL CLEAR
39Ø CALL SCREEN(4)
4ØØ PRINT TAB(4);"ENTER LARGEST VAL
    UE:"::
41Ø PRINT " (LOWEST :1  HIGHEST:";C
    ;")"::
42Ø INPUT R
43Ø IF (R<1)+(R>C)THEN 42Ø
44Ø PRINT ::
45Ø PRINT TAB(4);"ENTER SMALLEST VA
    LUE"::
46Ø PRINT " (LOWEST :Ø  HIGHEST:";R
    ;")"::
47Ø INPUT S
48Ø IF (S<Ø)+(S>R)THEN 47Ø
49Ø CALL CLEAR
5ØØ CALL SCREEN(1Ø)
51Ø PRINT "PRESS 'X' TO RETURN TO M
    ENU"::::::::::
52Ø FOR I=1 TO 4ØØ
53Ø NEXT I
54Ø CALL CLEAR
55Ø CALL SCREEN(12)
56Ø Z=Ø
57Ø ZZ=Ø
58Ø GOSUB 241Ø
59Ø GOSUB 251Ø
6ØØ GOSUB 258Ø
61Ø TR=Ø
62Ø ZZ=ZZ+1
63Ø RANDOMIZE
64Ø L=INT(RND*(R-S+1))+S
65Ø IF (Q=3)*(T=1)THEN 67Ø
66Ø K=INT(RND*(R-S+1))+S
67Ø F$=STR$(K)
68Ø Y=9
69Ø W=15
7ØØ IF K>=L THEN 72Ø
71Ø Y=14
72Ø GOSUB 284Ø
73Ø Y=14
74Ø IF L<=K THEN 76Ø
75Ø Y=9
76Ø F$=STR$(L)
77Ø GOSUB 284Ø
78Ø ON Q GOSUB 296Ø,296Ø,3Ø4Ø
79Ø IF Q<>1 THEN 81Ø
8ØØ M=K+L
81Ø IF (Q<>2)+(K<L)THEN 83Ø
82Ø M=K-L
83Ø IF (Q<>2)+(K>=L)THEN 85Ø
84Ø M=L-K
85Ø IF Q<>3 THEN 87Ø
86Ø M=K*L
87Ø CALL HCHAR(18,9,1Ø4,14)
88Ø MM=1
89Ø IF M<=9 THEN 91Ø
9ØØ MM=2
91Ø IF M<=99 THEN 93Ø
92Ø MM=3
93Ø V=Ø
94Ø GOSUB 241Ø
95Ø FOR J=Ø TO MM-1
96Ø CALL HCHAR(22,2Ø-4*J,94)
97Ø CALL KEY(Ø,K1,ST)
98Ø IF ST=Ø THEN 97Ø
```

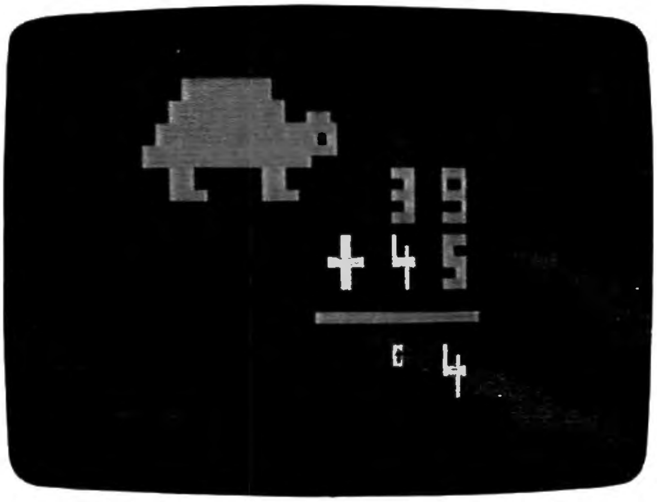
```
990 IF ((K1<48)+(K1>57))*(K1<>88)TH
    EN 970
1000 IF (K1=88)*(ZZ=1)THEN 160
1010 IF K1<>88 THEN 1060
1020 CALL CLEAR
1030 PRINT TAB(3);"PERCENTAGE :";IN
     T(Z/(ZZ-1)*100)
1040 PRINT ::::
1050 GOTO 190
1060 CALL HCHAR(22,20-4*J,32)
1070 P=K1-48
1080 V=V+(P*10^J)
1090 X=19-4*J
1100 Y=20
1110 GOSUB 1430
1120 NEXT J
1130 IF M=V THEN 1310
1140 CALL SOUND(300,110,2)
1150 FOR I=20 TO 24
1160 CALL HCHAR(I,1,32,30)
1170 NEXT I
1180 IF TR=1 THEN 1230
1190 TR=1
1200 GOSUB 2660
1210 GOSUB 2010
1220 GOTO 930
1230 M$=STR$(M)
1240 FOR OO=1 TO MM
1250 P=VAL(SEG$(M$,OO,1))
1260 X=19-(MM-OO)*4
1270 GOSUB 1430
1280 NEXT OO
1290 FOR T=1 TO 400
1300 NEXT T
1310 GOSUB 2510
1320 IF TR<>0 THEN 1390
1330 CALL HCHAR(5,23,136)
1340 GOSUB 1950
1350 Z=Z+1
1360 CALL SOUND(200,196,2)
1370 CALL SOUND(200,262,2)
1380 CALL SOUND(200,294,2)
1390 FOR I=9 TO 24
1400 CALL HCHAR(I,2,32,30)
1410 NEXT I
1420 GOTO 610
1430 IF P<>0 THEN 1460
1440 GOSUB 1920
1450 RETURN
1460 ON P GOSUB 1480,1500,1550,1600
     ,1650,1710,1790,1850,1890
1470 RETURN
1480 CALL VCHAR(Y,X+1,115,4)
1490 RETURN
1500 GOSUB 2190
1510 GOSUB 2160
1520 CALL HCHAR(Y+2,X,115)
1530 GOSUB 2140
1540 RETURN
1550 GOSUB 2190
1560 GOSUB 2160
1570 CALL HCHAR(Y+2,X+2,115)
1580 GOSUB 2140
1590 RETURN
1600 CALL VCHAR(Y,X,115,2)
1610 CALL HCHAR(Y+2,X,115,3)
1620 CALL HCHAR(Y+1,X+1,114)
1630 CALL HCHAR(Y+3,X+1,114)
1640 RETURN
1650 GOSUB 2190
1660 CALL HCHAR(Y+1,X,115)
1670 CALL HCHAR(Y+1,X+1,112,2)
1680 CALL HCHAR(Y+2,X+2,115)
1690 GOSUB 2140
1700 RETURN
1710 GOSUB 2190
1720 CALL HCHAR(Y+2,X+2,115)
1730 CALL HCHAR(Y+1,X,115)
1740 CALL HCHAR(Y+1,X+1,112,2)
1750 CALL HCHAR(Y+2,X,115)
1760 CALL HCHAR(Y+2,X+2,115)
1770 GOSUB 2140
1780 RETURN
1790 GOSUB 2190
1800 CALL HCHAR(Y+1,X+2,115)
1810 CALL HCHAR(Y+2,X+1,115)
1820 CALL HCHAR(Y+2,X+2,113)
1830 CALL HCHAR(Y+3,X+1,115)
1840 RETURN
1850 GOSUB 1500
1860 CALL HCHAR(Y+1,X,115)
1870 CALL HCHAR(Y+2,X+2,115)
1880 RETURN
1890 GOSUB 1850
1900 CALL HCHAR(Y+2,X,32)
1910 RETURN
1920 GOSUB 1850
1930 CALL HCHAR(Y+1,X+1,32)
1940 RETURN
1950 H$="GOOD!"
1960 ROW=3
1970 COL=12
1980 GOSUB 110
1990 RETURN
2000 REM  CORRECT
2010 H$="TRY"
2020 ROW=2
2030 COL=13
2040 GOSUB 110
2050 H$="AGAIN"
2060 ROW=3
2070 COL=12
2080 GOSUB 110
2090 FOR I=1 TO 200
2100 NEXT I
2110 RETURN
2120 CALL VCHAR(Y,X,115,4)
2130 RETURN
2140 CALL HCHAR(Y+3,X,115,3)
2150 RETURN
2160 CALL HCHAR(Y+1,X,112,2)
2170 CALL HCHAR(Y+1,X+2,115)
2180 RETURN
2190 CALL HCHAR(Y,X,115,3)
2200 RETURN
2210 CALL CLEAR
2220 CALL SCREEN(4)
2230 PRINT "DO YOU WISH TO PRACTICE
     :"::::
2240 PRINT TAB(3);"1) TIMES TABLES,
      OR"::
2250 PRINT TAB(3);"2) RANDOM NUMBER
     S ?"::::::::
2260 PRINT TAB(5);"(ENTER 1 OR 2)"
2270 CALL KEY(0,K1,ST)
2280 IF ST=0 THEN 2270
2290 IF (K1<>49)*(K1<>50)THEN 2270
2300 T=K1-48
2310 IF T=2 THEN 380
2320 CALL CLEAR
2330 PRINT TAB(6);"ENTER TIMES TABL
     E"::
```

```
2340 PRINT TAB(6);"(ENTER 1 TO 14)"
     ::
2350 INPUT K
2360 IF (K<1)+(K>14)THEN 2350
2370 S=0
2380 R=14
2390 GOTO 490
2400 REM  DRAW THE SHELL
2410 R5=5
2420 COL=13
2430 FOR I=1 TO 4
2440 CALL HCHAR(I,COL,96,R5)
2450 R5=R5+2
2460 COL=COL-1
2470 NEXT I
2480 CALL HCHAR(5,9,96,12)
2490 RETURN
2500 REM  DRAW THE HEAD
2510 CALL HCHAR(3,21,97)
2520 CALL HCHAR(3,22,96,2)
2530 CALL HCHAR(4,21,96,3)
2540 CALL HCHAR(4,22,128)
2550 CALL HCHAR(5,21,96,3)
2560 RETURN
2570 REM  DRAW THE FEET AND TAIL
2580 FOR I=1 TO 8
2590 READ R5,C
2600 CALL HCHAR(R5,C,96)
2610 NEXT I
2620 RESTORE
2630 DATA 6,9,6,12,6,18,7,12,7,13,7
     ,18,7,19,5,22
2640 RETURN
2650 REM  ERASE THE HEAD
2660 FOR I=3 TO 5
2670 CALL HCHAR(I,21,32,3)
2680 NEXT I
2690 RETURN
2700 REM  DEFINE CHARS & COLORS
2710 CALL CHAR(96,"FFFFFFFFFFFFFFFF
     ")
2720 CALL CHAR(97,"0103070F1F3F7FFF
     ")
2730 CALL CHAR(104,"000000FFFF00000
     0")
2740 CALL CHAR(128,"000000000F0F0F0
     F")
2750 CALL CHAR(136,"3030180C0703000
     0")
2760 CALL COLOR(9,3,1)
2770 CALL COLOR(13,6,16)
2780 CALL COLOR(14,14,3)
2790 CALL CHAR(112,"00000000FFFFFFF
     F")
2800 CALL CHAR(113,"F0F0F0F0F0F0F0F
     0")
2810 CALL CHAR(114,"070707070707070
     7")
2820 CALL CHAR(115,"FFFFFFFFFFFFFFF
     F")
2830 RETURN
2840 IF LEN(F$)=2 THEN 2890
2850 P=VAL(SEG$(F$,1,1))
2860 X=W+4
2870 GOSUB 1430
2880 RETURN
2890 P=VAL(SEG$(F$,1,1))
2900 X=W
2910 GOSUB 1430
2920 P=VAL(SEG$(F$,2,1))
2930 X=W+4
2940 GOSUB 1430
2950 RETURN
2960 CALL VCHAR(14,11,115,3)
2970 CALL HCHAR(15,10,115)
2980 CALL HCHAR(15,12,115)
2990 IF Q=2 THEN 3010
3000 RETURN
3010 CALL HCHAR(14,11,32)
3020 CALL HCHAR(16,11,32)
3030 RETURN
3040 CALL HCHAR(14,9,115)
3050 CALL HCHAR(14,11,115)
3060 CALL HCHAR(15,10,115)
3070 CALL HCHAR(16,9,115)
3080 CALL HCHAR(16,11,115)
3090 RETURN
3100 END
```

Program 5: Snertle For The Color Computer

```
100 CLS(1):B$=CHR$(32)
110 PRINT@74,"**SNERTLE**"
120 PRINT@138,"SELECT 1"
130 PRINT@202,"1) ADDITION"
140 PRINTTAB(10)"2) SUBTRACTION"
150 PRINTTAB(10)"3) MULTIPLICATION"
155 PRINTTAB(10)"4) END"
160 PRINTTAB(10)"(ENTER 1,2,3 OR 4)
    ";:INPUTQ:IF Q>4 OR Q<1 THEN 16
    0
185 C=14:IF Q=1 OR Q=2 THEN C=99
187 IF Q=3 THEN 1000
188 IF Q=4 THEN END
190 CLS(1):PRINT@37,"ENTER LARGEST
    VALUE"
200 PRINTTAB(5)"(MIN.:1  MAX.:";C;"
    )";:INPUTR:IF R<1 OR R>C THEN 2
    00
230 PRINT@133,"ENTER SMALLEST VALUE
    "
240 PRINTTAB(5)"(MIN.:0  MAX.:";R;"
    )";:INPUTS:IF S<0 OR S>R THEN 2
    40
263 CLS:PRINT@227,"PRESS X TO RETUR
    N TO MENU";:FORI=1TO750:NEXTI:C
    LS(0)
270 Z=0:ZZ=0
275 GOSUB 1100:GOSUB 1170:GOSUB1230
301 TR=0:ZZ=ZZ+1
305 L=INT(RND(R-S)+S)
310 IF Q=3ANDT=1THEN320
315 K=INT(RND(R-S)+S)
320 F$=STR$(K):W=0
325 IF K<L AND Q=2  THEN TR=0:GOTO3
    05
330 W=0:GOSUB3000
335 W=64
340 F$=STR$(L)
345 W=96:GOSUB 3000
346 ON Q GOSUB 6000,6000,6004
350 IF Q=1 THEN M=K+L
355 IF Q=2 THEN M=K-L
360 IF Q=3 THEN M=K*L
380 MM=1:IF M>9 THEN MM=2
385 IF M>99 THEN MM=3
390 GOSUB 740
393 V=0:GOSUB 1100
395 FOR J=0 TO MM-1
397 POKE 1466-(4*J),94
399 HH$=INKEY$
400 H$=INKEY$
405 IF H$=""THEN 400
410 IF H$="X" AND ZZ=1 THEN 100
```

```
411 IF H$="X" THEN CLS(1):PRINT@68,
    "YOUR PERCENTAGE IS ";INT(Z/(ZZ
    -1)*100):GOTO120
413 IF H$<>"0" AND VAL(H$)=0 THEN 4
    00
415 P=VAL(H$)
420 V=V+(P*10^J):X=1466-(4*J):GOSUB
    480:NEXTJ
450 IF INT(M)=INT(V) THEN 470
451 SOUND 80,6:FORI=1TO20:NEXTI:SOU
    ND 80,6:FORI=1TO20:NEXTI:SOUND6
    0,12
452 FOR I=1439 TO 1535:POKEI,128:NE
    XT I
456 IF TR=1 THEN 460
458 TR=1:GOSUB 1500:GOSUB 770:GOTO3
    93
460 M$=STR$(M)
461 FORI=1 TO 11-MM:READA:NEXTI
462 FOR OO=1 TO MM
464 P=VAL(MID$(M$,(OO+1),1))
465 READX:GOSUB 480:NEXT OO:RESTORE
470 GOSUB1170:IF TR=0THEN GOSUB 250
    0: GOSUB 755:Z=Z+1:GOSUB 6500
471 GOSUB 2225:GOTO301
480 IF P=0 THEN 720
485 ON P GOSUB 500,525,555,585,610,
    633,660,680,700:RETURN
500 POKEX,143:POKEX+32,143:POKEX+64
    ,140:RETURN
525 POKEX,140:POKEX+1,143:POKEX+33,
    140:POKEX+32,143:POKEX+64,140:P
    OKEX+65,140
530 RETURN
555 POKEX,140:POKEX+32,140:POKEX+64
    ,140:POKEX+65,140
560 POKE X+1,143:POKEX+33,143:RETUR
    N
585 POKEX,138:POKEX+32,140:POKEX+1,
    130:POKEX+33,142
590 POKEX+64,128:POKEX+65,136:RETUR
    N
610 POKEX,143:POKEX+32,140:POKEX+64
    ,140
615 POKEX+1,140:POKEX+33,143:POKEX+
    65,140:RETURN
633 POKEX,143:POKEX+32,143:POKEX+64
    ,140:POKEX+1,140
635 POKE X+33,141:POKEX+65,140:RETU
    RN
660 POKE X,140:POKEX+32,129:POKEX+6
    4,132
670 POKEX+65,128:POKE X+1,141:POKEX
    +33,138:RETURN
680 POKEX,142:POKEX+32,142:POKEX+64
    ,140:POKEX+65,140
685 POKEX+1,141:POKEX+33,141:RETURN
700 POKEX,142:POKEX+32,140:POKEX+64
    ,140
710 POKEX+1,141:POKEX+33,141:POKEX+
    65,140:RETURN
720 POKEX,142:POKEX+1,141:POKEX+32,
    138:POKEX+33,133
725 POKEX+64,140:POKEX+65,140:RETUR
    N
740 FORI=1392 TO 1404:POKEI,131:NEX
    TI:RETURN
755 PRINT@103,"GOOD";:FORI=1TO500:N
    EXTI:RETURN
770 PRINT@72,"TRY";:PRINT@103,"AGAI
    N";:FOR I=1 TO 500:NEXTI:RETURN
```



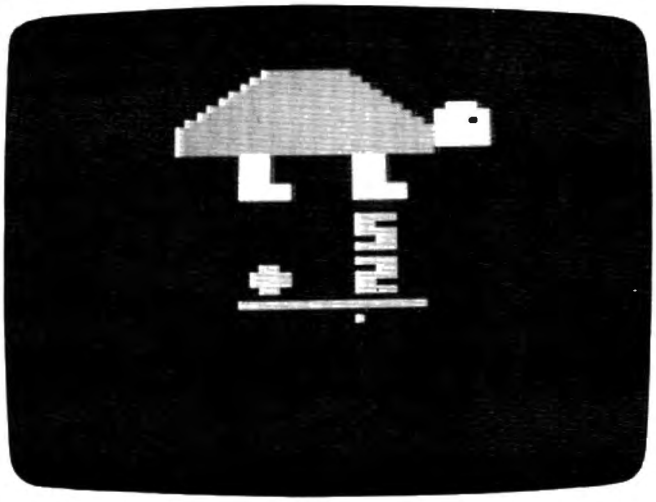
"Snertle," Color Computer version.

```
1000 CLS(1):PRINT@66,"DO YOU WISH T
     O:"
1010 PRINT@130,"1) PRACTICE TIMES T
     ABLES"
1020 PRINT@162,"2) RANDOM NUMBERS"
1030 PRINT@224,"(ENTER 1 OR 2)";:IN
     PUTT:IF T<1 OR T>2 THEN 1030
1050 IF T=2 THEN 1090
1060 CLS(1):PRINT@66,"ENTER TIMES T
     ABLE"
1070 PRINT@100,"(1-14)";:INPUT K:IF
      K<1 OR K>14 THEN 1070
1090 S=0:R=14:GOTO 263
1100 FOR I=1056 TO 1152 STEP 32
1110 READ A,B
1120 FOR J=1TOB
1130 POKEI+J+A,143
1140 NEXTJ:NEXTI:RESTORE:RETURN
1170 POKE 1169,140:POKE1167,140:POK
     E1168,140
1180 POKE 1103,129:POKE1104,131:POK
     E1105,130
1190 POKE1135,143:POKE1136,142:POKE
     1137,143:RETURN
1230 POKE 1196,143:POKE1197,143:POK
     E1189,143:POKE1190,143
1240 POKE 1228,140:POKE1229,140:POK
     E1230,140:POKE1221,140:POKE122
     2,140:POKE1223,140:RETURN
1500 FORI=1103 TO 1167 STEP 32:FOR
     J=0 TO 3:POKE I+J,128:NEXTJ:NE
     XTI:POKE 1167,143:RETURN
2225 FOR I=1140 TO 1236 STEP 32
2230 FOR J=1 TO 11:POKEJ+I,128:NEXT
     J:NEXTI:FOR I=1260 TO 1535 STE
     P 32
2235 FOR J=1 TO 16:POKE J+I,128:NEX
     TJ:NEXTI:RETURN
2500 POKE 1167,139:RETURN
3000 IF LEN(F$)>2 THEN 3030
3015 P=VAL(MID$(F$,2,1))
3020 X=1210+W:GOSUB480
3025 RETURN
3030 P=VAL(MID$(F$,2,1))
3035 X=1206+W:GOSUB480
3040 P=VAL(MID$(F$,3,1))
3045 X=1210+W:GOSUB480
3050 RETURN
5000 DATA 5,7,4,9,3,11,2,13,1458,14
     62,1466
```

```
6000 POKE 1298,143:POKE1330,143:POK
     E 1362,140:POKE 1331,140:POKE1
     329,140
6001 IF Q=2 THEN POKE 1298,128:POKE
     1330,140:POKE1362,128
6003 RETURN
6004 POKE 1297,131:POKE1299,131:POK
     E1330,140:POKE1329,131:POKE133
     1,131:RETURN
6500 SOUND 100,7:SOUND130,10
6510 RETURN
```

Program 6: Snertle For Apple

```
110  TEXT : HOME : VTAB 2: HTAB 15: PRINT
     "**SNERTLE**": VTAB 5
120  PRINT : VTAB 5: HTAB 10: PRINT "SE
     LECT ONE:"
130  PRINT : PRINT : HTAB 10: PRINT "1)
      ADDITION"
140  PRINT : HTAB 10: PRINT "2) SUBTRAC
     TION"
150  PRINT : HTAB 10: PRINT "3) MULTIPL
     ICATION"
155  PRINT : HTAB 10: PRINT "4) END PRO
     GRAM"
160  PRINT : PRINT : HTAB 10: PRINT "(E
     NTER 1,2,3 OR 4) ";: INPUT Q: IF Q
      < 1 OR Q > 4 THEN 160
185 C = 14: IF Q = 1 OR Q = 2 THEN C =
     99
187  IF Q = 3 THEN 1000
188  IF Q = 4 THEN  END
190  HOME : VTAB 3: HTAB 10: PRINT "ENT
     ER LARGEST VALUE"
200  HTAB 10: PRINT "(MIN.:1  MAX.:";C;
     ")";: INPUT R: IF R < 1 OR R > C THEN
     200
230  HTAB 10: VTAB 10: PRINT "ENTER SMA
     LLEST VALUE"
240  HTAB 10: PRINT "(MIN.:0  MAX.:";R;
     ")";: INPUT S: IF S < 0 OR S > R THEN
     240
263  HOME : VTAB 10: HTAB 7: PRINT "TYP
     E ";: INVERSE : PRINT "X";: NORMAL
     : PRINT " TO RETURN TO THE MENU"
265  FOR I = 1 TO 2000: NEXT I: HOME
270 Z = 0:ZZ = 0: GR
275  GOSUB 1100: COLOR= 12: GOSUB 1170:
      GOSUB 1230
301 TR = 0:ZZ = ZZ + 1
305 L =  INT ( RND (1) * (R - S + 1)) +
     S
310  IF Q = 3 AND T = 1 THEN 320
315 K =  INT ( RND (1) * (R - S + 1)) +
     S
320 F$ =  STR$ (K):W = 0
325  IF K < L AND Q = 2 THEN 305
330 W = 0: GOSUB 3000
340 F$ =  STR$ (L)
345 W = 6: GOSUB 3000
346  ON Q GOSUB 6000,6000,6004
350  IF Q = 1 THEN M = K + L
355  IF Q = 2 THEN M = K - L
365  IF Q = 3 THEN M = K * L
380  GOSUB 740:MM = 1: IF M > 9 THEN MM
      = 2
385  IF M > 99 THEN MM = 3
393 V = 0: COLOR= 12: GOSUB 1170
395  FOR J = 0 TO MM - 1
397  COLOR= 1: PLOT 21 - (5 * J),34
399  POKE  - 16368,0
400 H$ = "":H =  PEEK ( - 16384) - 128:
      IF H > 0 THEN H$ =  CHR$ (H)
```



"Snertle," Apple version.

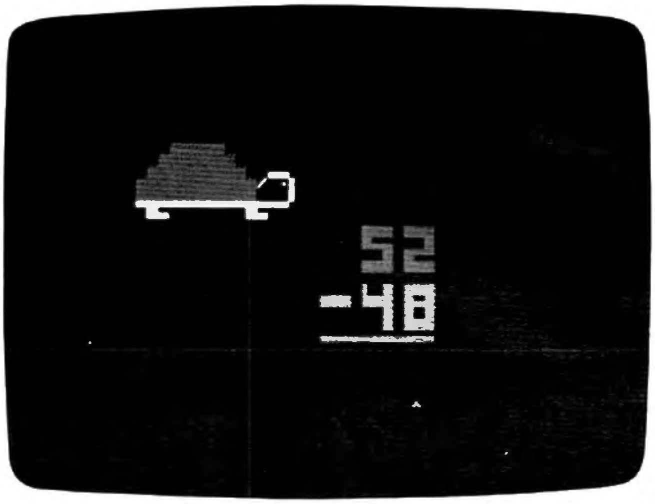
```
407  IF H$ = "X" AND ZZ = 1 THEN  POKE
      - 16368,0: GOTO 110
410  IF H$ = "X" THEN  TEXT : HOME : HTAB
     15: PRINT "PERCENTAGE="; INT (Z /
     (ZZ - 1) * 100): POKE  - 16368,0: GOTO
     120
412  IF H < 48 OR H > 57 THEN 400
415 P =  VAL (H$)
420 V = V + (P * 10 ^ J):W = 14:X = 21 -
     (5 * J): GOSUB 480: NEXT J
450  IF M = V THEN 470
451  FOR I = 1 TO 40: FOR J = 1 TO 2: NEXT
     J:L =  PEEK ( - 16336): NEXT I
452  COLOR= 0: FOR I = 33 TO 38: HLIN 7
     ,34 AT I: NEXT I: COLOR= 1
456  IF TR = 1 THEN 460
458 TR = 1: COLOR= 0: GOSUB 1170: GOSUB
     770:V = 0: GOTO 395
460 M$ =  STR$ (M)
461  IF MM < 3 THEN  FOR I = 1 TO 3 - M
     M: READ X: NEXT I
462  FOR OO = 1 TO MM
464 P =  VAL ( MID$ (M$,OO,1))
465  READ X: GOSUB 480: NEXT OO: RESTORE

467  FOR I = 1 TO 900: NEXT
470  COLOR= 12: GOSUB 1170: IF TR = 0 THEN
      GOSUB 2500: GOSUB 755:Z = Z + 1: GOSUB
     6500: HOME
471  GOSUB 2225: GOTO 301
480  COLOR= 1: IF P = 0 THEN  GOSUB 720

485  ON P GOSUB 500,525,555,585,610,633
     ,660,680,700: RETURN
500  VLIN 20 + W,24 + W AT X: VLIN 20 +
     W,24 + W AT X + 1: RETURN
525  HLIN X,X + 3 AT 20 + W: PLOT X + 2
     ,21 + W: PLOT X + 3,21 + W: HLIN X
     ,X + 3 AT 22 + W
530  VLIN 23 + W,24 + W AT X: VLIN 23 +
     W,24 + W AT X + 1: PLOT X + 2,24 +
     W: PLOT X + 3,24 + W: RETURN
555  VLIN 20 + W,24 + W AT X + 2: PLOT
     X,20 + W: PLOT X,22 + W: PLOT X,24
      + W
560  PLOT X + 1,20 + W: PLOT X + 1,22 +
     W: PLOT X + 1,24 + W: RETURN
585  VLIN 20 + W,22 + W AT X: PLOT X +
     1,22 + W: VLIN 20 + W,24 + W AT X +
     2: PLOT X + 3,22 + W: RETURN
```

```
610  HLIN X,X + 3 AT 20 + W: HLIN X,X +
     3 AT 22 + W: HLIN X,X + 3 AT 24 +
     W: PLOT X + 2,23 + W: PLOT X + 3,2
     3 + W
615  PLOT X,21 + W: PLOT X + 1,21 + W: RETU
     RN
633  VLIN 20 + W,24 + W AT X: VLIN 20 +
     W,24 + W AT X + 1: VLIN 22 + W,24 +
     W AT X + 3: HLIN X + 2,X + 3 AT 20
      + W
635  PLOT X + 2,22 + W: PLOT X + 2,24 +
     W: RETURN
660  HLIN X + 1,X + 3 AT 20 + W: PLOT X
      + 3,21 + W: PLOT X + 2,22 + W
665  VLIN 23 + W,24 + W AT X + 1: RETURN

680  GOSUB 720: HLIN X + 1,X + 2 AT 22 +
     W: RETURN
700  HLIN X,X + 3 AT 20 + W: HLIN X,X +
     3 AT 22 + W: HLIN X,X + 3 AT 24 +
     W: VLIN 20 + W,24 + W AT X + 3
705  VLIN 21 + W,22 + W AT X: RETURN
720  VLIN 20 + W,24 + W AT X: VLIN 20 +
     W,24 + W AT X + 3: HLIN X + 1,X +
     2 AT 20 + W: HLIN X + 1,X + 2 AT 2
     4 + W: RETURN
740  HLIN 10,27 AT 32: RETURN
755  VTAB 21: HTAB 19: PRINT "GOOD!": FOR
     I = 1 TO 300: NEXT I: RETURN
770  VTAB 21: HTAB 16: PRINT "TRY AGAIN
     ": FOR I = 1 TO 1000: NEXT I: HOME
     : RETURN
1000  HOME : VTAB 4: HTAB 13: PRINT "DO
      YOU WISH TO:"
1010  PRINT : HTAB 9: PRINT "1) PRACTIC
     E TIMES TABLES"
1020  PRINT : HTAB 9: PRINT "2) PRACTIC
     E RANDOM NUMBERS"
1030  PRINT : HTAB 9: PRINT "(ENTER 1 O
     R 2) ";: INPUT T: IF T < 0 OR T >
     2 THEN 1030
1050  IF T = 2 THEN 190
1060  HOME : VTAB 5: HTAB 11: PRINT "EN
     TER TIMES TABLE (1-14)"
1070  INPUT K: IF K < 1 OR K > 14 THEN
     1070
1090 S = 0:R = 14: GOTO 263
1100 J = 12:JJ = 20: COLOR= 4: FOR I =
     0 TO 8: HLIN J,JJ AT I:J = J - 1:J
     J = JJ + 1
1110  NEXT I: FOR I = 8 TO 11: HLIN J +
     1,JJ - 1 AT I: NEXT I: RETURN
1170  HLIN 30,32 AT 5: FOR I = 6 TO 10:
      HLIN 29,33 AT I: NEXT I: COLOR= 0
     : PLOT 32,7: RETURN
1230  COLOR= 12: FOR I = 12 TO 15: HLIN
     10,12 AT I: HLIN 21,23 AT I: NEXT
     I
1240  FOR I = 16 TO 17: HLIN 10,14 AT I
     : HLIN 21,25 AT I: NEXT I: RETURN
2225  COLOR= 0: FOR I = 20 TO 38: HLIN
     10,39 AT I: NEXT I: COLOR= 1: RETURN
2500  COLOR= 0: PLOT 32,10: PLOT 31,9: COLO
      R= 1: RETURN
3000  IF  LEN (F$) > 1 THEN 3030
3015 P =  VAL ( MID$ (F$,1,1))
3020 X = 21: GOSUB 480
3025  RETURN
3030 P =  VAL ( MID$ (F$,1,1))
3035 X = 16: GOSUB 480
3040 P =  VAL ( MID$ (F$,2,1))
3045 X = 21: GOSUB 480
3050  RETURN
5000  DATA   12,16,22
6000  HLIN 11,14 AT 29: HLIN 11,14 AT 2
     8: IF Q = 1 THEN  VLIN 27,30 AT 12
     : VLIN 27,30 AT 13
6001  RETURN
6004  PLOT 12,27: PLOT 14,27: PLOT 13,2
     8: PLOT 12,29: PLOT 14,29: RETURN
6500  FOR I = 1 TO 20:L =  PEEK ( - 163
     36): NEXT I: FOR I = 1 TO 10: NEXT
     I: FOR I = 1 TO 40:L =  PEEK ( - 1
     6336): NEXT I: RETURN
```



"Snertle," PC/PCjr version.

Program 7: Snertle For PC/PCjr

```
10 DEF SEG=0:POKE 1047,192
20 SCREEN 0,1:WIDTH 40:KEY OFF
25 S$=CHR$(219):D$=CHR$(31):L$=CHR$(29):
R$=CHR$(28):U$=CHR$(30):TB$=CHR$(223):BB
$=CHR$(220):LB$=CHR$(221):RB$=CHR$(222):
SP$=CHR$(32)
100 B$=CHR$(13):C$=CHR$(9)
110 COLOR 12:CLS:LOCATE 24,9,0:PRINT"***
**** SNERTLE *******"
120 PRINT B$B$B$B$B$C$"       SELECT ONE:
"
130 COLOR 2:PRINT B$C$"1) ADDITION"
140 COLOR 4:PRINT B$C$"2) SUBTRACTION"
150 COLOR 6:PRINT B$C$"3) MULTIPLICATION
"
155 COLOR 14:PRINT B$C$"4) END PROGRAM"
160 PRINT B$B$B$B$C$"(ENTER 1,2,3 OR 4)"
;
170 Q$=INKEY$:X=RND(1):Q=VAL(Q$):IF Q<1
OR Q>4 THEN 170
175 C=14:IF Q=1 OR Q=2 THEN C=99
185 C=14:IF Q=1 OR Q=2 THEN C=99
187 IF Q=3 THEN 1000
188 IF Q=4 THEN END
190 CLS:LOCATE 10,12:PRINT "ENTER LARGES
T VALUE"
200 PRINT:PRINT   "(MIN.:0 MAX.:";C;")";
:INPUT R:IF R<0 OR R>C THEN PRINT U$U$U$
:GOTO 200
230 PRINT:PRINT  "ENTER SMALLEST VALUE"
240 PRINT:PRINT   "(MIN.:0 MAX.:";R;")";
:INPUT S :IF S<0 OR S>R THEN PRINT U$U$U
$:GOTO 240
263 CLS:LOCATE 12,5:PRINT "PRESS ' X ' T
O RETURN TO MENU":FOR I =1 TO 1000 :NEXT
 I
```

```
265 CLS
270 Z=0:ZZ=0
275 COLOR 2:GOSUB 1100:GOSUB 1170:GOSUB
1230:GOSUB 1260: COLOR Q *2
301 TR=0:ZZ=ZZ+1
305 L=INT(RND(1)*(R-S+1))+S
310 IF Q=3 AND T=1 THEN 320
315 K=INT(RND(1)*(R-S+1))+S
320 F$=STR$(K):W=0
325 IF K<L THEN W=5
330 GOSUB 3000
335 W=5
337 IF L>K THEN W=0
340 F$= STR$(L)
345 GOSUB 3000
346 ON Q GOSUB 6000,6000,6004
350 IF Q=1 THEN M=K+L
355 IF Q=2 AND K>=L THEN M=K-L
360 IF Q=2 AND K<L THEN M=L-K
365 IF Q=3 THEN M=K*L
380 GOSUB 740:MM=1:IF M>9 THEN MM=2
385 IF M>99 THEN MM=3
390 GOSUB 740
393 V=0:COLOR 2 :GOSUB 1100:COLOR Q*2
394 FOR A=1 TO 10:B$=INKEY$:NEXT
395 FOR J=0 TO (MM-1)
397 LOCATE 24,30-4*J:PRINT"^";
400 H$=INKEY$
405 IF H$="X"AND ZZ=1 THEN 100
406 IF H$="X" THEN CLS:PRINT B$"PERCENTA
GE:";INT(Z/(ZZ-1)*100):GOTO 120
407 IF H$="" OR H$<"0" OR H$>"9" THEN 40
0
412 FOR I= 21 TO 31:LOCATE 24,I:PRINT SP
$;:NEXT
415 P=VAL (H$):Y=20
420 V=V+(P*10^J):X=29-J*4:GOSUB 475:NEXT
 J
450 IF M=V THEN 470
452 FOR I= 20 TO 23:LOCATE I,21:FOR J=1
TO 11:PRINT SP$;:NEXT J,I
456 IF TR =1 THEN 460
458 TR =1:GOSUB 1500:GOSUB 770:GOTO 393
460 M$ =STR$(M):X =33:Y=20
462 FOR OO=MM TO 1 STEP -1
464 P = VAL (MID$ (M$,(OO+1),1))
465 X=X-4:GOSUB 475:NEXT OO:RESTORE
470 FOR I=1 TO 750:NEXT:GOSUB 1230: IF T
R=0 THEN GOSUB 2500::GOSUB 755: Z=Z+1:GO
SUB 6500
471 GOSUB 2225: GOTO 301
475 LOCATE Y,X
480 IF P=0 THEN GOSUB 720
485 ON P GOSUB 500,525,555,585,610,633,6
60,680,700:RETURN
500 PRINT R$R$;:FOR I=1 TO 4 :PRINT S$D$
L$;:NEXT :RETURN
525 PRINT S$S$S$D$L$S$D$L$TB$L$L$TB$L$L$
S$D$L$S$S$S$:RETURN
555 PRINT S$S$S$D$L$S$D$L$S$L$L$TB$D$L$L
$S$S$S$:RETURN
585 PRINT LB$R$S$D$L$L$L$S$S$S$D$L$S$D$L
$S$:RETURN
610 PRINT S$S$S$D$L$L$L$S$BB$BB$D$L$S$D$
L$L$L$S$S$S$:RETURN
633 PRINT S$S$S$D$L$L$L$S$BB$BB$D$L$L$L$
S$R$S$D$L$L$L$S$S$S$:RETURN
660 PRINT S$S$S$D$L$S$D$L$L$S$D$L$L$S$:R
ETURN
680 PRINT S$S$S$D$L$L$L$S$BB$S$D$L$L$L$S
$R$S$D$L$L$L$S$S$S$:RETURN
700 PRINT S$S$S$D$L$L$L$S$BB$S$D$L$S$D$L
$S$:RETURN
720 PRINT S$S$S$D$L$L$L$S$R$S$D$L$L$L$S$
R$S$D$L$L$L$S$S$S$:RETURN
740 LOCATE 18,21:FOR I=1 TO 11:PRINT BB$
;:NEXT:RETURN
755 LOCATE 4,7:PRINT "GOOD!":RETURN
770 LOCATE 3,8:PRINT "TRY" D$L$L$L$L$ "A
GAIN"
780 FOR I=1000 TO 500 STEP -250:SOUND I,
4:NEXT:FOR TD=1 TO 500:NEXT:RETURN
960 FOR I=1 TO 4:LOCATE X,I:PRINT S$:NEX
T:RETURN
1000 CLS:LOCATE 7,10:PRINT"DO YOU WISH T
O:"
1010 PRINT:PRINT:PRINT C$"1) PRACTICE TI
MES TABLE"
1020 PRINT:PRINT C$"2) RANDOM NUMBERS
1030 PRINT:PRINT:PRINT C$"(ENTER 1 OR 2)
";:INPUT T:IF T<1 OR T>2 THEN PRINT U$U$
U$U$:GOTO 1030
1050 IF T=2 THEN GOTO 190
1060 CLS:PRINT:PRINT:PRINT C$"ENTER TIME
S TABLE"
1070 PRINT:PRINT C$"(1-14)";:INPUT K:IF
K<1 OR K>14 THEN PRINT U$U$U$:GOTO 1070
1090 S=0:R=14:GOTO 263
1100 FOR I= 2 TO 6
1110 READ A :READ B
1120 FOR J= 1 TO B
1130 LOCATE I,J+A :PRINT CHR$(176)
1140 NEXT J:NEXT I:RESTORE:RETURN
1170 LOCATE 7,4:FOR I= 1 TO 11 :PRINT TB
$;:NEXT :RETURN
1230 COLOR 2:LOCATE 5,15:PRINT CHR$(47)U
$BB$BB$D$L$CHR$(249)LB$D$L$LB$D$L$L$L$L$
TB$TB$TB$:COLOR Q*2:RETURN
1240 LOCATE 7,5:PRINT S$ :LOCATE 7,14:PR
INT S$
1250 RETURN
1260 COLOR 2:GOSUB 1240:LOCATE 8,5:PRINT
 TB$TB$:LOCATE 8,14:PRINT TB$TB$:RETURN:
COLOR Q *2
1270 RETURN
1500 FOR I=4 TO 7:LOCATE I,15:FOR J=1 TO
 4:PRINT SP$;:NEXT J,I:RETURN
2225 FOR I= 9 TO 23:LOCATE I,21: FOR J=
1 TO 11 :PRINT SP$;:NEXT J,I:RETURN
2500 COLOR 2:LOCATE 6,17:PRINT CHR$(126)
:RETURN:COLOR Q*2
3000 COLOR Q*2:X=29:IF LEN (F$)>2 THEN 3
030
3015 P=VAL (MID$(F$,2,1))
3020 Y=9+W:GOSUB 475
3025 RETURN
3030 P=VAL(MID$(F$,3,1))
3035 Y=9+W:GOSUB 475
3040 P=VAL(MID$(F$,2,1))
3045 X=X-4:GOSUB 475
3050 RETURN
5000 DATA 6,5,5,7,4,9,3,11,3,11
6000 LOCATE 14,22:PRINT S$D$L$L$S$S$S$D$
L$L$S$;
6002 IF Q=2 THEN PRINT L$SP$U$U$L$SP$
6003 RETURN
6004 LOCATE 14,21:PRINT S$D$S$U$S$D$D$L$
L$L$S$R$S$:RETURN
6500 FOR I=500 TO 1000 STEP 250:SOUND I,
4:NEXT:RETURN
```
©

PENTOMINOS
A Puzzle-Solving Program

Jim Butterfield, Associate Editor

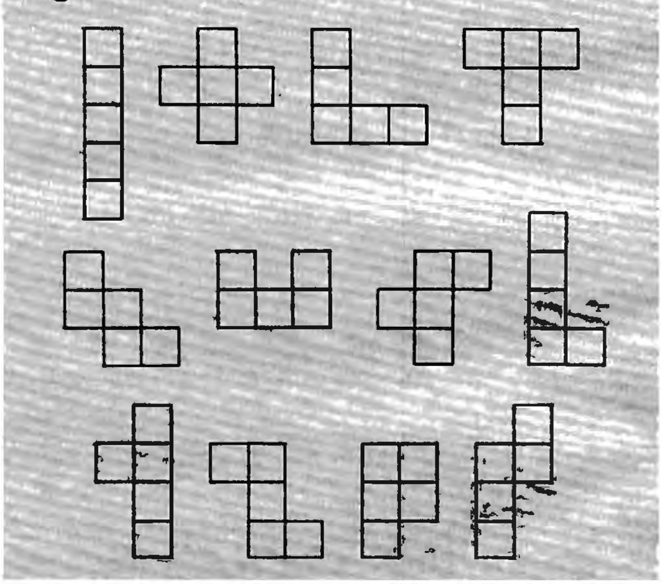
Computers can solve puzzles. With the right set of instructions, a program will follow the same logic as humans, trying things to see if they fit. It's interesting to watch the computer working in this way.

This famous puzzle is dealt with at some length in Arthur C. Clarke's novel Imperial Earth. *The characters of the novel don't use a computer to solve the puzzle.*

The original program works on all Commodore computers. Additional versions are included here for the Atari, IBM PC and PCjr, TI-99/4A, Radio Shack Color Computer, and Apple.

NOTE: *IBM, TI, Color Computer, and Apple users should insert lines 110–860 from Program 1, the Commodore version, into their programs. The* rem *statements at the ends of these lines should be ignored.*

Figure 1: The 12 Pentominos



Pentominos are like dominos, except that they are made up of five elements rather than two. If we put five squares end to end and glued them together, we'd get a long strip, often called the I pentomino. On the other hand, if we took a central square and glued the other four squares to the sides, top, and bottom, we'd get something that looks like a plus sign, which many people call the X pentomino.

Allowing for the differences that are caused by rotating or turning over a piece, there are 12 different pentominos. They are shown in Figure 1; but you might find it fun to try discovering them yourself by drawing them out on a piece of paper. Most of them look a little like letters—you can see a T, an X, and a W among them, for example.

What's The Puzzle?

The 12 different pentominos, each with an area of 5 squares, give a total of 60 squares. Suppose you had to cut these pentominos out of a rectangle without wasting any space: How big would the rectangle need to be?

We know two things: The total area is 60 squares; and the rectangle must be at least three wide (otherwise, we couldn't cut out the plus sign). So it might be possible to get all the pentominos from a rectangle that is 3 x 20, or 4 x 15, or 5 x 12, or 6 x 10. As it turns out, we can do it in any of these ways.

We can turn the question inside out and put it this way: Can you fit all 12 pentominos into a rectangle of size: 3 x 20, or 4 x 15, or 5 x 12, or 6 x 10?

The Brain Bender

Don't let the following computer program take the fun out of the puzzle for you. Cut the pieces out of cardboard and try your hand at the puzzle.

4 Color
80 COLUMN Letter Quality
PRINTER / PLOTTER

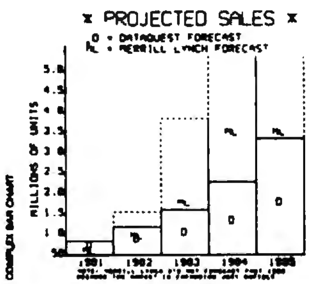
Super Special

$99 $99

1/2 PRICE SALE **1/2 PRICE SALE**

• LOWEST PRICE IN U.S.A.

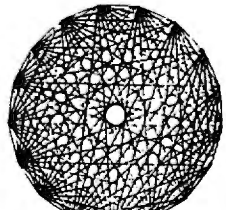
• List your programs • High resolution graphics for bar charts and geometric figures (like spirograph) • Plugs directly into VIC 20 and Commodore 64 — Interface included • Lowest cost letter quality printer in the country.



ACTUAL PRINT SAMPLES

UPPER CASE - ABCDEFGHIJKLMNOPQRSTUVWXYZ

LOWER CASE - abcdefghijklmnopqrstuvwxyz



At last you can list your programs (even control characters) and make beautiful high resolution graphics at an affordable price. This **80 column letter quality printer/plotter** is great for making complex bar charts for business plus fancy greeting cards and geometric designs. Great for homework too. **Everyone must have a 4 color printer plotter for their VIC-20 or Commodore-64.** List $199.00. **Sale $99.00.**

• 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 - MasterCard - C.O.D.

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

Olympia®

COMBINATION "DAISY WHEEL" PRINTER / TYPEWRITER

"WORLD'S FINEST"
LIST $799.00
SALE $489.00

- SUPERB COMPUTER PRINTER COMBINED WITH WORLD'S FINEST ELECTRONIC TYPEWRITER!
- BETTER THAN IBM SELECTRIC — USED BY WORLD'S LARGEST CORPORATIONS!
- TWO MACHINES IN ONE — JUST A FLICK OF THE SWITCH!
- SUPERB EXECUTIVE CORRESPONDENCE — HOME, OFFICE, WORD PROCESSING!
- EXTRA LARGE CARRIAGE — ALLOWS 14-1/8" PAPER USAGE!
- DROP IN CASSETTE RIBBON — EXPRESS LIFT OFF CORRECTION OR ERASER UP TO 46 CHARACTERS!
- PRECISION DAISY WHEEL PRINTING — MANY TYPE STYLES!
- PITCH SELECTOR — 10, 12, 15 CPS, AUTOMATIC RELOCATE KEY!
- AUTOMATIC MARGIN CONTROL AND SETTING! KEY IN BUFFER!
- ELECTRONIC RELIABILITY, BUILT IN DIAGNOSTIC TEST!
- CENTRONICS PARALLEL INTERFACE BUILT-IN (SERIAL OPTIONAL)!
- 15 DAY FREE TRIAL — 90 DAY FREE REPLACEMENT WARRANTY!

Add $17.50 for shipping and handling!!

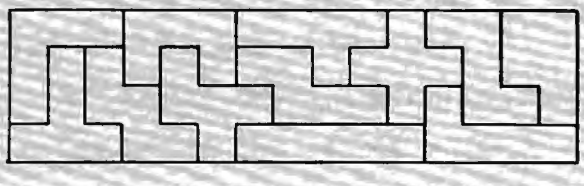
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

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

It's an interesting way to wile away the hours. 6 x 10 and 5 x 12 are not too hard; 4 x 15 will make you work; and 3 x 20, which seems at first to be the easiest, proves to be a real brain bender.

A sample solution to the 4 x 15 problem is given in Figure 2.

Figure 2: **A 4 x 15 Solution**



If humans can waste time trying to fit the pieces, computers can do it too. "Pentominos" does not run at blinding speed; it tries the pieces at about the same speed as humans do. It's dumber than human puzzle solvers: It will try to make a piece fit in places we know instinctively are hopeless. But the computer has no intuition: It will plod along, making dumb moves until it finds a combination that fits.

The program tries the pieces "visibly"—that is, you can see it putting the pieces in place, thinking about its next move, and then taking a piece back out when it becomes obvious (even to the dumb computer) that it can't work there.

In a moment we'll get to more detail on how it works. The computer always thinks about fitting the upper-leftmost empty square, and it will tell you which piece it is trying to fit there; that piece's identity will be shown in a corner of the screen. So you can track the computer's thoughts if you wish.

It can take a few minutes or several hours to find the next solution. This program is a good one to set up for an overnight run. You might want to turn off your TV set or monitor and let the computer hum away quietly all by itself.

When a solution is found, you can type CONT at any blank place on the screen, and the computer will go after the next solution.

How It Works

The pentominos and all their possible rotations are stored in DATA statements. Only four squares need to be described for each pentomino rotation, since the information gives coordinates based upon the starting square.

After reading in the data, the computer uses the following logic. Line numbers are given for those who would like to try examining the program.

1. (Line 2010) The computer looks through the list of pieces to find the first one that isn't being used. Then it searches the board for a blank square, starting at the left and searching each column top to bottom. That's the next place it will try to fit a piece. If it can't find a blank, we have a solution and will go to step 5.

2. (Line 2030) The piece just picked is set to its first rotation.

3. (Line 2060) The computer tries to fit the piece starting at the square it has identified. If it doesn't fit, it will skip ahead to step 7.

4. (Line 2120) The piece fits, so the computer puts it onto the board, onto the screen, and marks off the piece as used. It then goes back to step 1 to look for a new place to fit pieces.

5. (Line 2170) We have a solution! Stop and wait for the user to admire us. If the user types CONT, we'll keep going into step 6.

6. (Line 2190) We've reached a dead end, so we go back and remove the last piece placed on the board. If there are no pieces left, we quit; at this point we will have found all the solutions.

7. (Line 2260) Let's rotate the current piece so that we can try it in a different way. If we can find a new rotation, we go back to step 3 to try the piece. If not, we continue to step 8.

8. (Line 2300) The computer looks through the list of pieces to find the next piece to be tried. Then it goes back to step 2.

Variables And Arrays

If you're trying to read the program, it will be worthwhile to have some information on variables and arrays. Here are some useful ones:

Array B(X,Y) is the board. If the value is zero, that part of the board is blank. When a board square is used, the appropriate value in this array is set to the number of the occupying piece; but the important thing to remember is that it's set to nonzero.

The DATA statements show all rotations of all pieces. They are transferred to arrays X and Y:

Arrays X(rotation,C) and Y(rotation,C) tell where to find the squares (X and Y) of each piece's rotation. The rotation is taken from the DATA statements.

Array P(rotation) tells which piece is involved for each rotation of the above table.

Each Piece Has Data

Array P$(piece) is the name of the piece.

Array S(piece) tells where to find the starting rotation for piece X.

Array T(piece) tells which rotation is currently being used (or tried) for piece X.

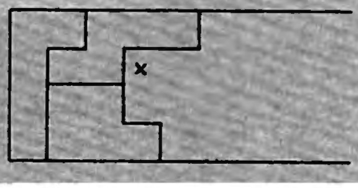
Arrays X2(piece) and X2(piece) list the starting square where piece A has been placed.

Tracking The Moves

Array U(move) lists the pieces in the order in which we tried them.

The piece under consideration is designated

Figure 3: Partial Solution. The Program Will Be Trying To Fit The Point Marked X.



by P; its current rotation, of course, will be T(P).

When we place a piece, we log it into array U and use P1 to keep track of how many pieces have been used.

Program Variations

The program could be speeded up significantly by using a compiler or by converting it to machine language. I have chosen not to do that for two reasons: compatibility and readability.

A machine language version would nevertheless be quite straightforward to write. No special math or other logic is involved. Such a program would be very fast. But it would not be universal, since different machines would need to load the program into different memory locations.

If you go for many solutions, you should realize that some of the solutions are transformations of others. Given one solution, others can be found by inverting it left to right or top to bottom. This means that each solution is really four solutions; but the computer will find each of the four as it works. If this is not desired, the extra solutions can be eliminated by removing all but two of the rotations of a single eight-rotation piece. That way, the reflected solutions couldn't happen: That piece can appear in only one orientation.

For example, we could eliminate reflected solutions by changing line 770 to DATA R,2 and then deleting lines 800 to 850 inclusive.

Making It Smarter

The program would run faster if it didn't show its moves on the screen, but watching it work is most of the fun. For one thing, it may remind you of an important aspect of computers: They're dumb, but they're faithful.

The computer will lumber along, trying dumb moves. But it won't get tired, and it will eventually reach the solution.

Yes, we could add extra logic to make the computer smarter. We could ask the computer to scan for some of the obviously impossible situations that it does not recognize at all with the present program. But there's a danger: The computer could waste more time being smart than it does being dumb.

Program 1: Pentominos For Commodore

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

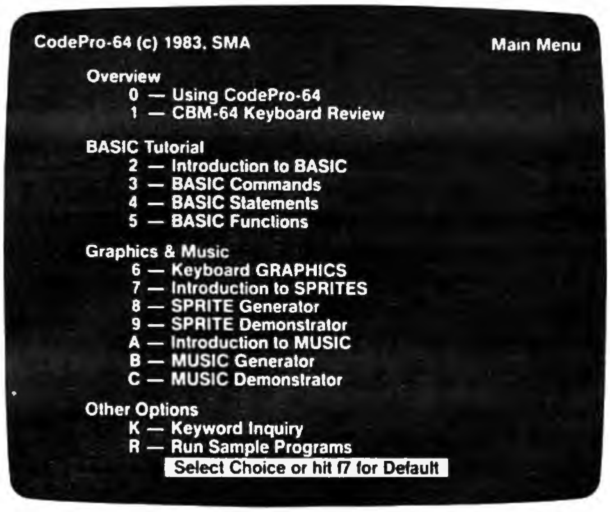
```
100 PRINT CHR$(142)"{CLR}{5 RIGHT}PENTOMI
    NOS{DOWN}"                          :rem 140
110 DATA I,2                            :rem 83
120 DATA 0,1,0,2,0,3,0,4               :rem 107
130 DATA 1,0,2,0,3,0,4,0               :rem 108
140 DATA X,1                           :rem 100
150 DATA 1,-1,1,0,2,0,1,1              :rem 152
160 DATA V,4                           :rem 103
170 DATA 0,1,0,2,1,0,2,0               :rem 108
180 DATA 0,1,0,2,1,2,2,2               :rem 113
190 DATA 1,0,2,0,2,1,2,2               :rem 114
200 DATA 1,0,2,0,2,-1,2,-2             :rem 196
210 DATA T,4                            :rem 97
220 DATA 0,1,0,2,1,1,2,1               :rem 106
230 DATA 1,0,1,1,2,0,1,2               :rem 107
240 DATA 1,0,2,0,1,-1,1,-2             :rem 198
250 DATA 2,-1,2,0,2,1,1,0              :rem 155
260 DATA W,4                           :rem 105
270 DATA 0,1,1,1,1,2,2,2               :rem 113
280 DATA 1,0,1,1,2,1,2,2               :rem 114
290 DATA 0,1,1,-1,1,0,2,-1             :rem 202
300 DATA 1,-1,1,0,2,-2,2,-1            :rem 242
310 DATA U,4                            :rem 99
320 DATA 0,2,1,0,1,1,1,2               :rem 107
330 DATA 2,0,0,1,1,1,2,1               :rem 108
340 DATA 0,1,1,0,2,0,2,1               :rem 108
350 DATA 1,0,0,1,0,2,1,2               :rem 109
360 DATA F,8                            :rem 93
370 DATA 0,1,1,-1,1,0,2,0              :rem 155
380 DATA 1,-1,2,-1,1,0,1,1             :rem 203
390 DATA 1,-1,1,0,1,1,2,1              :rem 159
400 DATA 1,-1,1,0,2,0,2,1              :rem 151
410 DATA 0,1,1,1,1,2,2,1               :rem 108
420 DATA 1,0,1,1,2,1,1,2               :rem 109
430 DATA 1,0,1,1,2,-1,2,0              :rem 154
440 DATA 1,-2,1,-1,2,-1,1,0            :rem 246
450 DATA L,8                            :rem 99
460 DATA 1,0,2,0,3,0,3,1               :rem 114
470 DATA 0,1,0,2,0,3,1,3               :rem 115
480 DATA 1,-3,1,-2,1,-1,1,0            :rem 251
490 DATA 1,0,2,0,3,0,3,-1              :rem 162
500 DATA 1,0,2,0,3,0,0,1               :rem 106
510 DATA 0,1,0,2,0,3,1,0               :rem 107
520 DATA 0,1,1,1,2,1,3,1               :rem 111
530 DATA 1,0,1,1,1,2,1,3               :rem 112
540 DATA Y,8                           :rem 112
550 DATA 0,1,0,2,0,3,1,1               :rem 112
560 DATA 1,0,2,0,3,0,1,1               :rem 113
570 DATA 1,-1,1,0,1,1,1,2              :rem 159
580 DATA 1,-1,1,0,2,0,3,0              :rem 160
590 DATA 0,1,0,2,0,3,1,2               :rem 117
600 DATA 1,0,2,0,3,0,2,1               :rem 109
610 DATA 1,-2,1,-1,1,0,1,1             :rem 199
620 DATA 1,0,2,0,3,0,2,-1              :rem 156
630 DATA Z,4                           :rem 109
640 DATA 0,1,1,1,2,1,2,2               :rem 114
650 DATA 1,0,1,1,1,2,2,2               :rem 115
660 DATA 1,-2,1,-1,1,0,2,-2            :rem 251
670 DATA 2,-1,1,0,2,0,0,1              :rem 159
680 DATA P,8                           :rem 108
690 DATA 0,1,1,0,1,1,2,0               :rem 115
700 DATA 1,0,0,1,1,1,0,2               :rem 107
710 DATA 0,1,1,0,1,1,1,2               :rem 109
720 DATA 1,0,0,1,1,1,2,1               :rem 110
730 DATA 1,-1,1,0,2,-1,2,0             :rem 202
740 DATA 1,-1,1,0,0,1,1,1              :rem 156
750 DATA 0,1,0,2,1,1,1,2               :rem 114
```

```
760 DATA 1,Ø,2,Ø,1,1,2,1                :rem 115
770 DATA R,8                            :rem 110
780 DATA Ø,1,Ø,2,1,2,1,3                :rem 119
790 DATA 1,Ø,2,Ø,2,1,3,1                :rem 120
800 DATA 1,-1,1,Ø,2,-1,3,-1             :rem 247
810 DATA 1,-1,1,Ø,Ø,1,Ø,2               :rem 154
820 DATA Ø,1,1,1,1,2,1,3                :rem 114
830 DATA 1,Ø,1,1,2,1,3,1                :rem 115
840 DATA 1,Ø,2,-1,2,Ø,3,-1              :rem 206
850 DATA 1,-2,1,-1,1,Ø,Ø,1              :rem 204
860 DATA A,Ø                             :rem 85
870 V$="{HOME}{13 DOWN}"                :rem 138
880 H$="{23 RIGHT}"                      :rem 40
1000 DIM X(63,4),Y(63,4),P(64),P$(13),S(1
     3),T(13),B(6,20)                    :rem 36
1001 DIM X1(5),Y1(5),X2(12),Y2(12),U(12)
                                        :rem 241
1010 READ P$,N:IF N=Ø GOTO 1070          :rem 81
1020 T=T+1:P$(T)=P$:S(T)=V+1            :rem 41
1030 FOR J=V+1 TO V+N:P(J)=T            :rem 12
1040 FOR K=Ø TO 3:READ X(J,K),Y(J,K):NEXT
      K,J                               :rem 203
1050 V=V+N:PRINT P$;                    :rem 158
1060 GOTO 1010                          :rem 194
1070 PRINTLEFT$(V$,5);:PRINT"CHOOSE:
     {DOWN}"                             :rem 34
1080 FOR J=3 TO 6:PRINT J;"BY";60/J;"
     {DOWN}":NEXT J                     :rem 219
1090 INPUT "SELECT 3 THRU 6";W1         :rem 205
1100 IF W1<3 OR W1>6 OR W1<>INT(W1) GOTO
     {SPACE}1070                         :rem 77
1110 W2=60/W1                           :rem 166
1120 PRINT "{CLR}"                       :rem 40
2000 REM FIND NEW SPACE TO FILL         :rem 231
2010 GOSUB 3000:P=J:GOSUB 3200:IF X1>W2 G
     OTO 2170                           :rem 178
2020 REM GET A NEW PIECE                 :rem 25
2030 T(P)=S(P)                          :rem 235
2040 PRINT "{HOME}";P$(P);"{11 DOWN}"
                                         :rem 52
2050 REM TRY FITTING PIECE               :rem 37
2060 C$=P$(P):X1(Ø)=X1:Y1(Ø)=Y1:FOR J=1 T
     O 4                                 :rem 71
2070 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)+Y1:X1
     (J)=X:Y1(J)=Y                      :rem 100
2080 IF X<1 OR Y<1 OR X>W2 OR Y>W1 GOTO 2
     260                                  :rem 8
2090 IF B(Y,X)<>Ø GOTO 2260             :rem 119
2100 NEXT J                              :rem 76
2110 REM IT FITS - PUT PIECE IN PLACE
                                          :rem 3
2120 B=P:FOR J=Ø TO 4                    :rem 67
2130 X=X1(J):Y=Y1(J):GOSUB 3500         :rem 246
2140 NEXT J                              :rem 80
2150 X2(P)=X1:Y2(P)=Y1:P1=P1+1:U(P1)=P:GO
     TO 2010                            :rem 223
2160 REM BOARD FILLED                   :rem 197
2170 PRINT "{HOME}{2 SPACES}SOLUTION";:EN
     D                                  :rem 119
2180 REM UNDRAW LAST ONE                :rem 150
2190 P=U(P1):U(P1)=Ø:P1=P1-1:IF P1<Ø THEN
      PRINT"THAT'S ALL":END             :rem 112
2200 B=Ø:X=X2(P):Y=Y2(P):C$=" ":GOSUB 350
     Ø                                   :rem 13
2210 X1=X:Y1=Y:FOR J=1 TO 4             :rem 237
2220 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)+Y1:X1
     (J)=X:Y1(J)=Y                       :rem 97
2230 GOSUB 3500                          :rem 15
2240 NEXT J                              :rem 81
2250 REM ROTATE THE PIECE               :rem 195
2260 T(P)=T(P)+1:IF P(T(P))=P GOTO 2060
                                         :rem 58
2270 REM GIVE UP ON PIECE               :rem 130
2280 T(P)=Ø                              :rem 46
2290 REM LOOK FOR NEW PIECE              :rem 29
2300 P=P+1:IF P>12 GOTO 2190            :rem 189
2310 IF T(P)<>Ø GOTO 2300               :rem 242
2320 GOTO 2030                          :rem 197
3000 FOR J=1 TO 12:IF T(J)<>Ø THEN NEXT J
                                        :rem 130
3010 RETURN                             :rem 164
3200 FOR X1=1 TO W2:FOR Y1=1 TO W1:rem 19
3210 IF B(Y1,X1)=Ø GOTO 3230            :rem 149
3220 NEXT Y1,X1                          :rem 69
3230 RETURN                             :rem 168
3500 PRINT LEFT$(V$,Y+2);LEFT$(H$,X);C$:B
     (Y,X)=B                            :rem 231
3510 RETURN                             :rem 169
```

Program 2: Pentominos For Atari

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

```
FE 100 PRINT "(CLEAR)PLEASE WAIT...   I
       NITIALIZING ARRAYS":POKE 752,1:
       POSITION Ø,Ø
FD 110 DATA I,2
GL 120 DATA Ø,1,Ø,2,Ø,3,Ø,4
GM 130 DATA 1,Ø,2,Ø,3,Ø,4,Ø
GE 140 DATA X,1
JI 150 DATA 1,-1,1,Ø,2,Ø,1,1
GH 160 DATA V,4
GM 170 DATA Ø,1,Ø,2,1,Ø,2,Ø
HB 180 DATA Ø,1,Ø,2,1,2,2,2
HC 190 DATA 1,Ø,2,Ø,2,1,2,2
ME 200 DATA 1,Ø,2,Ø,2,-1,2,-2
GB 210 DATA T,4
GK 220 DATA Ø,1,Ø,2,1,1,2,1
GL 230 DATA 1,Ø,1,1,2,Ø,1,2
MG 240 DATA 1,Ø,2,Ø,1,-1,1,-2
JL 250 DATA 2,-1,2,Ø,2,1,1,Ø
GJ 260 DATA W,4
HB 270 DATA Ø,1,1,1,1,2,2,2
HC 280 DATA 1,Ø,1,1,2,1,2,2
MK 290 DATA Ø,1,1,-1,1,Ø,2,-1
PC 300 DATA 1,-1,1,Ø,2,-2,2,-1
GD 310 DATA U,4
GL 320 DATA Ø,2,1,Ø,1,1,1,2
GM 330 DATA 2,Ø,Ø,1,1,1,2,1
GM 340 DATA Ø,1,1,Ø,2,Ø,2,1
GN 350 DATA 1,Ø,Ø,1,Ø,2,1,2
FN 360 DATA F,8
JL 370 DATA Ø,1,1,-1,1,Ø,2,Ø
ML 380 DATA 1,-1,2,-1,1,Ø,1,1
JP 390 DATA 1,-1,1,Ø,1,1,2,1
JH 400 DATA 1,-1,1,Ø,2,Ø,2,1
GM 410 DATA Ø,1,1,1,1,2,2,1
GN 420 DATA 1,Ø,1,1,2,1,1,2
JK 430 DATA 1,Ø,1,1,2,-1,2,Ø
PG 440 DATA 1,-2,1,-1,2,-1,1,Ø
GD 450 DATA L,8
HC 460 DATA 1,Ø,2,Ø,3,Ø,3,1
HD 470 DATA Ø,1,Ø,2,Ø,3,1,3
PL 480 DATA 1,-3,1,-2,1,-1,1,Ø
KC 490 DATA 1,Ø,2,Ø,3,Ø,3,-1
GK 500 DATA 1,Ø,2,Ø,3,Ø,Ø,1
GL 510 DATA Ø,1,Ø,2,Ø,3,1,Ø
GP 520 DATA Ø,1,1,1,2,1,3,1
HA 530 DATA 1,Ø,1,1,1,2,1,3
HA 540 DATA Y,8
HA 550 DATA Ø,1,Ø,2,Ø,3,1,1
HB 560 DATA 1,Ø,2,Ø,3,Ø,1,1
JP 570 DATA 1,-1,1,Ø,1,1,1,2
KA 580 DATA 1,-1,1,Ø,2,Ø,3,Ø
HF 590 DATA Ø,1,Ø,2,Ø,3,1,2
```



A new concept in interactive visual learning.

CodePro-64™

Now you can learn to code in BASIC and develop advanced programming skills with graphics, sprites and music—*visually*. You learn by interacting with CodePro-64, a new concept in interactive visual learning.

SEE PROGRAM EXECUTION

Imagine actually seeing BASIC statements execute. CodePro-64 guides you through structured examples of BASIC program segments. You enter the requested data or let CodePro-64 do the typing for you. (It will not let you make a mistake.)

You step through and actually see the execution of sample program statements by simply pressing the space bar. CodePro-64 does the rest. You see statements with corresponding *graphics* and variable value displays.

EXTENSIVE TUTORIAL

CodePro-64's extensive tutorial guides you through each BASIC command, program statement, and function. You get clear explanations. Where appropriate, you invoke BasicView to see examples execute and watch their flow charts and variables change.

By seeing graphic displays of program segment execution you learn by visual example. You *learn faster and grasp programming concepts easier* with CodePro-64 because you immediately see the results of your input.

You control your learning. You can go through the tutorial sequentially, or return to the main menu and select different topics, or *use keywords* to select language elements to study. You can page back and forth between screens within a topic at the touch of a function key.

Once you have practiced and mastered the BASIC language elements you move on to more advanced concepts. You learn about sprite and music programming.

SPRITE GENERATOR & DEMONSTRATOR

CodePro-64's sprite generator lets you *define your own sprites* on the screen. You learn how to define sprites and what data values correspond to your sprite definitions. (You can then save your sprite data to a diskette file for use in your own programs.) You can *easily experiment* with different definitions and make changes to immediately see the effects.

We also help you learn to program with sprites by giving you a *sprite demonstrator* so you can see the effect of changing register values. You can experiment by moving your sprite around in a screen segment, change its color and see the effects of your changes. You learn by visual examples.

MUSIC GENERATOR & DEMONSTRATOR

Our *Music Generator* and *Music Demonstrator* will provide hours of instruction and creative enjoyment. From the beginning of your instruction you can *compose simple tunes* on the screen using the generator. Once you've completed a composition you can *save the tune* and its associated SID parameters to a diskette file. Our music sample program can be used alone or incorporated into your own programs to *read the saved music file* and replay your songs.

Our music demonstrator *lets you experiment* with various combinations of music programming parameters and hear the results. All you do is enter rows of SID parameters on the screen to create a particular sound. Then you *hear each sound by playing the "keyboard organ"* in real time as you shift from row to row of SID parameters. By seeing your input and hearing the result you quickly learn how to create new musical sounds and special sound effects.

Whether you're a beginning programmer or an experienced professional, CodePro-64 will help you improve you Commodore 64 programming skills. We're sure because CodePro-64 was developed by a team of two professionals with *over 25 years* of software development experience.

CodePro-64 is a professional quality educational program for the serious student of personal computing. And it's *fully guaranteed*. Order yours today.

OUR GUARANTEE

We guarantee your satisfaction. You must be satisfied with CodePro-64 for the Commodore-64. Try it for 10 days and if for any reason you are not satisfied return it to us (undamaged) for a full refund. *No risk.*

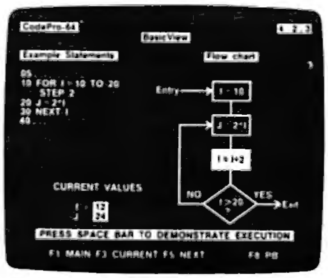
HOW TO ORDER

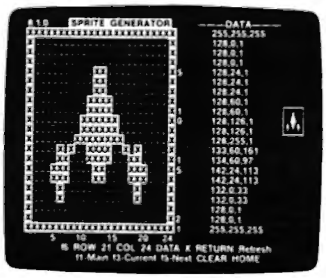
Order your copy of CodePro-64 today by mail or phone. Send only $59.95 plus $3.00 shipping and handling to:

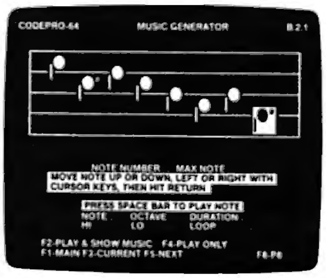
SYSTEMS MANAGEMENT ASSOCIATES
3700 Computer Drive, Dept. G-1
Raleigh, N.C. 27609

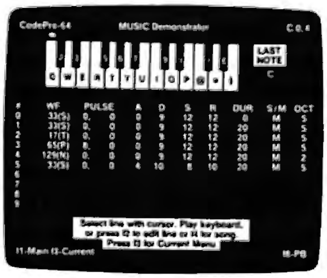
Available on diskette only. MasterCard/VISA accepted. For faster service on credit card orders, call toll free 1-800 SMA-RUSH. (1-800-762-7874). *Dealer inquiries invited.*

Commodore 64 is a trademark of Commodore Business Machines, Inc.









SYSTEMS MANAGEMENT ASSOCIATES

```
GN 600  DATA 1,Ø,2,Ø,3,Ø,2,1
MH 610  DATA 1,-2,1,-1,1,Ø,1,1
JM 620  DATA 1,Ø,2,Ø,3,Ø,2,-1
GN 630  DATA Z,4
HC 640  DATA Ø,1,1,1,2,1,2,2
HD 650  DATA 1,Ø,1,1,1,2,2,2
FL 660  DATA 1,-2,1,-1,1,Ø,2,-2
JP 670  DATA 2,-1,1,Ø,2,Ø,Ø,1
GM 680  DATA F,8
HD 690  DATA Ø,1,1,Ø,1,1,2,Ø
GL 700  DATA 1,Ø,Ø,1,1,1,Ø,2
GN 710  DATA Ø,1,1,Ø,1,1,1,2
GO 720  DATA 1,Ø,Ø,1,1,1,2,1
MK 730  DATA 1,-1,1,Ø,2,-1,2,Ø
JM 740  DATA 1,-1,1,Ø,Ø,1,1,1
HC 750  DATA Ø,1,Ø,2,1,1,1,2
HD 760  DATA 1,Ø,2,Ø,1,1,2,1
GO 770  DATA R,8
HH 780  DATA Ø,1,Ø,2,1,2,1,3
HI 790  DATA 1,Ø,2,Ø,2,1,3,1
PH 800  DATA 1,-1,1,Ø,2,-1,3,-1
JH 810  DATA 1,-1,1,Ø,Ø,1,Ø,2
HC 820  DATA Ø,1,1,1,1,2,1,3
HD 830  DATA 1,Ø,1,1,2,1,3,1
MO 840  DATA 1,Ø,2,-1,2,Ø,3,-1
MM 850  DATA 1,-2,1,-1,1,Ø,Ø,1
FF 860  DATA A,Ø
HE 1000 DIM X(63,4),Y(63,4),P(64),PP$(
        13),S(13),T(13),B(6,20)
CI 1001 DIM X1(5),Y1(5),X2(12),Y2(12),
        U(12),C$(1),P$(1)
MP 1002 Z=Ø:FOR I=Ø TO 63:P(I)=Z:FOR J
        =Ø TO 4:X(I,J)=Z:Y(I,J)=Z:NEXT
         J:NEXT I
OA 1003 P(64)=Z:FOR I=Ø TO 12:S(I)=Z:T
        (I)=Z:X2(I)=Z:Y2(I)=Z:U(I)=Z:N
        EXT I:S(13)=Z:T(13)=Z
BK 1004 FOR I=Ø TO 6:FOR J=Ø TO 2Ø:B(I
        ,J)=Z:NEXT J:NEXT I:FOR I=Ø TO
         5:X1(I)=Z:Y1(I)=Z:NEXT I
GD 1005 PRINT "{CLEAR}":POSITION 15,Ø:
        PRINT "PENTOMINOS":PRINT
EH 1010 READ P$,N:IF N=Ø THEN 1070
PJ 1020 T=T+1:PP$(T,T)=P$:S(T)=V+1
AM 1030 FOR J=V+1 TO V+N:P(J)=T
DI 1040 FOR K=Ø TO 3:READ L,M:X(J,K)=L
        :Y(J,K)=M:NEXT K:NEXT J
JD 1050 V=V+N:PRINT P$;
MC 1060 GOTO 1010
JM 1070 POSITION 1,5:PRINT "CHOOSE:":P
        RINT
EL 1080 FOR J=3 TO 6:PRINT J;" BY ";60
        /J:NEXT J
JF 1090 PRINT :PRINT "SELECT 3 THRU 6:
         ";:INPUT W1
HM 1100 IF W1<3 OR W1>6 OR W1<>INT(W1)
         THEN GOTO 1070
KG 1110 W2=60/W1
BC 1120 PRINT "{CLEAR}"
OH 2000 REM FIND NEW SPACE TO FILL
OB 2010 GOSUB 3000:P=J:GOSUB 3200:IF X
        1>W2 THEN GOTO 2170
BJ 2020 REM GET A NEW PIECE
OL 2030 T(P)=S(P)
NP 2040 POSITION 1,1:PRINT PP$(P,P):PO
        SITION Ø,12
CF 2050 REM TRY FITTING PIECE
BD 2060 C$=PP$(P,P):X1(Ø)=X1:Y1(Ø)=Y1:
        FOR J=1 TO 4
GE 2070 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)
        +Y1:X1(J)=X:Y1(J)=Y
DH 2080 IF X<1 OR Y<1 OR X>W2 OR Y>W1
        THEN GOTO 2260
KG 2090 IF B(Y,X)<>Ø THEN GOTO 2260
EM 2100 NEXT J
AD 2110 REM IT FITS - PUT PIECE IN PLA
        CE
ED 2120 B=P:FOR J=Ø TO 4
PG 2130 X=X1(J):Y=Y1(J):GOSUB 3500
FA 2140 NEXT J
NP 2150 X2(P)=X1:Y2(P)=Y1:P1=P1+1:U(P1
        )=P:GOTO 2010
MF 2160 REM BOARD FILLED
DF 2170 POSITION Ø,12:PRINT "SOLUTION"
        ;:POKE 752,Ø:END
JG 2180 REM UNDRAW LAST ONE
HA 2190 P=U(P1):U(P1)=Ø:P1=P1-1:IF P1<
        Ø THEN PRINT "THAT'S ALL":END
AN 2200 B=Ø:X=X2(P):Y=Y2(P):C$=" ":GOS
        UB 3500
ON 2210 X1=X:Y1=Y:FOR J=1 TO 4
SB 2220 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)
        +Y1:X1(J)=X:Y1(J)=Y
AP 2230 GOSUB 3500
FB 2240 NEXT J
MD 2250 REM ROTATE THE PIECE
GJ 2260 T(P)=T(P)+1:IF P(T(P))=P THEN
        GOTO 2060
IC 2270 REM GIVE UP ON PIECE
CO 2280 T(P)=Ø
BN 2290 REM LOOK FOR NEW PIECE
OM 2300 P=P+1:IF P>12 THEN GOTO 2190
OI 2310 IF T(P)<>Ø THEN 2300
MF 2320 GOTO 2030
IC 3000 FOR J=1 TO 12:IF T(J)<>Ø THEN
        NEXT J
KE 3010 RETURN
BD 3200 FOR X1=1 TO W2:FOR Y1=1 TO W1
IL 3210 IF B(Y1,X1)=Ø THEN 3230
JC 3220 NEXT Y1:NEXT X1
HI 3230 RETURN
MO 3500 POSITION X,Y+2:PRINT C$:B(Y,X)
        =B
KJ 3510 RETURN
```

Program 3: Pentominos For IBM PC/PCjr

Insert lines 110–860 from the Commodore version (Program 1).

```
100 CLS:PRINT "                PENTOMINOS":P
RINT
1000 DIM X(63,4),Y(63,4),P(64),P$(13),S(
13),T(13),B(6,20)
1001 DIM X1(5),Y1(5),X2(12),Y2(12),U(12)

1010 READ P$,N:IF N=0 GOTO 1070
1020 T=T+1:P$(T)=P$:S(T)=V+1
1030 FOR J=V+1 TO V+N:P(J)=T
1040 FOR K=0 TO 3:READ X(J,K),Y(J,K):NEX
T K,J
1050 V=V+N:PRINT P$;
1060 GOTO 1010
1070 LOCATE 5,1:PRINT"CHOOSE:":PRINT
1080 FOR J=3 TO 6:PRINT J;"BY";60/J;"":P
RINT:NEXT J
1090 INPUT "SELECT 3 THRU 6";W1
1100 IF W1<3 OR W1>6 OR W1<>INT(W1) GOTO
 1070
1110 W2=60/W1
1120 CLS
2000 REM FIND NEW SPACE TO FILL
2010 GOSUB 3000:P=J:GOSUB 3200:IF X1>W2
GOTO 2170
2020 REM GET A NEW PIECE
2030 T(P)=S(P)
2040 LOCATE 1,1:PRINT P$(P)
2050 REM TRY FITTING PIECE
2060 C$=P$(P):X1(0)=X1:Y1(0)=Y1:FOR J=1
TO 4
```

```
2070 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)+Y1:X
1(J)=X:Y1(J)=Y
2080 IF X<1 OR Y<1 OR X>W2 OR Y>W1 GOTO
2260
2090 IF B(Y,X)<>0 GOTO 2260
2100 NEXT J
2110 REM IT FITS - PUT PIECE IN PLACE
2120 B=P:FOR J=0 TO 4
2130 X=X1(J):Y=Y1(J):GOSUB 3500
2140 NEXT J
2150 X2(P)=X1:Y2(P)=Y1:P1=P1+1:U(P1)=P:G
OTO 2010
2160 REM BOARD FILLED
2170 LOCATE 15,1:PRINT "  SOLUTION";:END
2180 REM UNDRAW LAST ONE
2190 P=U(P1):U(P1)=0:P1=P1-1:IF P1<0 THE
N PRINT"THAT'S ALL":END
2200 B=0:X=X2(P):Y=Y2(P):C$=" ":GOSUB 35
00
2210 X1=X:Y1=Y:FOR J=1 TO 4
2220 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)+Y1:X
1(J)=X:Y1(J)=Y
2230 GOSUB 3500
2240 NEXT J
2250 REM ROTATE THE PIECE
2260 T(P)=T(P)+1:IF P(T(P))=P GOTO 2060
2270 REM GIVE UP ON PIECE
2280 T(P)=0
2290 REM LOOK FOR NEW PIECE
2300 P=P+1:IF P>12 GOTO 2190
2310 IF T(P)<>0 GOTO 2300
2320 GOTO 2030
3000 FOR J=1 TO 12:IF T(J)<>0 THEN NEXT
J
3010 RETURN
3200 FOR X1=1 TO W2:FOR Y1=1 TO W1
3210 IF B(Y1,X1)=0 GOTO 3230
3220 NEXT Y1,X1
3230 RETURN
3500 LOCATE Y+2,X:PRINT C$:B(Y,X)=B
3510 RETURN
```

Program 4: Pentominos For TI-99/4A

Insert lines 110–860 from the Commodore version (Program 1).

(**Note:** *If using a disk drive, type CALL FILES(1) before loading and running this program.*)

```
40 CALL CLEAR
50 PRINT "{8 SPACES}PENTOMINOS": :
60 GOTO 870
70 FOR I=1 TO LEN(A$)
80 CALL HCHAR(ROW,COL+I,ASC(SEG$(A$
   ,I,1)))
90 NEXT I
100 RETURN
870 DIM XX(63,4),YY(63,4),PP(64),PP
   $(13),SS(13),TT(13),BB(6,20)
880 DIM XX1(5),YY1(5),XX2(12),YY2(1
   2),UU(12)
890 CT=5
900 READ P$,N
910 IF N=0 THEN 1040
920 T=T+1
930 PP$(T)=P$
940 SS(T)=V+1
950 FOR J=V+1 TO V+N
960 PP(J)=T
970 FOR K=0 TO 3
980 READ XX(J,K),YY(J,K)
990 NEXT K
1000 NEXT J
1010 V=V+N
1020 PRINT P$;
1030 GOTO 900
1040 CALL CLEAR
1050 PRINT " CHOOSE:": :
1060 FOR J=3 TO 6
1070 PRINT J;" BY ";60/J
1080 NEXT J
1090 PRINT
1100 INPUT " SELECT 3 THRU 6: ":W1
1110 IF (W1<3)+(W1>6)+(W1<>INT(W1))
     THEN 1040
1120 W2=60/W1
1130 CALL CLEAR
1140 REM FIND NEW SPACE TO FILL
1150 GOSUB 1930
1160 P=J
1170 GOSUB 1970
1180 IF X1>W2 THEN 1500
1190 REM GET A NEW PIECE
1200 TT(P)=SS(P)
1210 ROW=CT
1220 COL=5+CT
1230 A$=PP$(P)
1240 GOSUB 70
1250 REM TRY FITTING PIECE
1260 C$=PP$(P)
1270 XX1(0)=X1
1280 YY1(0)=Y1
1290 FOR J=1 TO 4
1300 X=XX(TT(P),J-1)+X1
1310 Y=YY(TT(P),J-1)+Y1
1320 XX1(J)=X
1330 YY1(J)=Y
1340 IF (X<1)+(Y<1)+(X>W2)+(Y>W1)TH
     EN 1840
1350 IF BB(Y,X)<>0 THEN 1840
1360 NEXT J
1370 REM IT FITS - PUT PIECE IN PLA
     CE
1380 B=P
1390 FOR J=0 TO 4
1400 X=XX1(J)
1410 Y=YY1(J)
1420 GOSUB 2030
1430 NEXT J
1440 XX2(P)=X1
1450 YY2(P)=Y1
1460 P1=P1+1
1470 UU(P1)=P
1480 GOTO 1150
1490 REM BOARD FILLED
1500 ROW=15
1510 COL=5+CT
1520 A$="SOLUTION"
1530 GOSUB 70
1540 ROW=17
1550 COL=5
1560 A$="FIND ANOTHER SOLUTION?"
1570 GOSUB 70
1580 CALL KEY(3,K,S)
1590 IF S<>1 THEN 1580
1600 IF CHR$(K)="Y" THEN 1620
1610 END
1620 REM UNDRAW LAST ONE
1630 P=UU(P1)
1640 UU(P1)=0
1650 P1=P1-1
1660 IF P1>=0 THEN 1690
1670 PRINT "THAT'S ALL"
1680 STOP
1690 B=0
1700 X=XX2(P)
```

```
1710 Y=YY2(P)
1720 C$=" "
1730 GOSUB 2030
1740 X1=X
1750 Y1=Y
1760 FOR J=1 TO 4
1770 X=XX(TT(P),J-1)+X1
1780 Y=YY(TT(P),J-1)+Y1
1790 XX1(J)=X
1800 YY1(J)=Y
1810 GOSUB 2030
1820 NEXT J
1830 REM ROTATE THE PIECE
1840 TT(P)=TT(P)+1
1850 IF PP(TT(P))=P THEN 1260
1860 REM GIVE UP ON PIECE
1870 TT(P)=0
1880 REM LOOK FOR NEW PIECE
1890 P=P+1
1900 IF P>12 THEN 1630
1910 IF TT(P)<>0 THEN 1890
1920 GOTO 1200
1930 FOR J=1 TO 12
1940 IF TT(J)=0 THEN 1960
1950 NEXT J
1960 RETURN
1970 FOR X1=1 TO W2
1980 FOR Y1=1 TO W1
1990 IF BB(Y1,X1)=0 THEN 2020
2000 NEXT Y1
2010 NEXT X1
2020 RETURN
2030 ROW=Y+1+CT
2040 COL=X+CT
2050 A$=C$
2060 GOSUB 70
2070 BB(Y,X)=B
2080 RETURN
```

Program 5:
Pentominos For The Color Computer

Insert lines 110–860 from the Commodore version (Program 1).

```
100 CLS:PRINT"{11 SPACES}PENTOMINOS"
999 PCLEAR 1
1000 DIM X(63,4),Y(63,4),P(64),P$(1
     3),S(13),T(13),B(6,20)
1001 DIM X1(5),Y1(5),X2(12),Y2(12),
     U(12)
1010 READ P$,N:IF N=0 GOTO 1070
1020 T=T+1:P$(T)=P$:S(T)=V+1
1030 FOR J=V+1 TO V+N: P(J)=T
1040 FOR K=0 TO 3:READ X(J,K),Y(J,K
     ):NEXT K,J
1050 V=V+N:PRINT P$;
1060 GOTO 1010
1070 PRINT@64,"CHOOSE:"
1080 FOR J=3 TO 6:PRINT J;" BY";60/
     J:NEXT J
1090 INPUT "SELECT 3 THRU 6";W1
1100 IF W1<3 OR W1>6 OR W1<>INT(W1)
     GOTO 1070
1110 W2=60/W1
1120 CLS
2000 REM FIND NEW SPACE TO FILL
2010 GOSUB 3000:P=J:GOSUB 3200:IF X
     1>W2 GOTO 2170
2020 REM GET A NEW PIECE
2030 T(P)=S(P)
2040 PRINT@33,P$(P)
2050 REM TRY FITTING PIECE
2060 C$=P$(P):X1(0)=X1:Y1(0)=Y1:FOR
     J=1 TO 4
2070 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)
     +Y1:X1(J)=X:Y1(J)=Y
2080 IF X<1 OR Y<1 OR X>W2 OR Y>W1
     GOTO 2260
2090 IF B(Y,X)<>0 GOTO 2260
2100 NEXT J
2110 REM IT FITS - PUT PIECE IN PLA
     CE
2120 B=P:FOR J=0 TO 4
2130 X=X1(J):Y=Y1(J):GOSUB 3500
2140 NEXT J
2150 X2(P)=X1:Y2(P)=Y1:P1=P1+1:U(P1
     )=P:GOTO 2010
2160 REM BOARD FILLED
2170 PRINT@385,"SOLUTION":END
2180 REM UNDRAW LAST ONE
2190 P=U(P1):U(P1)=0:P1=P1-1:IF P1<
     0 THEN PRINT"THAT'S ALL":END
2200 B=0:X=X2(P):Y=Y2(P):C$=" ":GOS
     UB 3500
2210 X1=X:Y1=Y:FOR J=1 TO 4
2220 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)
     +Y1:X1(J)=X:Y1(J)=Y
2230 GOSUB 3500
2240 NEXT J
2250 REM ROTATE THE PIECE
2260 T(P)=T(P)+1:IF P(T(P))=P GOTO
     2060
2270 REM GIVE UP ON PIECE
2280 T(P)=0
2290 REM LOOK FOR NEW PIECE
2300 P=P+1:IF P>12 GOTO 2190
2310 IF T(P)<>0 GOTO 2300
2320 GOTO 2030
3000 FOR J=1 TO 12:IF T(J)<>0 THEN
     NEXT J
3010 RETURN
3200 FOR X1=1 TO W2:FOR Y1=1 TO W1
3210 IF B(Y1,X1)=0 GOTO 3230
3220 NEXT Y1,X1
3230 RETURN
3500 PRINT @X+(Y+2)*32,C$;:B(Y,X)=B
3510 RETURN
```

Program 6: Pentominos For The Apple

Insert lines 110–860 from the Commodore version (Program 1).

```
1000  DIM X(63,4),Y(63,4),P(64),P$(13),
     S(13),T(13),B(6,20)
1001  DIM X1(5),Y1(5),X2(12),Y2(12),U(1
     2)
1003  HOME : HTAB 16: PRINT "PENTOMINOS
     ": PRINT
1010  READ P$,N: IF N = 0 GOTO 1070
1020 T = T + 1:P$(T) = P$:S(T) = V + 1
1030  FOR J = V + 1 TO V + N:P(J) = T
1040  FOR K = 0 TO 3: READ X(J,K),Y(J,K
     ): NEXT K,J
1050 V = V + N: PRINT P$;
1060  GOTO 1010
1070  PRINT : VTAB (5): PRINT "CHOOSE:"
     : PRINT
1080  FOR J = 3 TO 6: PRINT J;" BY ";60
      / J: PRINT : NEXT J
1090  INPUT "SELECT 3 THRU 6? ";W1
1100  IF W1 < 3 OR W1 > 6 OR W1 <  >  INT
     (W1) GOTO 1070
1110 W2 = 60 / W1
1120  HOME
2000  REM  FIND NEW SPACE TO FILL
2010  GOSUB 3000:P = J: GOSUB 3200: IF
     X1 > W2 GOTO 2170
```

```
2020  REM  GET A NEW PIECE
2030 T(P) = S(P)
2040  VTAB 1: PRINT P$(P): VTAB 12
2050  REM  TRY FITTING PIECE
2060 C$ = P$(P):X1(0) = X1:Y1(0) = Y1: FOR
     J = 1 TO 4
2070 X = X(T(P),J - 1) + X1:Y = Y(T(P),
     J - 1) + Y1:X1(J) = X:Y1(J) = Y
2080  IF X < 1 OR Y < 1 OR X > W2 OR Y >
     W1 GOTO 2260
2090  IF B(Y,X) <  > 0 GOTO 2260
2100  NEXT J
2110  REM  IT FITS - PUT PIECE IN PLACE

2120 B = P: FOR J = 0 TO 4
2130 X = X1(J):Y = Y1(J): GOSUB 3500
2140  NEXT J
2150 X2(P) = X1:Y2(P) = Y1:P1 = P1 + 1:
     U(P1) = P: GOTO 2010
2160  REM  BOARD FILLED
2170  VTAB 1: PRINT "  SOLUTION";: END

2180  REM  UNDRAW LAST ONE
2190 P = U(P1):U(P1) = 0:P1 = P1 - 1: IF
     P1 < 0 THEN  PRINT "THAT'S ALL": END

2200 B = 0:X = X2(P):Y = Y2(P):C$ = " "
     : GOSUB 3500
2210 X1 = X:Y1 = Y: FOR J = 1 TO 4
2220 X = X(T(P),J - 1) + X1:Y = Y(T(P),
     J - 1) + Y1:X1(J) = X:Y1(J) = Y
2230  GOSUB 3500
2240  NEXT J
2250  REM  ROTATE THE PIECE
2260 T(P) = T(P) + 1: IF P(T(P)) = P GOTO
     2060
2270  REM  GIVE UP ON PIECE
2280 T(P) = 0
2290  REM  LOOK FOR NEW PIECE
2300 P = P + 1: IF P > 12 GOTO 2190
2310  IF T(P) <  > 0 GOTO 2300
2320  GOTO 2030
3000  FOR J = 1 TO 12: IF T(J) <  > 0 THEN
      NEXT J
3010  RETURN
3200  FOR X1 = 1 TO W2: FOR Y1 = 1 TO W
     1
3210  IF B(Y1,X1) = 0 GOTO 3230
3220  NEXT Y1,X1
3230  RETURN
3500  VTAB Y + 4: HTAB X: PRINT C$:B(Y,
     X) = B
3510  RETURN
```

Pitstop

Shay Addams

Racing games are nothing new, but *Pitstop* from Epyx incorporates a realistic element of the sport that sets it apart from everything else on the track. In addition to zooming around the course as fast as possible, you must develop a solid plan for maneuvering your three-man pit crew when you're forced to pull in for fresh tires and refueling. The game is available on cartridge for Atari, Commodore 64, and Coleco Adam computers.

The action takes place on one of six speedways, all based on genuine tracks such as Le Mans and Monaco. You can race at any one, or opt for the "Mini-Circuit," in which the program picks three courses at random for you to complete, one after the other. Hardcore speed demons will prefer the "Grand Circuit"—it requires you to cover all six tracks in succession, a grueling marathon event. The number of laps per race can be set to three, six, or nine; skill levels include Rookie, Semi-Pro, and Pro. Up to four players can compete by taking turns.

The race kicks off as you push forward on the stick to accelerate. The perspective and graphics are similar to *Enduro*, but unfortunately not as detailed as *Pole Position*. While you accelerate, the gears shift automatically, accompanied by authentic sound effects. The screen scrolls vertically, with a green background and yellow cars. Your speed, elapsed time, and current lap are constantly displayed.

No more than two other cars are on the track simultaneously, but they are programmed to swerve into your path or travel side by side to prevent your passing them. The main thing to watch out for is bumping into other cars or the sides of the road. An accident won't cause a colorful explosion the way it does in *Pole Position*, but it will reduce your speed as in *Baja Buggies*.

Trouble With Tires

This is where *Pitstop* takes a detour from the familiar "race around the track" scenario of similar games. When you smash into another car or the railing alongside the road, the corresponding tire is damaged. Starting off a deep blue, the tires change to a different hue each time you have an accident. Sustain too much damage and the tire explodes, knocking you out of the race. You've got to keep an eye on the color of all four tires and be ready to pull into the pits when they turn a bright red (indicating that they'll burst on the next collision).

The pit area is located to the right of the finish line. An inset map on the left displays an overhead view of the course, with your car's current position and the finish line prominently marked. Turn into the off-ramp on the right as you pass the finish line, and the scene cuts to a three-quarter perspective of your car sitting in the pits. Now your vehicle is revealed as one of those low-slung, Indy 500-type racers, and is larger and much more detailed.

Action In The Pits

A member of your pit crew waits on each side of the car, standing by to change the tires. Another is behind you, gas hose in hand. If the horizontal fuel gauge says you're running low, it's best to get the gas pumping immediately. This is done by using the joystick to move a cursor over the man, then hitting the fire button. Now you can steer him into place, where he automatically starts refilling your tank.

Tires are changed by activating one of the other men and moving him to the tire you want removed. He'll latch onto it, and you can guide him to a stack of fresh tires. When he touches the stack, the tire he's holding turns a deep blue to indicate that he's got a new tire, which he can then attach to the car. But keep your eyes on the gas gauge, because if you don't remove the nozzle when the tank's topped off, the gas spills over and you have to fill it up again.

While all this is going on, a timer at the top right of the screen shows the seconds ticking away to remind you how much time you're losing in the pits. Another digital display at bottom left tells you how much overall time has elapsed since the race began. To underscore the urgency of getting out of the pits as quickly as possible, the rest of the cars keep racing past in the background, their engines buzzing as they gain distance on you. When you're ready to roll, position the cursor over the man in front of the car and he'll raise his flag to wave you back onto the track.

Multiplayer Competition

You can make it through three laps around most tracks without a stop for gas or tire changes, but the only fun involved in this is trying to beat your best time for the same course. *Pitstop*'s more enjoyable in group play. When one driver completes the set number of laps, the next one takes a whirl around the track. After the race, each player's time is posted, along with his portion of the $94,000 prize money. If you're competing in a Mini- or Grand circuit, the overall winnings are displayed at the bottom. If two or more players tie, the one who started first wins, so flip a coin to determine who goes first.

In addition to the exciting competition and action, *Pitstop* requires strategy and split-second decision-making that are missing in other racing games. Should you try to finish the race in spite of a severely damaged tire, or pull into the pits and at least insure that you complete the race? Is there time to change all four tires? Situations like these put a real edge on the game play. Since veteran race car drivers agree that many professional races are won in the pits, not on the track, *Pitstop* has to be one of the most realistic and playable racing simulations available.

Pitstop
Epyx Computer Software
1043 Kiel Court
Sunnyvale, CA 94089
Atari, 64 versions, $39.95
Coleco Adam version, $53

©

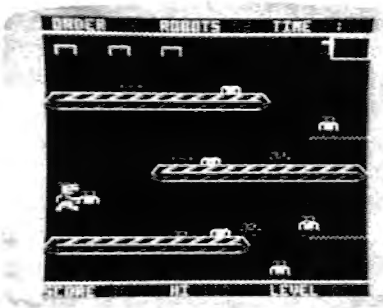
Panic Button For VIC And TRS-80 Color Computer

Michael B. Williams

Not wanting to imitate the other arcade games on the market, First Star has introduced a game which is refreshingly original—and very entertaining.

In *Panic Button*, you have been hired to assemble various objects whose parts parade on three continuously moving conveyor belts. On the first level, robot parts are ejected from the three chutes at the top of the screen. Not only must you catch up to them, but you must assemble them *in the proper order* to be given credit for the item. Should you accidentally place the robot's feet on its head (an improper sequence), no credit is given for the item, nor are its parts reusable, since there is no way to separate any two joined parts. I found it frustrating: No sooner had I completed two-thirds of an object than an incorrect part dropped from a chute and attached itself to mine. Surprisingly, this occurred in my favor as often as it did against me.

After a while, especially during the harder screens, these



Parts continuously flow from three conveyor belts in Panic Button *(Color Computer version).*

"rejected" objects (obviously thrown by your boss in anger) begin to fly around the screen, at times bumping into you and making your job even more difficult. I almost found it more than I could handle, having to race around the screen to retrieve objects moving nearly as quickly as I was.

Houses, Telephones, And Lamps

In later screens, you will find three-layered cakes, houses, telephones, televisions, and finally lamps dropping from the chutes. After every screen, it becomes progressively more challenging to complete your minimum order as the conveyor belts move faster and the number of objects you must assemble within the two-minute limit increases. After completing the second screen, you have a much-awaited opportunity to fling a pie into your boss's face—but that is not what gets you fired.

Panic Button breaks away from the three-man tradition and provides you with only one worker. Should he fail to fulfill his minimum order of assembled items, the boss spares no time in firing him (where's another pie?).

You have only one thing going for you in this game—the "panic button." You activate it by using the joystick button to move your character over to the operating switch. This slows the conveyors to a halt, allowing you to freely gather the objects around the factory. (Unfortunately, it has no effect on the clock, which continues to run down.) But your enraged boss soon comes to restart the conveyor belt, and you continue your frantic race against time.

An "external" panic button not mentioned in the rather skimpy documentation is the space bar: Pressed at any time during the game, it pauses the action indefinitely. I found myself using this panic button more than the other.

First Star's decision to develop a game with a unique concept is refreshing, but an original game is not always a good game. With *Panic Button*, however, First Star has succeeded. I recommend it to anyone who enjoys nonstop action—and even to those who do not. After all, that is the reason the "panic button" exists.

Panic Button
First Star Software, Inc.
22 East 41st Street
New York, NY 10017
Color Computer or VIC (8K expansion) tape, $24.95
VIC cartridge $34.95
Color Computer cartridge $39.95

©

ATARI®

600XL	CALL
800XL	CALL
1200XL	$409*

*Reflects $100 Atari Rebate

1050 DRIVE	$335	MICROBITS INFC	$78
1025 PRINTER	$399	80 COLUMN BD	$249
1020 COLOR PTR	$219	TECHNICAL NOTES	$25
1027 PRINTER	$279	REAL TIME CLK	$38
1010 RECORDER	$72	810 DRIVE	$419

DRIVES

ASTRA 1620	$469
RANA 1000	$295
PERCOM 88-S1	$298
PERCOM 40-S1	$418
PERCOM 40-S2	$718
PERCOM 44-S1	$510
PERCOM 44-S2	$929

MEMORIES

48K RAM (INTEC)	$95
64K RAM (INTEC)	$119
48K RAM (MOSAIC)	$109
64K RAM (MOSAIC)	$145
128K RAM DISK	$299
32K RAM (MOSAIC)	$68

ATARI SOFTWARE

ADVENTURE INT'L
- Adv. 1-12 each (C) $18
- Prepple (C/D) $20
- Prepple II (C/D) $23
- Diskey (D) $33
- Sea Dragon (C/D) $23

APX
- Eastern Front (C/D) .. $23
- 747 Land Sim. (C/D) .. $17
- Fig-Forth (C) $30

ATARI INC.
- Microsoft Basic II (R) . $62
- Mickey in Great Outdoors (C/D) $36
- Paint (D) $30
- Speed Reading (C) $54
- Qix (R) $30
- Dig Dug (R) $30
- Atari Writer (R) $68
- Donkey Kong (R) $30
- Time Wise (D) $23
- Visicalc (D) $139
- Juggles House (C/D) . $22
- Juggles Rnbw (C/D) . $22
- Pilot (Home) $55
- Galaxian $30
- Defender $30
- ET $34
- Microsoft Basic (D) .. $62
- Macro Ass. & Edit (D) . $62
- Assembler Editor (R) . $42
- Basic Cartridge (R) ... $45
- Pac Man (R) $30
- Centipede (R) $30
- Caverns of Mars (D) .. $28
- Star Raiders (R) $30
- Conv. Lang. Ea. (C) .. $42
- Music Composer (R) .. $31
- Super Breakout (R) .. $26
- My First Alphabet (D) . $26
- Prog. 2 & 3 (ea.)(C) $21
- Word Processor (D) . $102
- Pilot (Educ.) $92
- Touch Typing (C) $19
- Home File Mngr (D) ... $36

AUTOMATED SIMUL.
- Hellfire Warrior (C/D) . $27
- Kng Arthr's Heir (C/D) $20
- Invasion Orion (C/D) .. $17
- Temple of Aps. (C/D) . $27
- Star Warrior (C/D) $27
- Dragon's Eye (D) $20
- Crush Crumble (C/D) . $20

AVALON HILL
- VC (D) $17
- B-1 Nuc. Bomber (C) .. $12
- Legionnaire (C) $23

BRODERBUND
- Sky Blazer (D) $22
- Bank St. Writer (D) ... $46
- A.E. (D) $23
- Arcade Machine (D) .. $39
- Choplifter (D) $23

CBS
- Mountain King (R) ... $27
- Boulders & Bombs (R) $27
- Krazy (each) $34

CONTINENTAL SOFT.
- Home Accountant (D) $48
- Tax Advantage (D) ... $39

DATASOFT
- Text Wizard (D) $65
- Graphic Master (D) ... $27
- Micro Painter (D) $23
- Lisp Interpreter (D) .. $79
- Graphics Gen. (D) ... $17
- Basic Compiler (D) ... $65
- Zaxxon (C/D) $27

DON'T ASK
- Sam (D) $39
- P.M. Animator (D) $23
- Teletari (D) $27

EDU-WARE
- Prisoner II (D) $27
- Spelling Bee (D) $27
- Compu-Read (D) $20
- Compu-Math Fr. (D) . $27
- Compu-Math Dec. (D) . $27

EDUCATIONAL SOFT.
- Tricky Tutorial 1,2,3 or 4 (C/D) $15
- Tricky Tutorial 5,6 or 7 (C/D) $22

INFOCOM
- Suspended (D) $34
- Zork I, II or III (D) $27
- Starcross (D) $27
- Deadline (D) $34

JV SOFTWARE
- Jrny to Plnts (C/D) $20
- Action Quest (C/D) ... $20
- Ghost Encount. (C/D) . $20

LJK
- Letter Perfect (D) $74
- Data Perfect (D) $74

ON-LINE
- Ultima II (D) $39
- Marauder (D) $23
- Lunar Leeper (D) $20
- Wiz & Princess (D) ... $22
- Frogger (C/D) $23
- Crossfire (R) $23

OPTIMIZED SYSTEMS
- C-65 (D) $58
- Bug-65 (D) $23
- Max-65 (D) $58
- Basic A + (D) $58

ROKLAN
- Gorf (D) $27
- Gorf (R) $30
- Wizard of Wor (D) $27
- Wizard of Wor (R) $30

SIRIUS
- Alpha Shield (R) $27
- Wavy Navy (D) $23
- Bandits (D) $23

SPINNAKER
- Snooper Troop 1,2 (D) . $30
- Kindercomp (D) $20
- Rhymes & Riddles (D) $20
- Hey Diddle Diddle (D) . $20
- Srch Amzng Thngs (D) $27
- Story Machine (D) $23
- Face Maker (D) $23

STRATEGIC SIM.
- Cosmic Balance (D) .. $27
- Cosmic Balance II (D) $27
- Tigers In Snow (C/D) .. $27
- Battle of Shiloh (C/D) . $27
- Battle of Norm. (C/D) . $27
- Galactic Gladiator (D) $27
- Cytron Masters (D) ... $27

SYNAPSE SOFTWARE
- File Mngr 800 + ... $65
- Protector II (D) $23 (R) $29
- Shamus ... (D) $23 (R) $29
- Fort Apocalypse (C/D) $23
- Shamus II (C/D) $23
- Necromancer (C/D) ... $23
- Pharoh's Curse (C/D) . $23

THORN EMI
- Soccer (R) $34
- Jumbo Jet (R) $34
- Submarine Comm. (R) $34

USA
- Atari World (D) $39
- 3-D Sprgrphcs (C/D) .. $27

MISCELLANEOUS
- Sargon II ... (C) $20 (D) $23
- Financial Wizard (D) .. $41
- Castle Wolfenstein (D) $20
- Master Type (D) $27
- Millionaire (D) $52
- Astro Chase (D) $22
- Ali Baba (D) $22
- Miner 2049er (R) $34
- Sammy Sea Serp. (C) . $13
- Pinball (D) $20

SPECIALS

Gemini 15X Printer	$399
Axiom AT-100 Printer (with interface	$229
Astra Double Density Dual Drive	$469
Rana 1000 Drive	$319
Bit-3 80 Column Board	$245
Mannesmann Talley 160L Printer	$589
Atari 400 Keyboard (In Home)	$35
Programmer Kit .. $48 Entertainer Kit	$64
Wico Joystick ... $23 Wico Trackball	$49

*ASTRA 1620
DISK DRIVE SYSTEM

MORE FOR YOUR MONEY
DOUBLE OR SINGLE DENSITY
TWO DRIVES

SPECIAL $469

Printers/Etc.

GEMINI 10X	$279	PROWRITER	$345
GORILLA	$199	SMITH TPI	$488

CITOH
- Prowriter $345
- Prowriter II $629
- Starwriter $1149
- Printmaster $1448

NEC
- 8023 A-C $409
- 3510 .... $1375
- 3530 .... $1579
- 3550 .... $1779
- 7710/7730 .... $1998

- SILVER REED P $669
- QUME 11/40 + ... $1299

OKI-DATA
- Microline 82A $398
- Microline 83A $638
- Microline 84P $958
- Microline 92 .... $488
- Microline 93 .... $858

DIABLO
- 620R $939
- 630R $1719

MONITORS

AMDEK
- Color I $289
- V300 .... $139
- V300A $149
- Color II $449

NEC
- GRN (JB1260) $115
- GRN (JB1201) $155
- Color Composite ... $298
- RGB Color $598

MODEMS

HAYES
- Smartmodem $209
- Smartmodem 1200 .. $498
- Micromodem II $259

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

(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

CBM 64	CALL
1541 DISK DRIVE	$239

1701 Color Monitor	$255	1530 Recorder	$59
1525 Printer	$239	1600 Modem	$59
1520 Color Ptr	$169	1650 Auto Modem	$89
Card ? (Infc)	$60	CMB 64 Ref Guide	$18
Light Pen	$29	The Connection (Infc)	$85
Cassette Infc	$29	MSD Disk Drive	$339
Card ? Software	$16	PTI 45 Lot Board	$59

64 SOFTWARE 64

Multiplan	CALL
Script 64	$77
Calc Result Prof.	$114
Calc Result Easy	$68
The Home Accountant	$48
Delphis Oracle	$89
Word Pro 3 with Spell	$78

ACCESS SOFTWARE
- Neutral Zone (C/D) ... $26
- Sprite Master (C/D) ... $27

AVALON HILL
- Nukewar (C) $12
- Planet Miners (C) $12
- Androm. Conquest (C) $14
- Midway Campaign (C) $12
- North Atl. Convoy (C) . $12
- Comp. Stcks/Bnds (C) $15
- Computer Football (C) $18
- Telengard (C) $16

BATTERIES INCLUDED
- Paper Clip (D) $85
- Delphis Oracle (D) 89

BRODERBUND
- Choplifter (R) $29
- Serpentine (R) $27
- Seafox (R) $27
- David's Midnight (D) .. $23

COMMODORE
- Easy File (D) $75
- Easy Finance (D) $38
- Easy Mail (D) $38
- Easy Script (D) $75
- Easy Schedule (D) ... $59
- Logo (R) $75
- Pilot (D) $75
- Assembler (D) $38
- Music Machine (D) $25
- Music Composer (D) .. $25
- Meza Music (D) $75
- Video/Music Supt. (D) $38
- Jupiter Lander (R) $25
- Radar Rat Race (R) ... $25
- Sea Wolf (R) $25
- Kickman (R) $25

COMM-DATA
- Pakacuda . (C) $14 (D) $18
- Escp. MCP . (C) $14 (D) $18
- Centropods (C) $14 (D) $18

COMPUTERMAT
- Arcade-Pak (C) $18
- Education-Pak (C) $18

CREATIVE SOFTWARE
- Moondust (R) $25
- Trashman (R) $25
- Save New York (R) ... $25
- Astroblitz (R) $25
- Household Fin. (D) ... $25

DATA 20
- Video Pak 80 .... $139
- Z80 Video Pak $229

EN-TECH
- Finance Calc 64 .... $34
- Data Base 64 .... $56
- Invoice Ease 64 .... $56

EPYX
- Temple of APS (D) ... $27
- Upper Reach. APS (D) $14
- Jumpman (D) $27

HES
- HES Modem $59
- 6502 Prof.Dev.Sys.(C) . $22
- Hesmon 64 (R) $27
- Turtle Grapics II (R) .. $41
- Heswriter 64 (R) $32
- Gridrunner (R) $27
- Retroball (R) $27

INFOCOM
- Zork I, II or III (D) $27
- Deadline (D) $35
- Starcross (D) $27

JIN SAM
- Mini-Jini (R) $75

LITTLE WIZARD
- Pro.Mail.List (C) $22 (D) $25
- Stockmaster (Inventory) (C) $25 (D) $28

LOGISTIC
- Datacalc 64 (C) $55 (D) $59
- Home Journal (D) $55

MICROSPEC
- Payroll System (D) ... $73
- Inventory Pkg (D) $73
- General Ledger (D) $73
- Disk Data Mgr (D) $62
- Mail List Mgr (D) $41
- Checkbook Mgr (D) ... $39

M-SOFT
- M-File (D) $89

ON-LINE
- Frogger (D) $23
- Jawbreaker (D) $20

PACIFIC COAST SOFT.
- PCS (80 Col BD, Word Proc, D.Base,Spreadsheet) CALL
- Account PAC (C/D) ... $34
- File PAC (D) $30
- Editor PAC (D) $39
- Inquire PAC (D) $57
- Happy Tutor Typng (D) $18

PROFESS. SOFTWARE
- Wordpro 3 + /64 (D) ... $68

QUICK BROWN FOX
- Prof.Word Proc. (R) ... $50

RAINBOW
- Writers Assistant $95
- Spreadsheet Assist. .. $95
- File Assistant $95

SIRIUS
- Blade/Blackpoodle (D) $27
- Type Attack (D) $27
- Repton (D) $27
- Critical Mass (D) $27
- Snake Byte (D) $23
- Way Out (D) $27
- Fast Eddie (D) $23
- Turmoil (D) $23
- Spider City (D) $27
- Squish'Em (D) $23
- Final Orbit (D) $27
- Alpha Shield (D) $27

SKYLES ELEC. WORKS
- Busicalc (C/D) $52
- Busiwriter (D) $72

SPINNAKER
- Snooper Troops 1 (D) . $29
- Facemaker (D) $23
- Kindercomp (D) $20
- Hey Diddle (D) $20
- Most Amaz. Thing (D) . $27

SYNAPSE
- Fort Apocalypse (C/D) $23
- Survivor (C/D) $23
- Drelbs (C/D) $23
- Pharoh's Curse (C/D) . $23
- Protector II (D) $23
- Morgal (D) $23
- Shamus (D) $23

TAYLORMADE
- Touch Typing Tutor 3.0 (D) $21

TIMEWORKS
- Rbbrs/Lost Tomb (C/D) $21
- Wall Street (C/D) $21
- Money Manager (C/D) $21
- Data Master (C/D) $21
- Dungeons of Alg. Dragons (C/D) $21

TOTL
- Text 2.6 ... (C) $32 (D) $34
- Label 2.6 .. (C) $15 (D) $17
- Time Manager 2.6 (C) . $24
- Time Manager 2.6 (D) . $27
- Resrch Assist. 2.0 (C) . $24
- Resrch Assist. 2.0 (D) . $27

UMI
- Motor Mania (C) $20
- Renaissance (C) $27

VICTORY
- Annihilator (C/D) $16
- Kongo Kong (C/D) $16
- Trek (C/D) $14
- Adv. Pack #1 (C/D) ... $16
- Adv. Pack #2 (C/D) ... $16
- Grave Robbers (C/D) . $13
- Chomper Man (C/D) .. $18

Questions Beginners Ask

Tom R. Halfhill, Features Editor

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

Q What is a motherboard?

A A motherboard is the main circuit board of a computer. All other boards are connected to the motherboard.

The most important component on the motherboard is the central processing unit (CPU)—the central brain of the computer. The CPU is a microprocessor chip which performs or supervises all computer operations. It fetches each program instruction one at a time, executes it, stores the result, and then fetches the next instruction.

The motherboard also contains support chips required by the CPU: usually a video chip to control the TV display; input/output chips to handle the exchange of data with such peripherals as the disk drive, tape recorder, or printer; and perhaps a sound chip for music and sound effects.

In some computers—such as the Apple, Atari 800, and IBM PC/PCjr—the motherboard has long, narrow sockets called slots into which accessory boards can be plugged. Memory boards full of RAM chips (Random Access Memory) often fit into these slots. Other accessory boards (or cards) might include operating systems, disk drive controllers, printer interfaces, direct-connect modems, 80-column video expanders, graphics expanders, and even piggyback processors (boards with another CPU to allow the computer to run different types of software). That's why motherboards with several internal slots make a computer more versatile.

Some computers, including most home computers these days, contain only one circuit board—the motherboard. All the components are contained on this main board: the CPU, support chips, RAM chips, and ROM chips (Read Only Memory). Consolidating all the boards into one motherboard makes the computer smaller, lighter, and—most important from the manufacturer's point of view—cheaper to produce. For example, original Atari 800s contain six boards, and that's even before all the slots are filled with accessory boards. But the new Atari 800XL, which replaces the 800, contains only one board, even though it has more memory (64K RAM versus 8K–48K). Obviously, the 800XL costs less to manufacture.

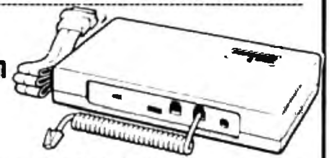
Of course, a computer without slots for accessory boards would not be as versatile. So single-board computers generally have an expansion slot or *system bus* on the rear. This allows accessory boards to be added externally. The accessory boards resemble large cartridges because they are enclosed in protective plastic or metal housings.

This still leaves one problem. How can more than one accessory board be plugged in at once? Naturally, there's a solution—an expansion box or motherboard extender. Both devices convert a lone expansion slot into several slots. For instance, you can expand a Commodore VIC-20 from the standard 5K RAM to 24K RAM by plugging a motherboard extender into the rear expansion slot, and then plugging 3K and 16K expanders into the motherboard extender.

Occasionally this is necessary even on computers with internal slots on the motherboard, such as the IBM PC. To fully equip a PC, sometimes the five internal slots just aren't enough. ©

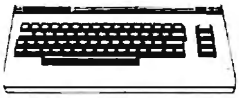
SPECIALS on INTEGRATED CIRCUITS
6502 @ 4.90 6520 @ 4.00 6522 @ 5.00 4116 @ 1.85
2532 @ 5.90 2716 @ 4.45 6116 @ 6.45 4164 @ 6.90

Anchor Automation Signalman MODEMS



FREE SOURCE MEMBERSHIP WITH SIGNALMAN
All Signalman Modems are Direct Connect, and provide the best price-performance values. **Dealer and OEM inquiries invited**

Volksmodem with computer cable	68
Mark VII Auto Dial/Auto Answer	99
Mark XII Smart Model 1200/300	279
DC HAYES Smartmodem	219
DC Hayes Smartmodem 1200/300	519



PROM QUEEN for C64 or VIC	130
Apple Emulator for Commodore 64	Call
STAT Statistics Package for C64	95
Solid Oak 2 Level Stand for C64 or VIC	29
C64/VIC Switch (networking)	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	72
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 SuperDrive for C64 or IEEE	365
MAE Assembler for C64	50
Koala Pad Touch Tablet—C64 or VIC	75
CBC 4/12 Analog to Digital 4 chan/12 bit	179
MULTIPLAN for C64	79
Dust Cover for C64 or VIC	6
Grand Master Chess for C64	24
COMAL Language for C64 with sprites, color graphics, sound, turtle graphics.	14
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 automatic switch between paddles and joystick	26
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	185
Solid Oak 2 Level Stand for Apple	29
Serial Card for Apple	89
MPC RAM/80 column card for IIe (AP/TXT)	139
Z80 Softcard and CP/M (Microsoft)	235
RANA Elite I with Controller	389
Parallel Printer Interface/Cable	69
Microtek and MPC Interfaces in stock	
Grappler + Interface	135
DC Hayes Micromodem II, IIe with Smartcom	245
PFS: File or PFS: Report or PFS: Graph	95
Videx 80 Column Card	209
Apple Blue Book	19

commodore

See us for Personal, Business, and Educational requirements. Educational Discounts available.

PETSCAN I $245 base price
Allows you to connect up to 30 CBM/PET Computers to shared disk drives and printers. Completely transparent to the user. Perfect for schools or multiple word processing configurations. Base configuration supports 2 computers. Additional computer hookups $100 each.

COMPACK/STCP $115
Intelligent Terminal Package for PET, CBM, C64
Includes ACIA Hardware / STCP Software

SCREENMAKER 80 Column Adapter for C64 139
Provide big screen capability for business applications.

Copy-Writer Word Processor for C64 49
Full-featured package with 800 lines of text in memory. Includes double column printing, graphic capability, full printer support.

Special Screenmaker/Copy-Writer Combo 179

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	
UMI products in stock	

FlexFile for PET/CBM/C64 $59
DataBase, Report Writer with calculations, Mailing Lists. Easy to use, and can be modified.

FORTH for PET/C64 full FIG model – Cargile/Riley 50
includes all FORTH 79 Standard extensions, structured 6502 assembler with nested decision macros, standard 16x64 screens, ability to read/write BASIC sequential files, sample programs, introductory + reference manual.

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 Disk-based ASSEMBLER 65

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

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/ 1.50	50/ 1.45	100/ 1.40
Wabash 5" ss/dd	10/ 1.80	50/ 1.75	100/ 1.65
Wabash 5" ds/dd	10/ 2.50	50/ 2.45	100/ 2.35

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	185
Quadboard for IBM available	
KOALA PAD Touch Tablets—Apple, Atari, IBM, CBM	
Peachtext 5000 Software Package	209
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	469
Brother HR-25 Daisy Wheel Printer (25 cps)	769
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	135
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.

David D. Thornburg, Associate Editor

Computers In The Workplace

I can't remember the exact occasion, but about three years ago my son (who was then seven years old) was being taken to lunch by a friend of mine in downtown Palo Alto. As the two of them walked down the street, my boy looked in the window of an office where he saw a woman typing some correspondence. "What is she doing?" my son asked. "She is typing a letter," my friend replied. At that, my son looked again and said, "That's funny; I thought only men typed."

I thought it was pretty funny too—for a while. My son knows that I spend a lot of time at a keyboard, much of it writing articles and books. In fact, I am happy he sees that keyboards are not the sole domain of female typists, but are becoming increasingly used by men. But any stereotype is dangerous; it is as dangerous for my son to think of men as typists as it is for women to be typecast in that role.

A Difference In Use

As I thought about the incident some more, it became apparent that there was perhaps a distinction in the ways that keyboards were being used by men and women, especially in business. In most businesses it appears that male keyboard users are using spreadsheet programs, or performing other analytical or forecasting activities with computers, while the majority of women employees are using keyboards connected to nothing more sophisticated (or career-enhancing) than an electric typewriter. In general, it appears that men compute and women type.

Because those who compute tend to earn more than those who type, it is worth exploring the potential of the business computer in eliminating sex-stereotyped jobs. I refer to sex stereotypes rather than discrimination because, as we shall see, a good portion of the job-selection process is induced by the very people who end up perpetuating the stereotype of women as typists.

No Access To The Professions

It is one of my pleasures to spend part of my time as a teacher. Sometimes my students range from third to sixth grade, and other times they are first-year graduate students in product design. In my graduate classes, I will often have only four or five women among my 40 students. Since product design is among the more "artsy" of the engineering fields, you would expect this number to be higher (assuming that you believe women are more interested in the arts than men).

In fact, I find it quite disappointing that there's such a small percentage of women. But the reasons for it are not hard to discern. In order to gain entrance to graduate school in an engineering field, students must have majored in engineering or the physical sciences in college. This, of course, requires a very solid background in mathematics.

As I look at the younger children I sometimes work with, I find that many of the girls are turned off to mathematics by the time they reach fourth grade, and that those who are not turned off have spent time with teachers who have a deep love and understanding of mathematics themselves. The mathphobia that sets in at an early age has a significant destructive power.

To allow any group to consider itself incapable of mastering mathematics is to essentially deny that group access to the professions. For whatever reasons, most of the high-paying technical, business, and medical professions require a significant number of advanced mathematics courses in col-

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.*

lege. By allowing some of our youngsters to become math illiterate, we are confining them to the lower end of the wage scale years before they seek their first jobs.

Working In A Man's Field

Unfortunately, mathematics is generally considered a man's field. In an attempt to counter this perception, Teri Perl wrote a book several years ago that should be on the shelves of every bookstore in the nation. This book is *Math Equals* (Addison-Wesley), a brief history of women in mathematics. Rather than presenting a dry historical treatise, Teri Perl portrays the women of her study as complete human beings and talks about their frustrations of being good in a man's field when they were expected instead to tend to matters of the home.

Of all the people who should read this book, among the most important would be the teachers of grammar school who pass on their own frustration and fear of mathematics to their female students, who in turn embrace them as their own.

But what does mathphobia have to do with men using computers while women type? The answer can be found in a myth that is as wrong as the belief that women aren't good at mathematics—that you need to be good at math in order to use computers. I would venture a guess that many of you are "good at computers," but are probably not "good at math." You already know that mathematics is not a prime requisite for computer literacy. And yet you are viewing the problem from the other side of the bridge—you have already made the passage.

Reinforcing The Myth

Imagine the plight of the woman with a degree in the arts or the humanities who wants to find a job in business. When offered an opportunity to learn about computers, many women say, "Oh, I couldn't learn how to use computers, I never was good at math"; or "I never was good at technical subjects." By making statements of this sort, these women are removing themselves from career paths that lead to high-paying jobs.

Because these fears are, in fact, unfounded, those who express them are allowing the persistence of a myth to restrict their professional growth.

While I don't know a sure-fire way to break through to people who hold themselves back in this way, two authors have done a marvelous job in trying to show working women the road to computer confidence and higher-paying jobs.

These authors are Dorothy Heller and June Bower, and their book is *Computer Confidence—A Woman's Guide,* published by Acropolis Books ($9.95 paperback). Because of the timeliness of its topic and its lucid style, this book deserves a wide readership. You could do your community a favor by seeing that your local bookstore has plenty of copies in stock.

A Highly Personal Book

As women who entered the computer field from backgrounds in the humanities, the authors have the rare perspective of those who have walked both sides of the street. The book is a highly personal account; in fact, it is the book they wish they had had (but couldn't find) when they entered the computer field. Topics range from a short history of women who "made it big" in computers, to case histories of working women who use computers without knowing how to solve partial differential equations. By blending case histories with enough technical data to make the reader a savvy shopper for computer technology the authors prepare the reader for the main goal of the book: to show women how they can enter career paths with unlimited upward potential.

This assistance covers the spectrum from worksheets to help the reader identify appropriate career choices, to practical tips on how to handle job interviews, and especially how to handle the inevitable objections that arise when the interviewer finds that the educational and work background of the applicant doesn't include the "right" degrees from the "right" schools. ©

Software Unlimited

KOALA SALE!

Koala Pads
Atari, C-64, Vic-20
w/Disk Software *$65.00*
Atari, C-64
w/ Cartridge Software *$73.00*
Apple II, IIe, II+ *$88.00*
IBM P.C. *$108.00*
Instant Programmers Guide
All Computers *$12.00*

THE Clone Machine
with "Unguard" C-64 .... *$32.00*
S.A.M., A, C, Disk *$39.95*
Chatterbee A, C, Disk *$27.00*

SOFTWARE UNLIMITED, 27 LOOKOVER LN., YARDLEY PA 19067. VISA & MC ADD 4%. PLEASE ADD $1.50 P&H. PERSONAL CHECKS REQUIRE 2 WEEKS TO CLEAR. CALL FOR A COMPLETE LIST OF OUR SOFT & HARDWARE.
(215) 493-1372
OUT OF STATE 1-800-225-3656

SAVE BIG ON COMPUTER PRODUCTS

MODEMS

HAYES	
Smartmodem 300	209
Smartmodem 1200	489
Smartmodem 1200B	425
Micromodem II E	240
Smartcom II Software (for IBM PC)	90
NOVATION	
J Cat 300B direct	99
103 SmartCat 300B Smart	159
103/212 Smart 300/1200 B	379
AppleCat II 300 baud for Apple	200
212 AppleCat 300/1200B for Apple	390
Access 1-2-3 1200B for IBM	379

MONITORS

GORILLA	
12" Green	88 00
12" Amber	94 00
TAXAN	
12" KG-12N (green)	120
12" KG-12N-UY (amber)	125
RGB vision III	499
AMDEK	
V300G 12" green	130
V300A 12" amber	145
V310A 12" amber (for IBM)	160
Color I + 13" composite	280
Color II 13" RGB	425
Color IV 13" RGB analog	740
Monitor Cables	
PA 400 for TI-99/4A or Commodore	15

QUADRAM

MICROFAZER	
QRMP-8 Par/Par	135
QRMSP-8 Ser/Par	145
QRMMSS-8 Ser/Ser	145
QRMPS-8 Par/Ser	145
QUADBOARD (for IBM PC)	
QR5064 64K	270
QR4064 64K	270
QUADLINK	
QR 3000 for IBM	475
QR 3010 for Compaq	475
QR 3020 for Columbia	475
QR 8201 Quadcolor-1	200
QR 8202 Quadcolor-2 (upgrade kit)	200
e Ram 80 — 80 col card for Apple IIe	115

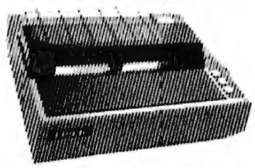
wabash maxell Dysan

DISKETTES

10 — 5¼" Floppy Diskettes
(packed with a storage box)

	SS/SD	SS/DD	DS/DD
Wabash	$19	$22 50	$30
Maxell	NA	27 00	39
Dysan	NA	27 00	39
10 — 8" floppy diskettes			
Maxell	NA	36 00	42

PRINTERS



Tremendous Discounts

Gemini 10X 275
Gemini 15X 400
Delta 10 390
Epson RX80 275
Other Epson models ...CALL
Panasonic 1090 .275
Most models IN STOCK of DIABLO • TTX MANNESMANN TALLY
CALL FOR PRICES

HEWLETT-PACKARD CALCULATORS

HP-11C	70	HP-15C	90
HP-12C	90	HP-16C	90
HP-41C	145	HP-41CX	245
HP-41CV	200	HP-97	560

all software & accessories too

PORTABLE COMPUTERS

HP-71B	399	HP-75	719

CABLES — INTERFACES

accessories for Computer Printers	
GRAPPLER PLUS	120
16K BUFFERED GRAPPLER	165
Apple Dumpling	70
Cardco G	65
10 ft Par Cable for IBM	32
10 ft 36x36 Parallel	32
10 ft 25x25 RS-232	25
Y Cable (for TI-99/4A)	25
6 ft TI-99/4A parallel cable	25
13x16 printer stand for 80 col. printers)	15
16x22 printer stand (heavy duty) for wide carriage printers	40

RIBBONS, PAPER

Paper — 1000 Sheet Pack	
9½ x 11 white, 20 lb	12
9½ x 11 green bar, 20 lb.	12
9½ x 11 trim edge, 20 lb.	15
14⅞ x 11 green bar, 20 lb.	19
Ribbon Cartridges	
for Epson 80 col.	4
for Epson 132 col	7
for Mann Tally 160	11
for Mann Tally 180	14
for M T Spirit 80	7 50
for Panasonic 1090	6 50
for Diablo Hytype II	5
for C. Itoh Starwriter	5

Twin spool ribbons
for Gemini 10/10X, 15/15X, Okidata 80, 82, 83
6 for 15 00 or 12 for 24 00

CALL TOLL FREE 800-621-1269 EXCEPT Illinois, Alaska, Hawaii

Corp. Accts. Invited. Min. Ord. $15.00. Mastercard or Visa by mail or phone Mail Cashier's Check, Mon Ord, Pers Check (2 wks to clr) Add $4 00 1st item (**AK, HI, P.R., Canada add $10.00 first item**) $1 00 ea add'l shpg & handl Shipments to IL address add 7% tax Prices subj to change **WRITE for free catalog. RETURN POLICY: Defectives Only: Most products replaced within 30 days of purchase with identical merchandise only. Computer and large peripherals replaced only when defective on arrival (within 3 work days of purchase). Other problems covered by mfr. warranty. ALL ELEK-TEK MERCHANDISE IS BRAND NEW, FIRST QUALITY AND COMPLETE.**

ELEK-TEK, inc. 6557 N. Lincoln Ave., Chicago, IL 60645
(312) 631-7800 (312) 677-7660

SAVE MORE THAN EVER ON...
3M Scotch® DISKETTES
AND OTHER COMPUTER NEEDS!

NEW! Lower Prices

3M Scotch® Diskettes

are boxed in 10's with labels, envelopes and reinforced hubs on 5¼" diskettes. **LIFETIME WARRANTY!**

$1.85 ea. 5¼" SSDD (744) Qty. 20

5¼" DSDD (745) **$2.35** ea. Qty. 20

5¼" SSQD-96TPI (746) **$2.60** ea.
5¼" DSQD-96TPI (747) **$3.25** ea.
8" SSSD (740) **$2.05** ea.
8" SSDD (741) **$2.50** ea.
8" DSDD (743) **$3.10** ea.

DISK CADDIES

...the original flip-up disk holder for 10 diskettes. Beige or grey only.

5¼" **$1.65** ea. + .20 Shpng. 8" **$2.29** ea. + .20 Shpng.

RIBBONS!
at bargain prices.

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

DISKETTE 70 STORAGE

The best buy we've seen! Dust-free storage for 70 5¼" diskettes with six dividers.

$14.95 + $3.00 Shpng.

FLIP 'N FILE

Redesigned for better appearance and greater ease of use. FLIP 'N FILE 25 holds 25 5¼" diskettes; FLIP 'N FILE 50 holds 50 5¼" diskettes.

FLIP 'N FILE 50
Retail $39.95
...DW Price
$26.95
+$5.00 Shpng.

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE!
Nationwide: 1-800-621-6827
Illinois: 1-312-944-2788
Hours: 9 AM-5 PM Central Time
Minimum Order: $35.00

Shipping: 5¼" DISKETTES—Add $3.00 per 100 or fraction thereof. 8" DISKETTES—Add $4.00 per 100 or fraction thereof. OTHER ITEMS: Shipping charges as shown in addition to diskette shipping charges. **Payment:** VISA or MasterCard. COD orders only, add $3.00. **Taxes:** Illinois customers, please add 8%.

DISK WORLD!

SUITE 4806
30 EAST HURON STREET
CHICAGO, ILLINOIS 60611

Authorized Distributor
Information Processing Products 3M

The Morning After: Anti-Computer Backlash And The Arrival Of The Mass-Market Home Computer Part 1

This is the text of the speech Fred delivered at the West Coast Computer Faire in late March. We are printing the speech in two parts.

We are at a watershed in home computing. The watershed has been caused by the computer price wars of 1983, the introduction of simple and inexpensive, yet powerful, new computer programs and peripherals, and the entry of IBM into the home computer market.

Over the next year, home computing users, vendors, and enthusiasts will divide into two major camps: the *computer intimates* and the *computer literates*. By the end of 1986 these two groups will have fused into a third camp: the *neo-programmers*, who will represent the bulk of the users of home computers through the next decade.

Literates Vs. Intimates

Hackers, computer professionals, old-line computer educators, programming teenagers, and computer hobbyists will make up the bulk of computer *literates*. Computer literates will stress the importance of learning how to program and learning how computers work. The computer itself will continue to be the prime concern of this group.

Computer *intimates* will far outnumber the computer literates. Computer intimates will consist of all the millions of Americans who were roped or forced into using computers and who demand that they be easier to use and more practical.

Computer intimates will believe that software and computer input devices are far more important than the computer itself. As a group they will preach ignorance of computer programming and ignorance of the computer's insides as virtues. The motto of the computer intimates will be: "You don't have to know how a computer works, only how to make it do work for *you*."

The Computer Freight Train

On December 6, 1983, I appeared on ABC's *Good Morning America* TV show as a computer expert. My task was to advise families on the type of computer they should purchase for Christmas. In less than seven and a half minutes I led the show's viewers and its two hosts, David Hartman and Joan Lunden, through a bewildering array of computer hardware and computer programs.

I am sure that when the segment was over, most viewers still couldn't tell the difference between a disk drive, a program recorder, or a touch pad. But I'll wager that they did have a better feeling for the risk involved in investing in a personal computer, for the daunting complexity of becoming a first-time user, and for the flood of computer products and the dearth of reliable guidelines for making a purchase.

"Most consumers see personal computers as a high-speed freight train," I told viewers. "They feel they have to take the risk of hopping on now,

UNDERWARE

Software that's priced UNDER the competitions'

HARDWARE SOFTWARE UNDERWARE

COMMODORE 64 & ATARI COMPUTER CLASSICS

ENTERTAINMENT
Just for You!

EPYX
- Temple of Apshai (D&C) ... $26
- Upper Reaches Apshai (D&C) ... $16
- Gateway to Apshai (CT) ... $26
- DragonRiders of Pern (CT) ... $26
- Jumpman (CT) ... $26
- Jumpman Junior (CT) ... $26
- PitStop (CT) ... $26
- Broderbund Choplifter (CT) ... $29
- Broderbund Lode Runner (D) ... $26
- First Star Flip Flop (D&C) ... $22
- First Star Bristles (D&C) ... $22
- HesWare Gridrunner (CT) ... $19
- Infocom Zork I, II, III (D) ... ea. $26
- Infocom Witness (D) ... $33
- Infocom PlanetFall (D) ... $33

PARKER
- Parker Q Bert (CT) ... $37
- Parker Star Wars (CT) ... $37
- Parker James Bond (CT) ... $37
- Parker Gyruss (CT) ... $37
- Parker Popeye (CT) ... $37
- Parker Frogger (CT) ... $37

SYNAPSE
- Synapse Blue Max (D&C) ... $22
- Synapse Fort Apocalypse (D&C) ... $22
- Synapse Necromancer (D&C) ... $22
- Synapse Pharh Curse (D&C) ... $22
- Synapse Zeppelin (D&C) ... $22
- Sega Star Trek (CT) ... $25
- Sega Congo Bongo (CT) ... $25
- Sega Buck Rogers (CT) ... $25
- Screenplay DunZhin (D) ... $29
- SubLogic Pinball (D&C) ... $22
- SubLogic Flight Simulator ... $37

EDUCATION

SPINNAKER
- Alf Color Caves (CT) ages 3-6 ... $26
- Alphabet Zoo (CT) ages 3-8 ... $23
- Delta Music (CT) ... $26
- Delta Drawing (CT) ... $26
- Face Maker (CT) ages 3-8 ... $23
- Kids On Keys (CT) ages 3-9 ... $23
- KinderComp (CT) ages 3-8 ... $19
- Bubble Burst (CT) ages 4-8 ... $26
- Grandma's House (D) ages 4-8 ... $23
- Story Machine (CT) ages 5-9 ... $26
- Ranch West (CT) ages 5-10 ... $26
- Fraction Fever (CT) ages 7-12 ... $23
- JukeBox (CT) ages 8-adult ... $26
- Aegean Voyage (CT) ages 8-adult ... $26
- Up For Grabs (CT) ages 8-adult ... $26
- Adventure Create (CT) ages 12 & up ... $26

HAYDEN (ages 4-10)
- MicroAddition (D&C) ... $20
- MicroSubtraction (D&C) ... $20
- MicroDivision (D&C) ... $20
- MicroMultiplication (D&C) ... $20
- Monkey See-Spell (D&C) ... $20

AMERICAN EDUCATIONAL COMPUTER (DISK ONLY)
- Learn About Sounds (grades K-3) ... $39
- Words In Reading I (grades 1-3) ... $39
- Words In Reading II (grades 1-3) ... $39
- Reading Compreh I (grades (1-4) ... $37
- Reading Compreh II (grades 3-7) ... $37
- Reading Compreh III (grades 6-8) ... $37
- Vocabul Word Bider (grades 2-8) ... $29
- Grammer Word Skill (grades 2-8) ... $29
- US Geography Facts (grades 2-8) ... $29
- World Geography (grades 2-8) ... $29
- Spanish Vocabulary (grades 2-8) ... $29

HOME APPLICATIONS
- Spinnaker Aerobics (D) ... $33
- CodeWriter Dialog (D) ... $45
- CodeWriter ELF Easy Lang. Form (D) ... $45

PERSONAL FINANCE
- Continental Home Accountant (D) ... $47
- Continental Tax Advantage (D) ... $33
- Continental FCM First Class Mail ... $29

ELECTRONIC ARTS
- Pinball Construction Set (D) ... $39
- M.U.L.E. (D) ... $39
- Worms? (D&C) ... $34
- Archon (D&C) ... $39
- Hard Hat Mack (D&C) ... $34
- Murder on the Zinderneuf (D) ... $39
- The Cut & Paste Word Processor (D) ... $49
- D Bug (D) ... $34
- Axis Assassin (D) ... $34
- Music Construction Set (D) ... $39

FREE DISKETTE with each purchase of Electronic Arts software

COM 64 EXCLUSIVES!

PERSONAL FINANCE
- Timeworks SwiftTax (D) ... $39
- Creative Household Finance (D) ... $23
- Creative Household Finance (C) ... $19
- HesWare Finance Manager (D) ... $49
- Cardco Tax Survival Program (D) ... $33

WORD PROCESSING
- Cardco Write Now! (D) ... $39
- Timeworks Word Writer (D) ... $39
- On-Line HomeWord (D) ... $39
- Creative Joe's Writer (D) ... $42
- HesWare OmniWriter & Spell (D) ... $49
- Blue Sky Script 64 & Spell (D) ... $69
- WordPro 3 Plus SpellRight (D) ... $69

SPREADSHEETS
- HesWare Multiplan (D) ... $75
- HesWare Omnicalc (D) ... $37
- MSI Practicalc (D&C) ... $35
- MSI Programmable Spreadsheet (D) ... $55
- Sim Home Calc (D&C) ... $26
- Creative Jack's Calc (D) ... $42
- Handic Calc Result Advanced (D) ... $75

DATABASES
- Creative Fred's Filer (D) ... $42
- TimeWorks Data Manager (D&C) ... $19
- TimeWorks Data Manager II (D&C) ... $39
- CodeWriter Home Filewriter (D) ... $55
- Entech Data Base 64 (D) ... $55
- MSI PractiFile (D) ... $39

BUSINESS SOFTWARE
- Cymbal General Ledger (D) ... $45
- Cymbal Accounts Receivable (D) ... $45
- Cymbal Accounts Payable (D) ... $45
- Cymbal Inventory Control (D) ... $45
- Cymbal Invoice Writer (D) ... $45

LANGUAGES & UTILITIES
- HesWare Hesmon 64 (CT) ... $26
- HesWare 64 Forth (CT) ... $45
- Acess Spritemaster (D&C) ... $23
- Timewks. Programg. Kit I, II, III (D&C) ea. $19
- Blue Sky Last One (D) ... $79
- Blue Sky 80 Column (D) ... $29
- Blue Sky Graphic Designer (D) ... $29
- Blue Sky 64 Statistics (D) ... $29
- Blue Sky Super Basic (D) ... $29
- Blue Sky Add On Basic (D) ... $29
- Blue Sky Super Copy (D) ... $29
- HesWare Graphics Basic (CT) ... $39
- HesWare HES CAT (D) ... $19
- HesWare HES FONT (CT) ... $16
- HesWare HES KIT (CT) ... $39

GAMES
Just for You!
- Acess Beach Head (D&C) ... $23
- Avalon Hill TelenGuard (C) ... $15
- Avalon Hill B-1 Bomber (C) ... $12
- Creative Crisis Mountain (CT) ... $25
- Synapse Sentinel (D&C) ... $22
- Synapse ZAXXON (D&C) ... $28
- ScreenPlay Pogo Joe (D&C) ... $19
- ScreenPlay Asylum (D&C) ... $23
- ScreenPlay K. Uston BlackJack (D) ... $57
- Quick Silva Ring Power (D&C) ... $23
- Quick Silva AquaPlane (D) ... $23
- First Star Boulder Dash (D&C) ... $25

VIC 20 HITS
- Creative Home Office (D) ... $22
- Creative Home Office (C) ... $19
- MSI Practicalc Plus (D) ... $35
- MSI Practicalc Plus (T) ... $33
- MSI Practicalc (D) ... $32
- MSI Practicalc (T) ... $29
- HesWare Vic Forth (CT) ... $39
- HesWare Hesmon (CT) ... $26
- Cardco Write Now ... $27
- Epyx Fun with Art (CT) ... $26
- Epyx Fun with Music (CT) ... $26
- KinderComp (CT) ... $23
- Story Machine (CT) ... $23
- Face Maker (CT) ... $23
- Kids On Keys (CT) ... $23
- Alphabet Zoo (CT) ... $23
- HesWare Turtle Graphics (CT) ... $26
- WizWare Turtle Tracks (C) ... $19
- Creative Choplifter (CT) ... $19
- Epyx Tempe of Apshai (C) ... $24
- Sega Congo Bongo (CT) ... $25
- Sega Star Trek (CT) ... $25

PERIPHERALS

GRAPHICS TABLET
- CHALKBOARD POWERPAD ... $79
  (C64 & Atari)

Software Packages
- Logic Master ... $27
- Leo's Links ... $27
- Programmers Kit ... $19
- BearJam ... $25
- KOALAPAD ATARI C64 ... $67

MONITORS
- BMC 13" Composite Color Plus ... $229
- BMC 12" Low Res Green ... $85
- USI 12" Green Hi-Hi Res ... $129
- USI 12" Amber Hi-Hi Res ... $139
- TAXAN 12" Green ... $99
- TAXAN 12" Amber ... $109
- NEC JB 1205 Amber ... $155
- Amdek Color I ... $287

PRINTERS
- NEC 8023A ... $379
- STAR GEMINI 10X ... $287
- ALPHA COM 81 ... $169
- Cable with AlphaCom ... FREE
- Okidata 82A ... $389
- Okidata 83A ... $589
- Transtar 1315 Color ... $489

WICO JOYSTICKS
- The Boss ... $14
- Bat Stick ... $19
- Red Ball ... $21
- Track Ball ... $30
- 3 Way Grip ... $21

MODEMS
- Anchor Mark 7 300 baud ... $129
- Novation J-Cat 300 baud ... $119
- Commodore 1600 Modem ... $58
- Commodore 1650 Auto M ... $155

CARDCO PERIPHERALS
- Numeric Keypad C64 ... $37
- Graphic Printer Interface ... $69
- Economy Printer Interface ... $39
- Commodore 64 5 Slot ... $49
- 16K Board ... $53
- Cassette Interface ... $29

DISK DRIVES

ATARI
- Rana 1000 176K ... $329
- Concorde Master 176K ... $329
- Concorde Slave 176K ... $269
- Concorde Master 348K ... $389
- Concorde Slave 348K ... $309

COMMODORE
- Concorde Master 174K ... $299
- Concorde Master 348K ... $369
- Commodore 1541 174K ... $269

BLANK DISKETTES

MAXELL DISKS
- MD1 SS DD Box of Ten ... $27
- MD2 DS DD Box of Ten ... $40

VERBATIM
- ValuLife SS DD Box of Ten ... $28
- ValuLife DS DD Box of Ten ... $42

$uch A Deal
12629 N. Tatum Blvd.
Suite 138
Phoenix, AZ 85032

CALL TOLL FREE 1-800-431-8697/602-957-3619

For Customer Service Call: 602-955-3857

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 piece, add $1.00 each additional piece. Hardware add 3% or $10.00 whichever is greater. Returns must have authorization number (call 602-968-9128 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 We pay shipping on backorders or they feel they will be run over or left behind."

The Hottest Thing Under The Christmas Tree

More computers were sold as Christmas gifts this year than in any year prior to 1983. By early 1984 over eight million Americans had personal computers.

Unfortunately, soon after Christmas, many of these Americans began suffering from "morning after" regrets and resentments. Too many Americans who had seen the slick commercials on TV and who had heard the daily press reports about the computer revolution were now wondering what they had gotten themselves into.

Most Americans have heard the word *software* but have only a vague idea what the word means. They have no understanding of what comprises a "complete" computer system. They have no appreciation of what operating or programming a computer entails.

Most Americans don't even know how to hook up a computer's cables, plug it in, or turn it off. I know of one family who finally turned their computer off at one in the morning, but who only did so after hours of agonized, fruitless searching of the manual. They were afraid they might break the computer if they turned it off the wrong way.

The Computer Kit

Why do people buy computers? Most Americans buy computers out of curiosity, for their work, to play games, or as an educational aid and tool for their children.

Most Americans buy computers at bargain-basement prices, usually at discount houses. Most Americans get their basic knowledge about computers from news stories and TV commercials.

When a person buys a computer, he thinks he has bought something equivalent to what he has seen on TV. He expects his computer to be able to do roughly the same things as the TV computer.

The average new-computer purchaser brings his computer home, struggles with the manuals, cables, and plugs, and finally powers the computer up. After all this effort, what does he get?

A blank screen.

After still more struggling with his manual, the astute newcomer finally realizes that what he has bought is a *kit*—like a bicycle or a puzzle that comes in a million pieces. Only it's worse. The kit's pieces are invisible. You don't get to see them until they appear on the computer's display screen after you have typed them in at the keyboard.

The pieces, of course, are the commands in the computer's BASIC programming language. Computer commands are more difficult to use than puzzle pieces for two reasons. First, puzzle pieces are combined in some sort of visual order to make up a picture. Second, pieces in a puzzle can usually be combined in only one way. And the picture fragment on each piece is a clue to where the piece belongs.

But computer commands are different. They carry no picture fragment that helps you see where in a picture (or a program) they belong. And they can be combined in an infinite number of ways. There is no set order to reach any given solution.

Most kits—for a bicycle, a lawn chair, a toaster oven, a sandbox, or swing set—come with explicit, printed directions. Computer kits don't usually come with printed directions. Instead, they come with a *dictionary* of commands organized, alphabetically, from A to Z. You get all the building blocks, but little or no help in how to put them together. And, before long, you realize, with a sinking feeling, that they can be put together in a million ways.

But where do you start?

Buying Half A Computer

It finally dawns on the consumer that what he has bought is only *half a computer*. Until he buys some software and some more equipment—a program recorder or disk drive, cassettes, disks, cartridges, and a printer—he can't do anything useful.

Of course this isn't exactly true. He can always assemble the kit himself. There are dozens of magazines and hundreds of books with prerecorded programs for his kind of computer. All he has to do is follow the blueprints—the listings—in the books and magazines, and soon he will be the proud owner of a real computer.

Of course he will need to spend dozens of hours entering in the programs, and dozens of hours more poring over the listings, trying to figure out why his programs don't work.

And he will have to invest in a storage device, so he can save his delicate, precious programs.

And he still needs a printer if he plans to use the computer as an electronic typewriter, bookkeeper, or filing cabinet, the three most popular home computer applications.

Voting No To The Home Computer

After the average consumer has forked over from $50 to $300, is he likely to invest another $100 to $1000 for additional hardware and software to "finish off" his computer?

After the consumer has made his purchase and found that he has only half a computer, is he likely to feel positively toward computers and computer companies?

After the average consumer has realized that he has bought a kit, is he likely to roll up his sleeves, master a programming language, or patiently enter in hundreds of lines of unintelligible commands?

The answer to all these questions, *for the average consumer*, is no.

The After-Christmas Backlash

Under these circumstances, the average person who bought or received a computer for Christmas is not likely to become a computer enthusiast. Instead, he is likely to become part of a growing anticomputer backlash.

More and more individuals and groups in society are coming to the conclusion that personal computers have not lived up to their promise. At the very least, they have not lived up to their commercials.

These individuals and groups are becoming more organized and outspoken. Like me, they see personal computers as a high-speed freight train, and they are set on derailing that train.

The other night I was listening to National Public Radio's "All Things Considered." A so-called computer expert was on the show decrying the use of computers in education. In his opinion, most people were using computers as fancy, expensive, electronic flash cards. He warned American parents and teachers that the computer industry was deceiving them in a major way.

Two nights later I read in *USA Today* that the American Academy of Pediatricians was warning against using computers with small children. The Academy reaffirmed its decade-old statement that "Advertising that promotes ... learning environments, programs, or systems is often guilt-producing, misleading and potentially destructive of human development and values." The Academy scolded parents who create a "superbaby syndrome" in which parents buy computers for small children and enroll them in computer classes even before they are toilet-trained.

Fighting Back

The American public has been dazzled by the glamour and high-tech chic of personal computers. On the surface, the public's attitude toward computers seems to have undergone a dramatic change. On the surface, it appears that most Americans approve of computers, if not for themselves, at least for their children. And even if they don't approve of them, they see them as inevitable.

This is, indeed, how Americans feel—*on the surface*. But what is going on beneath the surface?

I submit that the public's current attitude toward computers is superficial and can easily be changed. I further submit that the situation is becoming increasingly ripe for public opinion to take a swing in the opposite direction. This swing may be dramatic and quick.

The American public has been put on the defensive by the rapid spread of personal computers. But the public is likely to regain the offensive at the first opportunity. Beneath the thin veneer of approval lurk people's old prejudices and stereotypes against computers. These prejudices and stereotypes are fortified and aggravated by the bad experiences millions of people are having, firsthand, with computers.

The American public just needs a champion. As soon as groups and individuals appear who can articulate the public's feelings against computers, the public will rally around them. And then a major backlash against computers will begin.

A Consumer Uprising

People who are alienated by computers are not ignorant Luddites who oppose computers just because they are new and different.

Many people already oppose computers out of ignorance and prejudice. But many more may soon oppose computers because they feel computers have been misrepresented and oversold.

An anticomputer backlash may be in the cards. If so, it should not be viewed by those of us in the computer industry as an ignorant neo-Luddite rebellion. We should see it for what it is: a legitimate uprising by irate, unhappy consumers.

J. B. Shelton and Glenn M. Kleiman

Ready-to-Run Magazines

We met our first personal computer, an 8K PET, back in 1978. Soon thereafter we purchased one of the "new" PETs—a-state-of-the-art machine with 16K RAM memory, a full-size keyboard and a cassette recorder for external memory.

In those long gone days of almost six years ago, we eagerly sought information about our new machine, but little was available. It came with very little documentation, and what was provided was barely understandable. Today almost every bookstore has a large selection of computer books and even some drugstores carry computer magazines, but no books or magazines were readily available back then.

One source of valuable information was *Cursor* magazine, published by Ron Jeffries. Not a traditional magazine, *Cursor* arrived, somewhat irregularly, on a cassette tape. Each issue contained six programs that we could load and run right away. The programs were a mix of graphics and sound demonstrations, games, puzzles, programming utilities, educational programs, and simple applications programs (for example, for calculating mortgage rates). All the programs were at least reasonable; some were true gems.

A First Look

The programs in *Cursor* magazine gave us our first sense of the potential uses of personal computers. In addition, we could list and analyze the programs to learn new programming techniques. *Cursor* also has claim to being the all-time best buy in the personal computer industry. The price of a six-issue subscription was originally $20.

Cursor magazine continued publishing through May 1982. Copies of all 30 back issues are still available, and some of the programs have been made available for the Commodore 64. Another early cassette magazine for TRS-80 computers, *CLOAD*, continues to publish and is now available on disk also.

The idea of "magazines" of ready-to-run programs has grown. Two new magazines on disk have recently appeared, both focusing on education about and with computers. In this column, we review and compare *Microzine* and *Window*. Our reviews are based on the first three issues of *Microzine* and the second and third issues of *Window*. Both magazines are now available for Apple computers, and versions for other computers are being developed.

Microzine, Captivating For Children

Microzine, published five times a year by Scholastic, Inc., is designed for children ages 10 and up. Each issue contains four programs and a 48-page printed manual that supplements the onscreen instructions and provides additional ideas for using some of the programs.

One of the four programs in each issue is a Twistaplot story. These are stories in which the plot details and outcome are controlled by decisions the reader makes. For example, one issue contains a crime-solving adventure called "Mystery at Pinecrest Manor." This is an old-fashioned whodunit which makes the reader an active participant in the story. As the reader and participant, you study files containing background information about each of the suspects, search for clues, and spy on suspects. You play the part of a character in the story, deciding where to go and what to do at each choice point. You can reread the story many times, changing your responses and thereby encountering different events and outcomes each time.

The flexibility of the stories, excellent graphics, and the active role played by the reader make Twistaplots captivating for children. Interactive stories are an exciting new genre of fiction, and Twistaplots demonstrate some of the advantages of using computers to present these stories.

Educational Programs

Each *Microzine* also contains one or two computer *tool* programs. These provide a means for children to explore and learn about different uses of computers.

A Poster program provides a simple computer language for creating colorful, low-resolution

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.).*

CHART YOUR PROGRESS

SPECIALS OF THE MONTH

Atari 600XL	$189
Atari 800XL	$299
Atari 1030 Modem	$115
Rana 1000 Disk Drive	$339
1027 Letter Quality Printer	$299
Commodore SX64 Portable Computer	$959
Commodore Automodem	$ 95
Commodore Graphics Printer	$269
Verbatim Twin Pak	$ 5.25
Elephant Disks S/S	$19.50

HARDWARE SPECIALS

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

Monitors

Leading Edge Green Hi-Res 12"	$ 89
USI Amber	$169
NEC Green Screen	$169
Leading Edge Amber Hi-Res 12"	$ 89
Amdek Color 1+	$315

RECREATION

Q Bert/CRT	$ 35
Frogger/CRT	$ 35
Popeye/CRT	$ 35
Witness	$ 36
Planetfall	$ 36
Deadline	$ 36
Zork I, II, III	ea. $ 29
Enchanter	$ 36
Infidel	$ 36
Temple of Apshai	$ 27
Pitstop	$ 27
Starleague Baseball	$ 23
Starleague Football	$ 23
Blue Max	$ 23
Joust	$ 43
Robotron 2084	$ 33
Pole Position	$ 36
Ms. Pac Man	$ 36
B-1 Nuclear Bomber (C)	$ 12
Midway Campaign (C)	$ 12
Nukewar (C)	$ 12
Telengard (C)	$ 16
Jumpman Jr./CRT	$ 27
Pitstop/CRT	$ 27
Flight Simulator II/D	$ 39
Night Mission Pinball	$ 20
Sammy Light Foot (CRT)	$ 20
Apple Cider Spider	$ 23

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	$36
Mathblaster/D	$36
Spellakazan/D	$29
Crypto Cube/D	$29
Master Type/D-CRT	$29
Songwriter/D	$29
Alphabet Zoo/CRT	$22
Fraction Fever/CRT	$22
Delta Drawing/CRT	$22
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

BUSINESS

Atari

Atariwriter	$ 79
Visicalc	$179
Financial Wizard	$ 45
Letter Perfect	$ 69
Data Perfect	$ 79
Spell Perfect	$ 65
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

Commodore

Easy Script 64/D	$ 35
Easy Spell 64/D	$ 16
Easy Finance I, II ea./D	$ 16
Accts. Receivable/D	$ 35
Payroll/Checkwriting/D	$ 35
Assembler 64/D	$ 15
Logo/D	$ 49
Simon's Basic/CRT	$ 17
Bank Street Writer/D	$ 49
Home Accountant/D	$ 48
Tax Advantage	$ 45
FCM	$ 35
Paper Clip Word Processor	$ 89
Delphi's Oracle	$ 89
Practicalc 64	$ 40
Calc Result/Easy-CRT	$ 59
Calc Result/Advanced-D	$109
Multiplan/D	$ 65
General Ledger/D	$ 59
Data Manager/D	$ 17
Electronic Checkbook	$ 17
Swif tax/D	$ 35
Quick Brown Fox/D	$ 59
Quick Brown Fox/CRT	$ 56

COMPUTER OUTLET

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.

VISA **1-800-634-6766**

Information and Inquiries: (702) 796-0296 Order Status: (702) 369-5523.

SHIPPING: For fast delivery, cashier checks, money orders or direct bank wires. Personal and company checks, allow 3 weeks to clear. C.O.D. charges: $3 minimum or 1% on orders over $300. Nevada residents add 5¾% sales tax. Shipping charges based on weight $3 minimum. APO and FPO orders: $10 minimum and 15% on all orders over $100. School and business purchase orders welcome. All returns must be accompanied by return authorization number. Call (702) 369-5523 before returning goods for replacement. Prices reflect cash discount only and are subject to change. Catalogs. .50¢ U.S., $1.00 foreign.

pictures. This program is a good introduction to both computer graphics and some rudimentary programming concepts.

An Electronic Card Filer program demonstrates how computers can be used to store, sort, and retrieve information. This program is well designed for introducing data base and information retrieval concepts, but it is limited to small amounts of information. Each card, or record, can contain only five fields of information, with up to 25 letters or numbers per field.

Another tool program, Melody Maker, is for creating music on the computer. With Melody Maker you can enter notes over a two-octave range and have the computer play your song. You can also have the computer create a visual display to go with your music. One type of display shows a musical staff and the notes; other types of displays create colorful patterns. You can save your songs on disk to play again later.

No Editing Feature

The Melody Maker program can be very useful in helping children learn about reading music. Its main drawback is that it is difficult to change a song once you have entered it. You can go back and change any note to another note, but you cannot insert or delete new notes. Therefore, if you want to insert or delete a note at the beginning, you have to reenter the entire song.

There is also a program called Amazing Robot that is intended to introduce programming concepts. As you might expect, the commands the robot follows are like those of turtle graphics. You can instruct the robot to move forward or back a number of steps, or turn left or right a number of degrees. However, this robot does not draw with a pen, as turtles do. Instead, you command it to maneuver through different mazes and patterns displayed on the screen. This aspect of Amazing Robot is similar to *Karel the Robot*, which was reviewed in this column in January 1983.

Amazing Robot does introduce some programming concepts. But we found it to be awkward to enter and edit procedures. For example, if you make a typing mistake while entering a procedure or accidentally direct the robot to touch a wall, you are thrown out of the edit mode and have to use a reedit command. Amazing Robot does not encourage learning and exploration nearly as well as more complete programs such as Scholastic's *Turtle Tracks*, Spinnaker's *Delta Drawing*, or any of the available versions of Logo.

The remaining programs include one in which you select questions to see the answers actor Robert Macnaughton gave; a tutorial and simulation game about hot air balloons; a word game; and a chase game. None of these will teach children much or draw their attention away from *Pac-Man*, *Frogger*, or whatever videogame is their current favorite.

Window Is A Screen Magazine

Window, intended for adults as well as children, takes seriously its status as a magazine using the new medium of computers. No print materials are provided, except for a note about booting the disk and accessing the help screens. Everything else you need to know is shown on the computer screen.

Window provides a great deal of flexibility. It lets you take a guided tour of each issue. This is similar to skimming through a printed magazine. You control the speed of progress through the screens and you can stop, back up, or continue at any time. You can choose to explore any program further. While working with a program you can always stop and return to skimming or to the table of contents.

Each issue of *Window* has a central theme which is the focus of a feature program, one or more other programs related to the theme, several software reviews, columns, and some smaller programs called "window dressing." The themes of the two issues we have reviewed are data base programs and music programs.

Sample Data Bases

The feature program of the data base issue is called Notebook. It allows up to 20 fields in each record, and it lets you obtain hard copies if you have a printer.

Window also provides a variety of sample data bases for you to explore and extend. Several are examples of data bases students and teachers have created. There is also a data base called *clues*. This is used in conjunction with another program called Adventurefile, which is a computer mystery. To solve the mystery, you have to use the Notebook program and the clue data base. The sample data bases provide a good starting point for novices learning about data base programs and the varied functions they can serve.

The same issue contains reviews of two software packages, Geography Search and Dueling Digits. Magazines on disk are an ideal vehicle for software reviews. Not only are the programs described and evaluated, but you also get to see actual screen displays and use interactive demonstrations of parts of the programs. These reviews gave us a much better sense of the programs than any written review ever could.

Some Fun Features

The disk also contains two games, one a variation of Monopoly and the other a variation of Simon. The games are appropriately referred to as "window dressing," as they do not add a great deal to the magazine. Finally, there is a *VisiCalc* column. This provides a template for multiplication tables, but you have to have *VisiCalc* to use it.

The feature program on the music issue of *Window* is called Mini-Songwriter. This program overlaps in function with *Microzine*'s Melody Maker, but is different in style. You enter notes by moving a marker on a piano-like keyboard displayed on the screen and specifying the length of each note. You can play your songs, varying the speed as you go. You can easily edit and save songs. *Window* also provides sample songs and another program that uses the Mini-Songwriter. This is a Mystery Melody program that presents "name that tune" riddles.

There are comprehensive reviews of MECC's *Music Theory* program, Spinnaker's *Snooper Troops,* and Earthware's *Volcanoes* program. In the reviews, you get to try a set of "which note is wrong" problems like those presented by the MECC program; search for clues as you would in the actual *Snooper Troops* program; and see the type of data you would collect in the *Volcanoes* simulation program.

The rest of the disk contains an editorial about work with computer music and Logo at MIT; a sample of music created with the *Songwriter* program (the full version of the Mini-Songwriter, available from Scarborough Software); and a graphic demonstration of sorting algorithms. These are all interesting additions to the main features. There are also columns that provide *VisiCalc* templates and Logo procedures. These columns can be used only by people who have *VisiCalc* or MIT Logo.

Comparison of *Microzine* And *Window*

Both *Microzine* and *Window* are exploring new terrain. So far, *Window* has been more innovative in its attempt to use the new medium of the computer without support of any printed materials. We had no difficulty using any of their programs with the information available on disk. We enjoyed skimming through the programs and viewing *Window*'s experiments with different formats of displaying information on the computer screen. *Window* is inventively interactive—you interact with the computer in flexible ways with several programs.

Microzine is more conservative in its approach and depends upon printed materials to provide the instructions necessary for many programs. However, the print materials also provide useful suggestions for extending the computer activities.

In their first few issues, *Microzine* and *Window* have each provided simple data base and music programs, so these programs provide a good basis for direct comparisons. The programs in both magazines are suitable for introducing novices to using computers for data bases and for creating music. However, none of the programs can replace full data base or music creating programs.

Overall, the programs in the two magazines are comparable. *Window* has an edge in the flexibility of its data base program and the ease of editing in the music program. *Microzine*'s music program has more visual display options than *Window*'s.

While we do not find major advantages in either magazine's programs, there are important differences in the overall presentations of how computers can be used for data bases and music. *Window* provides sample data bases, songs, and games that use the data base and music programs; *Microzine* does not. These extras provide good demonstrations, help people get started, and show how each program can be used in many ways. So we tend to favor *Window*'s presentations of data base and music programs.

As for the other programs, *Microzine*'s Twistaplots provide good examples of interactive fiction and contain excellent graphics. There is nothing in *Window* that is directly comparable. On the other hand, *Window* contains useful reviews of programs and ongoing columns for *VisiCalc* and Logo users.

The producers of both magazines can be expected to continue to experiment, explore, and improve. In fact, improvements are already evident within the first few issues. Our reviews and comparison should be read as a report on the status of these magazines as of the first few issues. Exciting prospects lie ahead for both, and we expect to see many more ready-to-run magazines in the near future.

Cursor *Magazine*
The Code Works
P.O. Box 6905
Santa Barbara, CA 93160

CLOAD *Magazine*
P.O. Box 1448
Santa Barbara, CA 93102

Microzine
Scholastic, Inc.
P.O. Box 641
Lyndhurst, NJ 07071

Window, *Inc.*
469 Pleasant St.
Watertown, MA 02172

After this column was written, COMPUTE! Publications announced the availability of COMPUTE!'s GAZETTE DISK, premiering with the May 1984 issue of COMPUTE!'s GAZETTE. For more information, call TOLL FREE 800-334-0868 (in North Carolina 919-275-9809).

Richard Mansfield, Senior Editor

A Random Leap

One of the enjoyable things you can do with a computer is simulate real events: things which might be too dangerous, too expensive, or too time-consuming to try in real life. The Air Force and some commercial airlines use a flight simulator so true-to-life that it can serve for all but the most advanced pilot training.

We don't have enough RAM memory, or the computation speed, or the ultrahigh resolution screens necessary to create a flight simulation of breathtaking realism. But we can try a simple simulation and get a feel for how they are programmed. The basis of the simulation will be accidental, unpredictable events created by the RND (random) command in BASIC.

Lurching Across A Bridge

Imagine a frog, lurching across a bridge. Every time he leaps, you don't know if it will be to the left or to the right. He doesn't know either. The one thing you can count on is that he will never leap straight ahead.

There are three possibilities in this game. He will either fall off the left or right side of the bridge, or safely reach the other side of the river. For this simulation, we're going to assume that the bridge is as wide as your computer screen and that the frog starts his journey midway between the left and right sides. That gives him a fair chance to make it across.

By setting up this simulation, we'll learn how to make use of the RND command as well as a way to animate characters on the screen. Let's look at the program line by line, to see what each BASIC command contributes to the overall effect. (Atari computers don't have a TAB command, so the animation technique discussed below will not work on them.)

First, we've got to define the size of the bridge, its width. Leave line 100 as it is if you have a Commodore 64 or any other computer which allows 40 characters per screen line. If you have a VIC, you should change line 100 to read: COLS =22. The VIC has 22 characters per screen line. If you have a TI, change it to: COLS=32.

The variable Y in line 110 is going to signify the position of the frog each time it leaps. If Y is raised to a higher number, the frog will appear further to the right on the screen (and be nearer the right side of the "bridge"). If Y goes down, if something is subtracted from Y, the frog moves left. At the start of the game, though, we want to put the frog in the middle between the left and right sides of the bridge so we divide COLS by 2. If you've got a 40-column screen, Y starts off equaling 20. That means that the frog is 20 from the right edge and 20 from the left—smack in the middle.

Rounding Numbers

The variable X in line 130 will tell us whether the frog should leap to the right or the left each time he leaps. This is the only complicated-looking line in the program, but it contains an important trick: the INT command. It "rounds off" a decimal number. INT (12.3) becomes 12. INT (12.7) becomes 12. Wait a minute. That's not *rounding off* as we usually think of it. 12.7 should become 13 since .7 is closer to 13 than to 12.

In fact, INT merely throws away anything to the right of the decimal point. This isn't true rounding. That's why we need to add the +.5 in line 120. By adding .5, we force a number to be rounded correctly by INT. 12.7+.5 would be 13.2 and INT (13.2) would give us the right answer: 13. Likewise, 12.3+.5 would be 12.8 and INT (12.8) would give us the correctly rounded answer: 12.

It's not important to remember *why* you need to add .5 to any number you want rounded by INT; just remember to do it. In line 120 we're not rounding off 12 or 13, all we want is an answer that tells us to go in one of two directions, to go either left or right. This is like tossing a coin, you get heads or tails. So here X will be either a 0 or a 1 after INT gets through rounding off RND(0). But what does RND(0) do for us? It creates a random number. But, by itself, the random number is a decimal fraction between 0 and 1. Try this:

```
10 PRINT RND(0):GOTO 10
```

When you RUN this, you'll see a series of decimal fractions, all kinds of different numbers. How would you get higher random numbers? Just multiply RND(0) by something. Try: PRINT RND(0) * 10. If you just want whole numbers (called *integers*), use INT.

Anyway, in our frog simulation we don't need these higher random numbers. If X becomes a 0 in line 120, we move the frog to the left (in line 160). If X becomes a 1 in line 120, we move the frog to the right (in line 140). Line 130 is the test to see which number is in X.

Notice that we don't need to write a line like: IF X=1 THEN 140. You *could* write that test and put it in line 135 if you wanted to. It wouldn't do any harm. But you don't need to. The computer will go to line 140 all by itself if X is anything other than a 0 when it's tested in line 130. The computer always performs each action in the order listed *unless you force it not to* with a GOTO, IF, or GOSUB command. If it doesn't come across one of those commands, it will go from line 140 to 150 to 160 and on up the list in simple line-number order.

Also on line 120 is another counter, the variable C. It will keep track of the total number of leaps the frog has made (either left or right). This lets us know how far he got before he fell off. It also sometimes shows that he's won the game. If he manages to leap a certain distance without falling, he's crossed the bridge.

But back to our simulation. After lines 130–160 make an adjustment to variable Y (our "position-of-the-frog" counter) we come to a series of tests in lines 170–190. Each of these tests will end the program in a different way. In 170, if the frog position is greater than (>) the total number of columns, he has fallen off the right side. In 180, if his position is less than 1, he has fallen off the left side. And, finally, in line 190, if he has taken more leaps than the width of the bridge, he made it across. You can change this line if you want to make it harder for him to cross the bridge. Just replace COLS with a higher number.

Line 200 prints the frog symbol on the screen to show us his position. The TAB command is just like a TAB key on a typewriter: It moves over a certain number of spaces from the left side of the screen. In this case, the number of spaces is controlled by the position variable Y.

Finally, to slow the frog down a bit, we put in line 210. This is often called a *delay loop* or a *do-nothing loop* because it simply takes up some time and serves no other purpose. Here we're asking the computer to count from 1 to 10 before going back down to line 120 and figuring out the frog's next leap.

```
100 COLS=40:REM PUT YOUR SCREEN SIZE HERE
                                     :rem 232
110 Y=COLS/2                         :rem 186
120 X=INT(RND(0)+.5):C=C+1           :rem 176
130 IFX=0THEN160                     :rem 174
140 Y=Y+3                            :rem 226
150 GOTO170                          :rem 103
160 Y=Y-3                            :rem 230
170 IFY>COLSTHENPRINT" >>>FROG FELL OFF R
    IGHT SIDE. IN"C"LEAPS.":END      :rem 120
180 IFY<1THENPRINT" <<<FROG FELL OFF LEFT
     SIDE. IN"C"LEAPS.":END           :rem 30
190 IFC>COLSTHENPRINT"FROG SAFELY CROSSED
     THE BRIDGE!":END                :rem 160
200 PRINTTAB(Y)"*"                    :rem 14
210 FORT=1TO10:NEXTT                  :rem 13
220 GOTO120                          :rem 96
```

BASIC Style— Program Evolution

Jim Butterfield, Associate Editor

Sometimes you see programs that are so crisp and neat that you wonder how the programmer's mind can be so orderly. The statements come out in an elegant, incisive style. Every line zeros in on exactly the right thing to do.

How does a programmer develop an elegant style? Why can't you write like that? Sometimes a lowly hacker can feel inferior when facing such immaculate programming style. Yet the program you see is often a matter of evolution—rewriting and tidying up. It's not always written that way from the beginning.

I have been accused of writing "squeaky clean" programs. It seems to me that you might like to see how my murky first programs get reworked and tightened up into their final version. In some ways, programming style isn't what you write (at least at first)—it's knowing what to look for when you clean up.

A Simple Lister

I needed to do an almost trivial job: list a file from disk to the printer. I had a minor extra feature to add: I wanted individual pages, so that the lines needed to be counted; I needed a title on each page; and at the end of the run, for the sake of neatness, I wanted the printer to eject the page.

It's not a demanding task, but I'd like to show you how I went about it. Even a simple job like that can be revised and tightened up extensively.

Here's my first program: I'll talk my way through the listing.

```
100 OPEN 4,3
```

Open file number four to the screen. Why? So I can send the program's output to the screen and see that it's working right. After the program looks good, I'll change the above line to OPEN 4,4.

```
105 OPEN 1,8,3,"CONTROL"
```

That's my input file to be listed.

```
110 REM START OF PAGE
120 FOR J=1 TO 2:PRINT#4:L=L+1:NEXT J
130 PRINT#4,"{5 SPACES}TITLE{3 SPACES}":L
    =L+1
140 PRINT#4:L=L+1
```

This prints the page title. I know I'll come back here for each new page, so I'm placing a REM statement here to mark the place. I rigorously add 1 to the line count, L, each time I print a line.

```
150 INPUT#1,A$:SW=ST
170 PRINT#4,A$:L=L+1
```

Here's where I input from disk and output (to the screen first, later to the printer). I need to save the value of ST (the status variable) so that later I can check to see if this is the last line from the file. ST will be changed by the PRINT# command, so I save its input value in variable SW.

```
180 IF L<62 GOTO 250
190 IF L=66 THEN L=0:GOTO 250
200 PRINT#4:L=L+1:GOTO 190
```

If I have printed the maximum number of lines desired, I want to eject the paper by printing until the line count L equals 66. Since each page has 66 lines, I'm now at the start of the next page and can set L back to zero.

```
250 IF SW<>0 GOTO 300
260 IF L=0 GOTO 110
270 GOTO 150
```

If I'm at the end of the input file (SW = 0), I'll go to line 300 and wind things up. Otherwise, I want to go back.

Here's a cute touch—perhaps too cute for some tastes. Variable L can only be equal to zero if I've just ejected a page. If so, I want to go back

GET THE MOST OUT OF YOUR COMMODORE-64 OR VIC-20 COMPUTER

ULTRABASIC-64...Add 50 commands: graphics, music, TURTLE and game features.Tutorial,demo plus. **TAPE $39.95 DISK $42.95**

SYNTHY-64... Sets the standard for all of the rest. Best 64-synthesizer anywhere. Samples and manual. **CASSETTE $29.95 DISK $32.95.** Also available: 3 great companion music albums; **Classical, Christmas,** and **Ragtime Sing-Along. DISK $12.95 Each.**

CHARTPAK-64...Professional qualtiy pie, line and bar charts. Menu driven, interactive, hardcopy. **DISK $42.95**

ZOOM PASCAL-64...Produces 6502 machine code for speed. Floating point, Integers, strings File handling. **DISK $39.95**

SUPER DISK UTILITY-64... Speed copy 4 ways: Total, Bam, Append or File. Dump or modify sectors. More. **DISK $22.95**

ASSEMBLER-MONITOR-64 High speed language development. Eleven function monitor. Screen editing of source file. **DISK $32.95**

GRAPHICS DESIGNER-64... Menu-driven drawings, floor plans and illustrations etc.. Slide program capability. **DISK $32.95**

TINY FORTH-64/20...Exciting language-low price. Powerful, extensible, 200 + word vocabulary. **TAPE $24.95 DISK $27.95**

CHARTPLOT-64...Same fine features as CHARTPAK-64 with high quality output to plotters. **DISK $84.95**

POOL-64/20...Play Fullrack or nine ball using hires graphics. Vic-20 required 8K expander. **TAPE $14.95 DISK $17.95**

SCREEN GRAPHICS-64 Adds 24 hires, multicolor, sprite commands to 64-BASIC. Demo, tutorial and manual. **TAPE $24.95 DISK $27.95**

MERCURE-64...Simple, powerful file management with fast design, entry search report capabilities. Tutorial. **DISK $32.95**

CHECKBOOK MANAGER-64 Simple check account maintainance. Optional screen or printer report and backup. **DISK $22.95**

ANATOMY OF A COMMODORE-64 Complete guide. Full comment ROMS list, detailed internals, descriptions. **300 PAGE BOOK $19.95**

MASTER-64...Full ISAM file management; powerful screen management; excellent printer generator; programmer's aid; BASIC 4.0 commands; machine language monitor; Software developers: NO RUNTIME ROYALTIES; With 150 page manual in three-ring binder and development software. **SOFTWARE ON DISK $84.95**

✱ DEALER INQUIRIES INVITED

FREE CATALOG Ask for a listing of other Abacus Software for Commodore-64 or Vic-20

DISTRIBUTORS

Great Britain: ADAMSOFT, 18 Norwich Ave, Rochdale, Lancs, 01-788-8963

Belgium: Inter. Services, AVGuillaume 30, Brussel 1160, Belgium, 2-660-1447

France: Micro Application, 147 Avenue Paul-Doumer, Rueil Malmaison, France, 1-732-9254

Canada East: KING MICROWARE LTD., 5950 Cote des Neiges, Montreal, Quebec H3S 1Z6, 514/737-9335

West Germany: DATA BECKER, Merowingerstr 30, 4000 Dusseldorf, 0211/312085

Sweden: TIAL TRADING, PO 516, 34300 Almhult, 476-12304

Australia: CW ELECTRONICS, 416 Logan Road, Brisbane, Queens, 07-397-0808

New Zealand: VISCOUNT ELECTRONICS, 306-308 Church Street, Palmerston North, 63-86-696

AVAILABLE AT COMPUTER STORES, OR WRITE:

Abacus Software

P.O. BOX 7211 GRAND RAPIDS, MICH. 49510

For postage & handling, add $1.50 (U.S. and Canada), add $3.00 for foreign. Make payment in U.S. dollars by check, money order or charge card. (Michigan Residents add 4% sales tax).

FOR QUICK SERVICE PHONE 616-241-5510

to 110 and print a new title. If not, get another line from the input file starting at line 150.

```
300 IF L<>0 GOTO 190
```

Here's a supercute trick. I pondered this one for a while, since it's almost too clever; that sort of thing can trip up your logic. Here's the objective: If we're finished, but the paper hasn't been ejected, go back to line 190 and eject the paper. The program will branch back here again, but this time variable L will be zero and we can finish the job by closing the files.

```
310 CLOSE 1
320 CLOSE 4
```

That's it. It's really rather messy. It works, and for a temporary job that's all we would need.

But it doesn't feel right. The code feels messy: It seems to jump around, and I don't get a feeling of smoothness in the program. It's time to pick at the coding.

First Revision

The first awkward spot is around lines 190 and 200. The routine to eject the paper works but looks clumsy. Besides, we call it twice (once at 62 lines, and again at end of file).

I have feelings about this part of the program, too. It's a unit to do a particular job. I would feel better moving it to a separate subroutine where it can stand out as an identifiable action. Sometimes I create a subroutine out of some in-line code and then move it back later; it helps me identify the modules that make up the program. Let's move the eject routine to a subroutine at line 500, clean it up a bit, and see what we get:

```
100 OPEN 4,3
105 OPEN 1,8,3,"CONTROL"
110 REM START OF PAGE
120 FOR J=1 TO 2:PRINT#4:L=L+1:NEXT J
130 PRINT#4,"{5 SPACES}TITLE{3 SPACES}":L
    =L+1
140 PRINT#4:L=L+1
150 INPUT#1,A$:SW=ST
170 PRINT#4,A$:L=L+1
180 IF L<62 GOTO 250
190 GOSUB 500:GOTO 250
250 IF SW<>0 GOTO 300
260 IF L=0 GOTO 110
270 GOTO 150
300 IF L<>0 GOTO 190
310 CLOSE 1
320 CLOSE 4
330 END
500 FOR J=L TO 66:PRINT#4:NEXT J
510 L=0:RETURN
```

We can see that the GOTO 250 on line 190 is now redundant since we'll go there anyway. But we have other things to do. We're still trimming the program and have some distance to go yet.

Digging Deeper

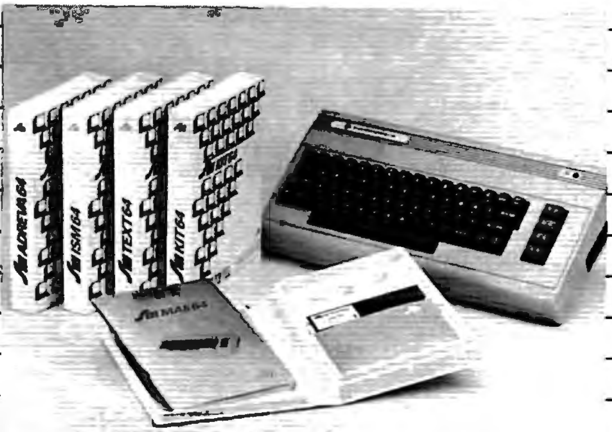
Around lines 250 to 270, we jump around a lot. We have one jump forward to 300 and two jumps back to 110 or 150. The logic seems scattered.

I have a thing about loops: I like to see them neatly nested, with short jumps entirely within longer jumps. It might even be summarized as a rule of thumb: Where possible, make short jumps as short as possible.

Using this rule, I want to get the loop back to 150 into logical order first. Then we'll work in the longer loop to 110 and finally the forward branch to 300. We'll need to expand the logic using an AND operator, but that's not too hard.

As the routine is written, certain logical things start to fall together. For example, we don't have to GOTO forward to line 300. When we're finished writing the two loops, we'll fall into 300 naturally. ("Naturally" seems to be a key word in how programs seem to come together as you tighten them up.)

We can also tighten up the page eject conditions. If we write line 180 correctly, there will be no need to go back to get a page ejection. One option would be to call the subroutine at 500 twice. But if we think of what our objective really is at line 180, we can do it all correctly the first time through. Inverting the logic and adding an OR connective does the trick nicely.



Look at how far the original program has come:

```
100 OPEN 4,4
105 OPEN 1,8,3,"CONTROL"
110 REM START OF PAGE
120 FOR J=1 TO 2:PRINT#4:L=L+1:NEXT J
130 PRINT#4,"{5 SPACES}TITLE{3 SPACES}":L
    =L+1
140 PRINT#4:L=L+1
150 INPUT#1,A$:SW=ST
170 PRINT#4,A$:L=L+1
180 IF L>61 OR SW<>0 THEN GOSUB 500
250 IF SW=0 AND L>0 GOTO 150
260 IF SW=0 GOTO 110
310 CLOSE 1
320 CLOSE 4
330 END
500 FOR J=L TO 66:PRINT#4:NEXT J
510 L=0:RETURN
```

This is pleasing, but we can do even more. The repeated SW=0 test in lines 250 and 260 still irks a little: It seems clumsy. The whole business is tied up with whether to print a title or not. Is there a better way? Could the test of L>0 be somehow shuttled up to the top of the loop instead of sitting at the bottom?

The Header Module

While we're thinking about it, that whole business of printing a header is really a module—we must do the whole thing, title and all, or nothing. If we move it out to a subroutine, we might see the logic flow more clearly. Let's do it and work on the logic flow. We end up with this:

```
100 OPEN 4,3
105 OPEN 1,8,3,"CONTROL"
110 IF L=0 THEN GOSUB 600
150 INPUT#1,A$:SW=ST
170 PRINT#4,A$:L=L+1
180 IF L>61 OR SW<>0 THEN GOSUB 500
260 IF SW=0 GOTO 110
310 CLOSE 1
320 CLOSE 4
330 END
500 FOR J=L TO 66:PRINT#4:NEXT J
510 L=0:RETURN
600 FOR J=L TO 2:PRINT#4:L=L+1:NEXT J
610 PRINT#4,"{5 SPACES}TITLE{3 SPACES}":L
    =L+1
620 PRINT#4:L=L+1
630 RETURN
```

Look at that main section from lines 100 to 330. It now seems tight and concise like a finely tuned instrument.

Both subroutines—at lines 500 and 600—are called only once. If it seemed important, we could put them back into the main program stream. But I'm happy to see them as clearly isolated modules. At this stage I would add comments (line 499: REM PAGE EJECT and line 599: REM PAGE TITLE) to neaten things up.

Moral

First, what you see published is not always the first idea that popped into the author's head. The programmer is not always smarter than you. Time has been taken to groom the program into its final shape. When many people are going to read your code, you like to take a few extra pains with its appearance.

Second, don't be afraid to revise your programs, even if they work correctly. Sure, a one-shot program often doesn't warrant picking over; use it and forget it. But sometimes the exercise can reveal, almost accidentally, powerful and effective programming methods.

Third, *style* isn't an inborn talent that some people have and some don't. You learn it as you go. Some things you will discover for yourself, and others you'll pick up by looking at other people's programs.

The odd thing is that we instinctively recognize better writing when we have written it. You may not know exactly why, but you often feel good about a certain piece of programming. Usually, it's because it has style.

VIC/64 Memdata

Michael M. Milligan

"Memdata" converts a machine language routine into DATA statements and then erases itself, allowing you to save the DATA to disk or tape for later use.

Transferring a machine language routine into DATA statements involves a lot of work. To simplify the job, "Memdata" takes memory bytes between two addresses, inclusively, and returns DATA statements complete with BASIC line numbers. Once the program has generated those statements, it automatically erases itself, leaving only the DATA—as you will see by typing LIST after the program is run.

The first part of Memdata is a modified version of Jim Wilcox's "Automatic Line Numbers" (*COMPUTE!'s First Book of VIC*). The line numbers are the decimal value for the address of the first byte in each line. This serves as a marker to be sure that every location is accounted for. Also, because many machine language subroutines are located at the top of RAM, it makes the data line numbers high enough to be appended to an existing BASIC program. The appending can be done with the Datassette or disk files, thus eliminating a lot of typing.

Once you save the DATA statements you have created, enter NEW and PRINT PEEK (43),PEEK(44). Write down these two numbers. LOAD the program to which you wish to append the DATA. Next, enter POKE 43,PEEK(45)–2:POKE 44,PEEK(46). Then, LOAD the DATA statements right in there with the first program. When it is loaded, POKE 43 and 44 with the numbers you wrote down after the earlier PEEK(43) and PEEK(44). This will merge the two programs if the DATA statement line numbers are higher than the highest line numbers in the original program.

Memdata erases itself in a novel way. Because line numbers used in Memdata are so high, the DATA statements will be the first lines in the BASIC program area. After the DATA statements are created, Memdata searches memory for DATA (token 131) following a line number. When it finds something besides a DATA token, it POKEs zeros into the high and low bytes of the link address for that line. These two zeros, plus the zero byte that signals end-of-line, make up the three zero bytes that convince the LIST and SAVE functions that the end of the BASIC program has been reached. Because of this, it's important to save the program before you run it for the first time.

Memdata

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

```
63720 PRINT"{CLR}{21 I}":PRINT"{RVS}TO
      {SPACE}CONVERT MEMORY TO {OFF}"
                                  :rem 159
63723 PRINT"{RVS}DATA STATEMENTS ENTER
      {OFF}":PRINT"{21 U}"         :rem 61
63730 PRINT"INCLUSIVE DECIMAL":PRINT"MEMO
      RY LOCATIONS":PRINT:INPUT"FROM";A
                                  :rem 138
63733 PRINT:INPUT"TO";C:PRINT:INPUT"BYTES
       PER LINE";B                :rem 170
63735 C=C/256:POKE251,(C-INT(C))*256:POKE
      252,C                        :rem 60
63740 POKE2,B:PRINT"{CLR}";       :rem 172
63750 B=A/256:POKE253,(B-INT(B))*256:POKE
      254,B                        :rem 55
63755 PRINT:PRINTMID$(STR$(A),2,LEN(STR$(
      A))-1);"DATA ";             :rem 247
63760 FORI=0TOPEEK(2)-1            :rem 76
63763 A$=STR$(PEEK(A+I))+","      :rem 223
63765 IFA+I>PEEK(251)+256*PEEK(252)GOTO63
      780                         :rem 221
63768 PRINTMID$(A$,2,LEN(A$)-1);    :rem 7
63770 IFA+I=PEEK(251)+256*PEEK(252)GOTO63
      830                         :rem 212
63775 NEXTI:GOTO63830              :rem 11
63780 PRINT"{LEFT} ":GOTO 63870   :rem 241
63830 PRINT"{LEFT} ":POKE631+PEEK(198),13
                                   :rem 72
63840 PRINT"GO63850":FORA=631TO634:POKEA,
      145:NEXT:POKEA,13:POKE636,13:POKE19
      8,6                         :rem 147
63841 END                         :rem 221
63850 PRINT"{2 UP}":FORA=1TO3:PRINT"
      {8 SPACES}":NEXT:PRINT"{3 UP}";
                                   :rem 28
63860 A=PEEK(253)+256*PEEK(254)+PEEK(2):G
      OTO63750                    :rem 227
63870 Q=PEEK(43):U=PEEK(44)        :rem 29
63880 IFPEEK(Q+4+256*U)<>131GOTO63900
                                   :rem 79
63890 Q1=PEEK(Q+256*U):U1=PEEK(Q+1+256*U)
      :Q=Q1:U=U1:GOTO63880         :rem 86
63900 P=Q+256*U:POKEP,0:POKEP+1,0:rem 173
63910 PRINT"{CLR}{21 I}"          :rem 177
63920 PRINT"{RVS}TYPE LIST TO SEE DATA
      {OFF}"                      :rem 145
63930 PRINT"{21 U}"               :rem 238
```

INSIGHT: Atari

Bill Wilkinson

Learning How

A month or two ago, I stated that I couldn't possibly teach beginning machine language programming in this column—it would consume my entire output for a year or more. And yet I continue to get letters that ask me "How do you learn to write programs?"

I believe that those who ask the question are not asking for a tutorial on the foibles and pitfalls of the FOR-NEXT loop. Nor are they really asking about the intricacies of the 6502 instruction set. Most of them have already mastered the tutorial-level material on their chosen language. What these perplexed people are really asking is "What good is all this programming stuff, anyway?"

And that is not really surprising. So many tutorials tell you *how* to write a program to do such and such. So few discuss *why*. Too often, learning to program is approached like learning a foreign language. Memorize the conjugations and punctuation; put sentences together like this; and if someone asks you "G'dye moya k'neega?" you know what to answer (providing you were studying Russian instead of Spanish).

Computer Conversations

But the need to learn human languages is obvious: The first time you feel hungry in Paris, you can ask for directions to a restaurant in your best Berlitz French. You don't have to "design" a conversation. Not so with learning to program: "Okay, now I know all these neat keywords and syntax and punctuation. How do I start a conversation?" Well, as I hinted above, the secret is that you must *design* a program.

To some, this design process is simple and obvious. Others never really get the hang of it. (Would it surprise you to learn that many professional programmers never become expert at designing? They make their living implementing other people's designs.) And many, like myself, become somewhat proficient at a few kinds of designs while remaining incompetent at others. (My lament: I don't think I will ever achieve the level of creativity necessary to design a really good game.)

Now, all the above philosophizing surely has some purpose, you hope. Indeed, I think it does.

Kibitzing

I have been promising for a few months now that I would provide patches to allow the Atari 1050 drive to work in enhanced mode with good old Atari DOS 2.0s. Well, I finally gathered enough information to begin the task, and I thought you might enjoy looking over my shoulder while I tackle the problem.

This will be a kind of short diary of what I have gone through. There have been more sidetracks and bugs and flat-out boo-boos than I can find room for here. And I won't even tell you how many assemblies I have made (though I will say I made about 10 or 12 just looking for the best of several possibilities for a series of shift instructions).

Even though I admire and strive for a "clean" design, I am apt to take the course of least resistance if I am confident it will work properly. With that in mind, then, let us begin tackling our task.

Note: I will make frequent reference to the listing of Atari DOS 2.0s as published in the book *Inside Atari DOS* from COMPUTE! Books. Page numbers and line numbers in square brackets [131: 1350] refer to the book.

It will *not* be necessary to own the book to understand most of what is going on, but having the book available will make it easier. Also, if you do not understand machine language, neither the book nor my explanations will be easy to follow, but you can still use the results (which will appear next month).

The 1050 And DOS 2.0s

The first thing we must always do is define the task. Here, that is deceptively simple to do: Make the enhanced density mode of the Atari 1050 drive work with Atari DOS 2.0s.

The next step is much harder: Design the implementation of the task. And, actually, this single step consists of many substeps. For example, let's first investigate the facts which I knew when I started.

The drives:

Item: An Atari 810 drive has 40 tracks of 18 sectors of 128 bytes each. That's a total of 720 sectors.

ATARI®

ATARI is a trademark of ATARI, INC

NEW LOWER PRICES!!

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

Memory Expansion for Atari 600XL .. Call

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
Riteman	Call	Okidata 92A	Call

Call for assorted Printer Packages

DISK DRIVES

Percom .... Call For Prices
Trak .... Call
Concorde .... For
Indus GT .... Prices

RS232 Modem Adaptor (Through serial port) 39.95

Koala Touch Tablet Cart or Disk ... 69.95

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

COMMODORE 64

CBM 64	Call
1541 Disk Drive	249.00
1525 80 Column Printer	229.00
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	The Manager-D	67.95
Assembler - D	17.95	General Ledger-	37.95
Easy Finance I,II,III,IV-D	19.95	Accts. Rec.-D	37.95
		Accts. Pay.-D	37.95
Easy Calc-D	64.95	Code Writer-D	37.95
Easy Mail-D	17.95	Zork I, II or III D	29.95
Easy Script-D	39.95	Suspended-D	29.95
Easy Spell-D	19.95	Starcross-D	29.95
Logo-D	39.95	Deadline-D	29.95

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

ATARI	
Conversational Lang. - T	44.95
My First Alphabet - D	26.95
Touch Typing - T	19.95
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
Qix - Cart	32.95
Dig Dug - Cart	32.95
Atariwriter - Cart	74.95
Donkey Kong - Cart	37.95
Ms Pac-Man - Cart	39.95
Tennis - Cart	35.95
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 - C	67.95
Paint - D	33.95
Caverns of Mars - Cart	32.95
Joust - Cart	39.95
Visicalc - D	159.95

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	41.94
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

BRODERBUND	
AE - D	23.95
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

PARKER BROS.	
Astrochase - Cart	34.95
Frogger - Cart	34.95
Q-Bert - Cart	34.95
Popeye - Cart	34.95

EPYX	
Gateway To Apshai - Cart	27.95
Pitstop - Cart	27.95
Swat Rescue - Cart	27.95
Silicon Warrior - 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

DATA SOFT	
Money Wizard - D	49.95
Letter Wizard - D	49.95
Spell Wizard - D	34.95
Genesis - D/T	20.95
Heathcliff - D/T	20.95
Pooyan - D/T	20.95
Zaxxon - D/T	27.95
Micropainter - D	23.95

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
Rhymes & Riddles - D	20.95
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

LJK	
Letter Perfect - D	74.95
Data Perfect - D	74.95
Spell Perfect - D	64.95

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
Chatterbee - D	27.95
Rally Speedway - Cart	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
Star Trek - Cart	27.95
Starbowl Football - D/T	22.95
Hockey - Cart	27.95
Computer War - Cart	27.95
Flip Flop - D/T	20.95
Boulder Dash - D/T	20.95
Carrier Force - D	41.95
Combat Leader - D/T	27.95
Encounter - D/T	23.95
Quasimoto - D/T	23.95
N.Y.C. - D/T	23.95
Compute Books	Call

Hundreds of items available for the Atari and CBM 64 ... please call.

COMMODORE 64 SOFTWARE

ATARISOFT	
Pac-Mac	34.95
Centipede	34.95
Defender	34.95
Dig Dug	34.95
Donkey Kong	34.95
Stargate	34.95
Robotron	34.95

ACCESS	
Neutral Zone - D/T	26.95
Spritemaster - D/T	27.95
Beachhead - D/T	26.95

BLUE SKY	
Calc Result/Easy - Cart	Call
Calc Result/Advanced - D	Call
Script 64 - D	Call

BATTERIES INCLUDED	
Paper Clip	99.95
Delphi Oracle	114.95

DATA 20	
Video Pak 80	139.95
Z80 Video Pack	229.95

HES	
Multiplan - D	74.95
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

CIMMARON	
Instaspeed - D/T	74.95
Instawriter - Cart	55.95
Instamail - D/T	29.95

SIRIUS	
Blackpoole - D	27.95
Repton - D	27.95
Type Attack - D	27.95
Wayout - D	27.95
Critical Mass - D	27.95

SYNAPSE	
Blue Max - D/T	23.95
Zaxxon - D/T	23.95
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

MISCELLANEOUS FOR CBM 64	
Flight Simulator II - D	37.95
Night Mission/Pinball - D	20.95
Praticalc PS - D	59.95
M-File - D	84.95
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	22.95
Zombies - D/T	23.95
Castle Wolfenstein - D	20.95
Swashbuckler - D	23.95
Sorcerer - D	34.95
Computer Baseball - D	27.95
Ringside Seat - D	27.95
Professional Golf - D	27.95
Congo Bongo - Cart	27.95
Mastertype - Cart	27.95
Solo Flight - D	23.95
Vic Switch	124.95
Cardco + G	64.95

ACCESSORIES	
WICO Joystick	Call
Flip'n' File Trunks	20.95
Flip'n' File Cart	20.95
Joysensor	24.95
Elephant Disks (Box of 10)	20.95
WICO Trakball	37.95
KRAFT Joystick	15.95

D - Disk
T - Cassette
Cart - Cartridge

ComputAbility™

To Order Call Toll Free
800-558-0003

For Technical Info, Order Inquiries, or for Wisc. Orders -
414-351-2007

NO SURCHARGE FOR MASTERCARD OR VISA

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

ORDER LINES OPEN
Mon-Fri 11 AM - 7 PM CST
Sat 12 PM - 5 PM CST

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.

Item: An Atari 1050 drive has 40 tracks of 26 sectors of 128 bytes each, for a total of 1040 sectors.

Item: A 1050 will automatically read either density diskette (single or enhanced), but it formats a new diskette according to the format command it receives. In particular, a ! command ($21) causes single-density formatting, while a " command ($22) causes enhanced density.

The software:

Item: DOS 2 is capable of accessing both 810 drives and their double-density equivalents (drives with 40 tracks of 18 sectors of 256 bytes each).

Item: There is an inherent limit of 1024 sectors in DOS 2, since it allows only a 10-bit sector number in the link field of each sector. Also, on a single density diskette, DOS 2 accesses only 719 of the 720 sectors.

Item: The listing of Atari DOS. Actually, this is *not* a "known" item, and much of what follows is a discussion of what I learned and applied from reading the listing several times.

Finding The Format

Armed with these knowns, let's tackle the unknowns. It seemed to me that the first point to attack was the disparity between what the 1050 was capable of and what DOS 2 would request of it. All of a sudden, DOS 2 must be able to understand three different kinds of disk formats. Question: How can DOS tell what format a particular diskette is?

The answer is to be found in the DOS listing [66: 2213–2222]. During initialization, a status request is made of each drive. When the drive responds, one of the bytes it returns to the computer describes the drive's type. In particular, the listing makes it clear that a double-density disk has bit 5 ($20) set on. DOS 2 uses this bit to differentiate between 128-byte and 256-byte sectors.

All very well, even assuming that an enhanced mode 1050 returns a zero bit here (which it does, thus properly indicating 128-byte sectors). But what distinguishes an enhanced density diskette? I confess that I obtained the answer to this question through a simple experiment: I simply booted a system with an Indus 1050-compatible drive as D2 and looked at the status value it returned during DOS initialization. Lo and behold, it returned $80. Not surprisingly, the high bit is off in 810 and double-density modes. Voilà.

Sector Limits

The second major question to investigate is "How many of the 1050's sectors can we make DOS 2 utilize?" Well, we already know that 1024 is an upper limit. Is there any other limiting factor? The answer is in the layout of the Volume Table Of Contents (VTOC) under DOS 2. The VTOC contains a single bit for each accessible sector on the disk (a scheme known as a *bitmap*, though Atari literature often uses *VTOC* and *bitmap* interchangeably). If a bit is on (1), the corresponding sector is available. If a bit is off (0), the sector is in use. With eight bits per byte, then, there must be 90 bytes in the bitmap.

DOS 2 allows only a single sector (in this case, 128 bytes) for the VTOC of each diskette. While we could circumvent this restriction, it would require a lot of work, and might cause some secondary problems. (I don't want to go into this subject more now, but it cost me four to six hours of investigation before I decided against a two-sector VTOC.)

In 128 bytes, there are 1024 bits. So it would seem that the limit on number of sectors is indeed 1024. Alas, it is not to be. The description of the VTOC clearly calls out usages for the first six bytes (DOS type, maximum number of sectors, current number of sectors, write-required flag) and reserves the next four. So now we are down to 118 bytes and 944 sectors. Is that our limit?

A Final Of 976 Sectors

At first, I was inclined to say it is. But I pored over the listing a couple more times, checked every memory reference that was related, and finally concluded that we could use the four reserved bytes. Which gives us 122 bytes and a final maximum of 976 sectors. Well, that doesn't seem too bad. We are only 64 sectors away from the theoretical maximum and surely a lot better off than with a limit of 720 sectors.

So this is our plan: Use the upper bit ($80) of the drive status to recognize an enhanced density diskette; allow 975 sectors (DOS 2 always throws away the first possible sector); displace the bitmap in the VTOC by 4 bytes on the low end and lengthen it to 122 bytes.

Implementing Our Plan

By the time I had decided on a plan, over half the time I had allotted to this project had elapsed. As I write this, all the allotted time is gone, and I am not done yet. Sounds like a typical software project. Anyway, this month I will tell you of the difficulties I faced. Next month we can decide how well I faced them. In any case, let's begin the next step.

Before I could start the actual coding of the modifications, I had to find all the places in DOS which would be affected by my scheme. While many parts of DOS are affected by a change in density (from 128- to 256-byte sectors), there are only a few routines which actually care about such things as disk status, where the VTOC's bitmap is, and how many sectors are available.

Some of the routines I could successfully ignore. For example, when you delete a file and free up its sectors for later use, you must bump the count of free sectors. But if the rest of DOS is working, you don't have to check for validity of the bumped value. The same thing is true when we allocate a free sector and must decrement the count. And the boot process cares whether we are using 128- or 256-byte sectors, but it doesn't care how many sectors are on the disk.

Some Areas Need Patching

But there *are* several spots which definitely need attention, so let's discuss them now (next month we discuss the solutions).

1. In the BSIO (Basic Sector Input Output) routine, there is a check for a format command [65: 2144]. DOS 2 simply compares the current command with $21 (!) and makes a decision according to an exact match. Now, though, we must allow for either $21 or $22 (") as format commands.

2. In DOS initialization [66: 2218], each accessible drive is checked for its status. DOS 2 ignores all bits of the status except bit 5 ($20) and stores a 1 or 2 (single or double density) in the drive table (DRVTBL) for each drive so checked. We need to find a way to capture and use bit 7 ($80), preferably by keeping it in DRVTBL, also. Fortunately, the only other routine which accesses DRVTBL is SETUP, which we discuss below.

3. In XFORMAT [79: 3510], the actual format command is stored in the DCB (for use by BSIO, as above). We need to allow for either $22 or $21, while DOS 2 allows only $21.

4. Also in XFORMAT [79: 3547, 3552], the maximum number of sectors and number of sectors available are stored in the VTOC which is being created (for the newly formatted disk). Currently, DOS 2 simply uses LDA # (load immediate value) to store what it thinks is the only possible count (707). We must provide for the enhanced density count as well.

5. Again in XFORMAT [80: 3559–3570], there are several assumptions made about how big the bitmap is and where the directory and boot sectors are to be represented in the map. Since we will move the base of the map down four bytes, we must provide for variable numbers here, as well.

6. In FRESECT [90: 5166], the base of the bitmap is assumed to be byte 10 ($0A) of the VTOC. We must change the assumption.

7. In GETSECTOR [91: 5199, 5202, 5239], similar assumptions about the bitmap are coded via immediate loads.

8. In SETUP [92: 5288], which is called by every major routine in DOS 2, the type byte stored in DRVTBL (see item 2, above) is simply transferred to a global location (DRVTYP) for use by other routines. If we change what is stored in DRVTBL, we need to change how and what we store in DRVTYP.

Keeping The Patches Small

And that's it. Not too bad, right? If only that were true. Remember, our goal here is to *patch* the standard version of DOS without affecting its normal operations and without requiring a reassembly of the whole thing to make our patches fit. In general, then, the smaller and fewer the patches, the better.

The real problem here is the number of load immediate instructions, used to implement what are now to become invalid assumptions. If these were three-byte instructions (such as loads from a non-zero page memory location), we would have a simple task: Change the values in the locations being loaded.

Since they are load immediate instructions, though, our only choices are to either make large and cumbersome patches (generally JSRs to subroutines which will do the work, but remember that JSR occupies three bytes), use loads from zero page (a neat alternative, but we have no zero page available to us), or to continue to use load immediate.

Self-Modifying Routines

My choice? Continue to use load immediate. But how? By producing some (shudder at this next phrase, please) self-modifying routines. Remember how I said at the beginning that I sometimes took the path of least resistance? This is one of those sometimes.

The "trick" which allows my scheme to work is relatively simple: Every routine which needs a load immediate changed is only used by DOS 2 after a call has been made to SETUP. Basically, SETUP examines the disk number and drive type and produces various pointers and values in fixed locations for use by other, higher-level routines. What would be more appropriate than for SETUP to also set up the needed values which will be loaded in immediate mode?

And this is, indeed, the plan I tried. At the point where SETUP stores the drive type [92: 5288], I placed a JSR to my patch-it routine. And my patch-it routine used the disk type information to determine which of a pair of immediate values would be used in each of the cases noted above. It looked like it would work.

Fitting The Patch Into DOS.SYS

Except (You knew that was coming, didn't you?) where do I put the patch? I have discussed this subject before, so let me succinctly say that the only sizable patch area in DOS.SYS is at location

$1501, in the gap between DOS.SYS and Mini-DUP (the root of DUP.SYS). There are exactly 63 bytes available there. And my routine was about 85 bytes long.

The story of how I pared my patch down to fit (just barely) will have to wait for next month. Fortunately, it is a short patch. Also fortunately, there are a couple of small patch spaces still floating around in DOS.

Incidentally, if you were looking for the continuation of my notes on how to load saved binary files, keep looking. It turns out that the subject has direct bearing on what we are doing here, so it seemed not inappropriate to postpone it a month (or possibly two). ©

A BASIC Cross-Reference

Jim Butterfield, Associate Editor

"Cross-Ref" is a valuable programming tool that serves several purposes. Not only does it locate all line number and variable references in a program, but it also helps you prepare documentation and even tighten up your program. It's for BASIC programs stored on disk and will output to the screen or printer. For PET/CBM (Upgrade and 4.0 BASIC) and Commodore 64.

"Cross-Ref" and "Cross-Ref64" will analyze a BASIC program stored on disk and give you information on all line number references and all variable references.

It works only with programs written in BASIC; it does not work with programs stored on tape. A program SAVEd on disk may be manipulated as if it were a data file; but a program on tape cannot be handled in that way.

All types of variables are detected and listed: regular variables, strings, integer variables, and arrays. This includes special variables such as TI, TI$, or ST. If a variable name contains more than two characters, only the first two will be shown. (They're the only ones used by BASIC.) So HOUSE is the same variable as HONK.

While Everything Is Fresh In Your Mind

If you have completed writing a program, the Cross-Ref output will serve as a valuable piece of documentation. As each line and variable is listed, you may note its purpose while everything is fresh in your mind: "Line 300 is the start of the analysis: variable A$ is the name of the input file...."

Even if your program is not complete, Cross-Ref can be useful. In large programs, you may wonder what variable names have been used; you want to pick a fresh variable name that won't conflict with anything else. Alternatively, a test run may reveal a problem that shows up within the subroutine that starts at line 750: You can find all calls to that subroutine.

If you're thinking of tightening up your program, you may want to pack two or three lines of code together into a single line. But you can't do this if some of the lines are referenced elsewhere in the program. Cross-Ref will tell you the story.

And if you're looking at somebody else's program, and don't know, say, what variable V3 is being used for, you can run Cross-Ref and find every occurrence of V3.

Running The Program

LOAD and RUN Cross-Ref. Be sure you place the disk with the program you want to cross-reference into the disk drive.

When Cross-Ref asks PROGRAM?, type in the name of the program you wish to analyze. You may use pattern matching if you wish: For example, BAG* will match program name BAGELS.

Everything happens very fast. The disk runs for about the same amount of time that is needed to load the program in question. Then you are asked PRINTER? At that time, the cross-reference is complete; the program wants to know where to deliver the results. Answer Y or N.

Output may be to screen or printer. The line number cross-reference appears first. The referenced line number appears, followed by a colon, then the lines where it is used.

Then the variable cross-reference appears, in alphabetical order. Arrays are shown with a single left parenthesis, so that A(M+NV%) will be shown as A(—and there will also be other entries for M and NV%, of course.

Sometimes a variable or line number will be used more than once on a single line of your program, for example, "100 X=X+7:IF X>20 THEN X=0". In this case, the cross-reference for X will show line 100 only once.

Machine Language For Speed

It's written mostly in machine language for speed. An early BASIC version of this program appeared in COMPUTE!, May/June 1980 (that's Issue 4); being a BASIC program, it ran slo-o-o-owly. But it worked on identical principles to this version of Cross-Ref.

If you're interested in the mechanics, the next few paragraphs give an insight into the unusual logic of both the original BASIC version and the machine language program presented here.

Because of the plethora of characters to be analyzed, an unusual approach was taken. It might be called a "state transition" program.

Here's the general idea. When we begin the analysis of a BASIC line, we start in state A. In this state, we are interested in only a few characters: an alphabetic, which signals the start of a variable; a GOTO, THEN, or GOSUB, which signals that a line number may be coming; a REM, which indicates we should ignore everything up to the end of the line; quote marks, which tell us that the next few characters will not be of interest to us; and binary zero, which signals end of line.

If we don't see any of these characters, we remain in state A and get the next character, throwing the old one away. But if we do see a character of interest, we switch to a new state.

Suppose we're looking at a line that says:

```
FOR J=1 TO 9:X35$="HELLO":GOTO 500
```

We start in state A. The first thing we get is the FOR—it's not a character, but a specially coded *token*. Throw it away; it's not on our list. Continuing on our line, we see a space, which we trash, followed by the letter J. Aha! It's an alphabetic, which tells us "we're in a variable—start collecting characters." At this point we don't know if the variable is called J, J5, JEEPERS, or JR$. We collect the J and switch to state B.

In state B, we are looking for a whole different set of characters. Alphabetic and numeric characters will be collected into our variable name and will move us to state C. On the other hand, a dollar or percent sign will also be collected, but will move us to state E, where we look for a possible array. Continuing the options: a left parenthesis would signal an array; collect it and wrap up this label. A space will be ignored. Almost anything else (in our example, the equals token) will cause the label to be wrapped up and put away, returning us to state A.

Back in state A again, we throw away the equals, the 1 character, the space, the TO token, the 9, and the colon. Suddenly we hit the X: Collect it, and we're off to state B again. This time, state C finds a numeric, collects it, and switches us to state D. State D throws away the 5. We stay in state D and discover the dollar sign, which is duly collected, and we flip to state E. The equals sign drops us back to state A; but we wrap up the collected characters X3$ and enter them into the results table. And so on. Each individual state searches for its own set of characters which trigger an action and a movement to another state.

The program to do all this is surprisingly small. The state transition table that directs the program from one state to another is surprisingly big.

There are tricky bits, some of which involve the strange syntax of the PRINT statement. It's possible to write BASIC lines such as:

```
PRINT A$B$C%D(3)E
```

I'd much rather use semicolons to separate those variables, but since we're allowed to code that way, extra programming must be added to Cross-Ref to pick out the variables when they are mushed together like that.

Typing Cross Reference

Both the PET/CBM and 64 versions of this program use a special technique to attach the machine language to the BASIC portion of the program. The ML is located immediately following the end of the BASIC program, then the zero-page pointer to the end of the program is changed to point to the end of the ML. This fools the computer into treating the ML as part of the BASIC program.

To enter the PET/CBM version, first type in Program 1. You must enter it *exactly* as it is shown because the ML must begin at exactly the end of BASIC. You can check by typing the following line in direct mode:

```
PRINT PEEK(1261),PEEK(1262),PEEK(1263)
```

If you have entered Program 1 correctly, you'll see:

```
58    160    52
```

If these are not the values you get, check for spaces added or left out. When you have Program 1 entered correctly, type the following line in direct mode:

```
POKE 41,10:POKE 2560,0:NEW
```

Then type in and RUN Program 2. Program 2 will check for DATA statement errors as it POKEs the ML into the proper locations. If no errors are detected, the program will change the pointers in zero page to attach the ML to the BASIC from Program 1. When you type LIST after Program 2 is finished, you should see the lines from Program 1. Although it doesn't show, the ML POKEd by Program 2 is also in place. You should immediately SAVE a copy of the completed Cross-Ref program. You will not need the old Program 1 or 2 again.

The 64 Version

To enter the 64 version (Program 3), you *must* use the MLX machine language editor. If you have not already typed in MLX from a previous issue of COMPUTE!, there's a copy elsewhere in this issue. Be sure you read the accompanying article and understand how to use MLX before you begin typing in the data from Program 3. The MLX listing in Program 3 contains the BASIC as well as the ML portions of Cross-Ref, so no separate BASIC program must be typed in. MLX makes things *much* easier—it's a program worth SAVEing for this, and future, programs.

Because Cross-Ref begins at the default start-of-BASIC address (where MLX would normally be located), you must adjust the 64 so that the BASIC area for MLX is above the area of memory which Cross-Ref will occupy. Do this by typing the following line in direct mode (no line number):

POKE 44,16:POKE 642,16:POKE 4096,0:NEW

If you do not finish typing all of Program 3 in one session, see the instructions in the MLX article on saving an unfinished version of your work. Note that you must also type the direct mode line above *before* loading MLX again to continue your work.

When MLX is first RUN, it will ask you for a starting and ending address. For Cross-Ref, the proper values are:

starting address	**2049**
ending address	**3398**

Use the MLX Save option to make a copy of your work. The version of Cross-Ref created by MLX can then be LOADed and RUN like a regular BASIC program.

An early version of Cross-Ref for PET/CBM, called XREF, was published in *Cursor* magazine (which came on cassette tape), issue 25. The details are different, but the program's general speed and other characteristics are about the same.

Could Cross-Ref be expanded to analyze other features? For example, FOR/NEXT loop matches or OPEN and CLOSE statements together with associated file usage? Perhaps, but I think not. Whether or not it's a good idea, BASIC allows a single FOR statement to be matched with more than one NEXT (and vice versa, for that matter). Files can be opened, closed and used with variable logical file numbers—for example, PRINT#X, "HELLO"—so that a single file's activity is difficult to trace. Cross-Ref wasn't constructed to follow the logic of your program, only the mechanics. You should find Cross-Ref a very useful programming support tool. You might discover that it leads to better programming.

The programs are set up for normal Commodore printers. If you have a printer that specifically needs a line feed character to be sent, you should modify Cross-Ref64 only as follows:

POKE 3181,10
POKE 3223,10

Program 1: BASIC Portion Of PET/CBM Version

```
100 PRINT"{CLR}CROSS REF":PRINT"
    {SHIFT-SPACE}{4 SPACES}JIM BUTTERFIEL
    D"
115 W=6:IFPEEK(32808)=32THENW=11
120 CLOSE1:INPUT"NAME OF PROGRAM";N$
130 OPEN1,8,3,N$+",P,R":GET#1,X$,Y$:IFX$<
    >CHR$(1)GOTO120
190 SYS1668:CLOSE1:INPUT"PRINTER";Z$:P=3:
    IFASC(Z$)=89THENP=4:W=11
200 OPEN4,P:PRINT#4,"CROSS-REF: ";N$:POKE
    208,W:SYS2102:PRINT#4:CLOSE4
```

Program 2: Loader For PET/CBM ML Portion

```
100 SA=1267:SL=200
110 FOR I=0 TO 8
120 CK=0:AD=SA+(I*120):LN=SL+(I*150)
130 FOR J=0 TO 119
140 READ BY:CK=CK+BY:POKE AD+J,BY
150 NEXT J:READ CV:IF CK<>CV THEN 190
160 NEXT I:PRINT "MACHINE LANGUAGE IS LOA
    DED"
170 POKE 40,1:POKE 41,4:POKE 42,43:POKE 4
    3,9
180 POKE 44,43:POKE 45,9:POKE 46,43:POKE
    {SPACE}47,9:END
190 PRINT "DATA ERROR IN LINES";LN;"-";LN
    +140:STOP
200 DATA 0,0,0,0,0,0,0,0
210 DATA 0,0,0,0,0,0,11,11
220 DATA 11,11,11,11,11,11,11,11
230 DATA 11,11,11,11,11,11,11,11
240 DATA 11,11,11,11,11,11,11,11
250 DATA 11,11,11,11,11,11,11,5
260 DATA 11,3,3,3,11,4,11,11
270 DATA 11,9,11,11,11,2,2,2
280 DATA 2,2,2,2,2,2,2,8
290 DATA 11,11,11,11,11,11,1,1
300 DATA 1,1,1,1,1,1,1,1
310 DATA 1,1,1,1,1,1,1,1
320 DATA 1,1,1,1,1,1,1,1
330 DATA 11,11,11,11,11,11,11,11
340 DATA 11,11,11,11,11,11,11,11
345 DATA 774
350 DATA 11,11,11,11,11,11,11,11
360 DATA 11,11,11,11,11,11,11,11
370 DATA 11,11,11,11,11,11,11,11
380 DATA 7,11,11,11,11,11,10,10
390 DATA 11,11,10,11,6,11,11,11
400 DATA 11,11,11,11,11,11,11,11
410 DATA 11,11,11,11,11,11,11,11
420 DATA 11,9,11,11,10,11,11,11
430 DATA 11,11,11,11,11,11,11,11
440 DATA 11,11,11,11,11,11,11,11
450 DATA 11,11,11,11,11,11,11,11
460 DATA 11,11,11,11,11,11,11,11
470 DATA 10,11,11,11,11,11,11,11
480 DATA 11,11,11,11,11,11,11,11
490 DATA 11,11,11,11,11,11,11,11
495 DATA 1304
500 DATA 11,11,11,11,11,11,11,11
510 DATA 11,11,11,11,11,11,11,11
520 DATA 11,11,11,11,11,11,11,11
530 DATA 11,11,11,11,11,0,12,12
540 DATA 12,12,12,12,12,12,12,12
550 DATA 12,0,224,72,12,12,24,36
560 DATA 48,12,12,60,12,0,24,24
570 DATA 24,24,12,24,24,24,24,24
580 DATA 24,0,36,36,36,36,36,36
590 DATA 36,36,36,36,36,0,48,48
600 DATA 48,48,48,48,48,12,48,48
610 DATA 48,0,224,212,12,12,24,36
620 DATA 48,12,60,60,12,0,72,72
630 DATA 12,12,24,36,48,12,12,60
640 DATA 12,0,12,212,12,12,24,36
645 DATA 3507
650 DATA 48,12,60,60,12,0,236,236
660 DATA 248,140,24,36,48,12,12,60
670 DATA 12,0,108,108,248,140,24,36
```

```
680 DATA 48,12,12,60,12,0,120,12
690 DATA 12,140,24,36,48,12,12,60
700 DATA 12,162,1,32,198,255,32,54
710 DATA 7,169,0,133,190,169,11,133
720 DATA 191,169,6,133,185,162,13,189
730 DATA 29,9,157,249,10,202,16,247
740 DATA 48,7,32,204,255,96,32,179
750 DATA 7,32,228,255,32,228,255,240
760 DATA 241,169,0,133,192,169,10,133
770 DATA 193,32,228,255,133,90,32,228
780 DATA 255,133,89,162,12,134,184,32
790 DATA 228,255,201,32,240,249,170,189
795 DATA 12998
800 DATA 0,5,168,177,184,16,3,32
810 DATA 11,7,41,127,164,184,133,184
820 DATA 201,84,176,7,192,84,144,3
830 DATA 32,64,7,201,120,208,19,192
840 DATA 120,208,15,142,122,2,32,64
850 DATA 7,174,122,2,169,12,133,184
860 DATA 208,205,201,0,240,160,208,191
870 DATA 41,127,72,201,84,240,20,138
880 DATA 162,0,180,84,192,32,240,7
890 DATA 232,224,5,208,245,240,18,149
900 DATA 84,240,14,138,162,0,180,85
910 DATA 148,84,232,224,4,208,247,133
920 DATA 88,104,96,162,4,169,32,149
930 DATA 84,202,16,251,96,72,165,192
940 DATA 164,193,56,233,7,133,186,176
945 DATA 14445
950 DATA 1,136,132,187,201,0,152,233
960 DATA 10,144,20,160,4,185,84,0
970 DATA 209,186,208,5,136,16,246,48
980 DATA 73,165,186,164,187,208,219,165
990 DATA 192,164,193,133,188,132,189,56
1000 DATA 233,7,176,1,136,133,186,132
1010 DATA 187,201,0,152,233,10,144,21
1020 DATA 160,6,56,177,186,145,188,249
1030 DATA 84,0,136,16,246,144,6,165
1040 DATA 186,164,187,208,214,160,6,185
1050 DATA 84,0,145,188,136,16,248,24
1060 DATA 165,192,105,7,133,192,144,2
1070 DATA 230,193,32,54,7,104,96,96
1080 DATA 165,190,164,191,133,186,132,187
1090 DATA 56,165,192,233,0,141,122,2
1095 DATA 15395
1100 DATA 165,193,233,10,141,123,2,13
1110 DATA 122,2,240,227,24,173,122,2
1120 DATA 101,186,133,190,133,188,173,123
1130 DATA 2,101,187,133,191,133,189,32
1140 DATA 39,8,165,192,56,233,7,164
1150 DATA 193,176,1,136,133,192,132,193
1160 DATA 201,0,152,233,10,144,184,165
1170 DATA 188,164,189,56,233,7,176,1
1180 DATA 136,133,188,132,189,160,6,56
1190 DATA 177,186,145,188,241,192,136,16
1200 DATA 247,144,6,32,39,8,76,250
1210 DATA 7,160,6,177,192,145,188,136
1220 DATA 16,249,48,190,165,186,164,187
1230 DATA 56,233,7,176,1,136,133,186
1240 DATA 132,187,96,162,4,134,84,32
1245 DATA 15168
1250 DATA 201,255,169,0,160,11,133,186
1260 DATA 132,187,160,4,185,84,0,209
1270 DATA 186,208,5,136,16,246,48,34
1280 DATA 169,13,32,210,255,169,10,32
1290 DATA 210,255,160,0,177,186,153,84
1300 DATA 0,32,210,255,200,192,5,144
1310 DATA 243,169,58,32,210,255,169,0
1320 DATA 133,188,230,188,165,188,197,208
1330 DATA 144,22,169,13,32,210,255,169
1340 DATA 10,32,210,255,160,5,169,32
1350 DATA 32,210,255,136,16,248,48,222
1360 DATA 160,5,177,186,133,90,200,177
1370 DATA 186,133,89,32,225,255,164,151
1380 DATA 200,208,248,32,192,8,24,165
1390 DATA 186,164,187,105,7,144,1,200
1395 DATA 16229
1400 DATA 133,186,132,187,197,190,165,187
1410 DATA 229,191,144,134,96,169,0,162
1420 DATA 2,157,122,2,202,16,250,120
1430 DATA 248,160,15,6,89,38,90,162
1440 DATA 2,189,122,2,125,122,2,157
1450 DATA 122,2,202,16,244,136,16,235
1460 DATA 216,88,162,0,169,48,133,189
1470 DATA 134,192,189,122,2,72,74,74
1480 DATA 74,74,9,48,32,16,9,104
1490 DATA 41,15,9,48,224,2,208,2
1500 DATA 198,189,32,16,9,166,192,232
1510 DATA 224,3,144,220,96,197,189,208
1520 DATA 4,169,32,208,2,198,189,76
1530 DATA 210,255,0,0,0,0,0,0
1540 DATA 0,78,79,78,69,32,0,0
1545 DATA 12648
```

Program 3: MLX Listing For 64

```
2049 :043,008,100,000,153,034,083
2055 :032,067,082,079,083,083,177
2061 :032,082,069,070,034,058,102
2067 :153,034,160,032,032,032,206
2073 :032,074,073,077,032,066,123
2079 :085,084,084,069,082,070,249
2085 :073,069,076,068,034,000,101
2091 :052,008,115,000,087,178,227
2097 :048,054,000,081,008,120,104
2103 :000,160,049,058,133,034,233
2109 :078,065,077,069,032,079,205
2115 :070,032,080,082,079,071,225
2121 :082,065,077,034,059,078,212
2127 :036,000,126,008,130,000,123
2133 :159,049,044,056,044,051,232
2139 :044,078,036,170,034,044,241
2145 :080,044,082,034,058,161,044
2151 :035,049,044,088,036,044,143
2157 :089,036,058,139,088,036,043
2163 :179,177,199,040,049,041,032
2169 :137,049,050,048,000,176,069
2175 :008,190,000,158,050,054,075
2181 :057,050,058,160,049,058,053
2187 :133,034,080,082,073,078,107
2193 :084,069,082,034,059,090,051
2199 :036,058,080,178,051,058,100
2205 :139,198,040,090,036,041,189
2211 :178,056,057,167,080,178,111
2217 :052,058,087,178,049,049,130
2223 :000,224,008,200,000,159,254
2229 :052,044,080,058,152,052,107
2235 :044,034,067,082,079,083,064
2241 :083,045,082,069,070,058,088
2247 :032,034,059,078,036,058,240
2253 :151,049,057,048,044,087,129
2259 :058,158,051,049,051,054,120
2265 :058,152,052,058,160,052,237
2271 :000,000,000,000,000,000,223
2277 :000,000,000,000,000,000,229
2283 :000,000,000,000,000,000,235
2289 :000,000,000,000,000,000,241
2295 :000,000,000,000,000,000,247
2301 :000,000,000,000,011,011,019
2307 :011,011,011,011,011,011,069
2313 :011,011,011,011,011,011,075
2319 :011,011,011,011,011,011,081
2325 :011,011,011,011,011,011,087
```

```
2331 :011,011,011,011,011,011,093
2337 :011,005,011,003,003,003,069
2343 :011,004,011,011,011,009,096
2349 :011,011,011,002,002,002,084
2355 :002,002,002,002,002,002,063
2361 :002,008,011,011,011,011,111
2367 :011,011,001,001,001,001,089
2373 :001,001,001,001,001,001,075
2379 :001,001,001,001,001,001,081
2385 :001,001,001,001,001,001,087
2391 :001,001,001,001,011,011,113
2397 :011,011,011,011,011,011,159
2403 :011,011,011,011,011,011,165
2409 :011,011,011,011,011,011,171
2415 :011,011,011,011,011,011,177
2421 :011,011,011,011,011,011,183
2427 :011,011,011,011,011,011,189
2433 :011,011,007,011,011,011,191
2439 :011,011,010,010,011,011,199
2445 :010,011,006,011,011,011,201
2451 :011,011,011,011,011,011,213
2457 :011,011,011,011,011,011,219
2463 :011,011,011,011,011,009,223
2469 :011,011,010,011,011,011,230
2475 :011,011,011,011,011,011,237
2481 :011,011,011,011,011,011,243
2487 :011,011,011,011,011,011,249
2493 :011,011,011,011,011,011,255
2499 :011,011,011,011,011,011,005
2505 :011,011,010,011,011,011,010
2511 :011,011,011,011,011,011,017
2517 :011,011,011,011,011,011,023
2523 :011,011,011,011,011,011,029
2529 :011,011,011,011,011,011,035
2535 :011,011,011,011,011,011,041
2541 :011,011,011,011,011,011,047
2547 :011,011,011,011,011,011,053
2553 :011,011,011,011,011,011,059
2559 :011,000,012,012,012,012,058
2565 :012,012,012,012,012,012,077
2571 :012,000,224,072,012,012,087
2577 :024,036,048,012,012,060,209
2583 :012,000,024,024,024,024,131
2589 :012,024,024,024,024,024,161
2595 :024,000,036,036,036,036,203
2601 :036,036,036,036,036,036,001
2607 :036,000,048,048,048,048,019
2613 :048,048,048,012,048,048,049
2619 :048,000,224,212,012,012,055
2625 :024,036,048,012,060,060,049
2631 :012,000,072,072,012,012,251
2637 :024,036,048,012,012,060,013
2643 :012,000,012,212,012,012,087
2649 :024,036,048,012,060,060,073
2655 :012,000,236,236,248,140,199
2661 :024,036,048,012,012,060,037
2667 :012,000,108,108,248,140,211
2673 :024,036,048,012,012,060,049
2679 :012,000,120,012,012,140,159
2685 :024,036,048,012,012,060,061
2691 :012,162,001,032,198,255,023
2697 :032,054,011,169,000,133,024
2703 :075,169,015,133,076,169,012
2709 :010,133,070,162,013,189,214
2715 :052,013,157,249,014,202,074
2721 :016,247,048,007,032,204,203
2727 :255,096,032,179,011,032,004
2733 :228,255,032,228,255,240,131
2739 :241,169,000,133,077,169,200
2745 :014,133,078,032,228,255,157
2751 :133,093,032,228,255,133,041
2757 :092,162,012,134,069,032,186
2763 :228,255,201,032,240,249,128
2769 :170,189,000,009,168,177,154
2775 :069,016,003,032,011,011,101
2781 :041,127,164,069,133,069,056
2787 :201,084,176,007,192,084,203
2793 :144,003,032,064,011,201,176
2799 :120,208,019,192,120,208,082
2805 :015,142,060,003,032,064,049
2811 :011,174,060,003,169,012,168
2817 :133,069,208,205,201,000,049
2823 :240,160,208,191,041,127,206
2829 :072,201,084,240,020,138,000
2835 :162,000,180,087,192,032,160
2841 :240,007,232,224,005,208,173
2847 :245,240,018,149,087,240,242
2853 :014,138,162,000,180,088,107
2859 :148,087,232,224,004,208,178
2865 :247,133,091,104,096,162,114
2871 :004,169,032,149,087,202,186
2877 :016,251,096,072,165,077,226
2883 :164,078,056,233,007,133,226
2889 :071,176,001,136,132,072,149
2895 :201,000,152,233,014,144,055
2901 :020,160,004,185,087,000,029
2907 :209,071,208,005,136,016,224
2913 :246,048,073,165,071,164,096
2919 :072,208,219,165,077,164,240
2925 :078,133,073,132,074,056,143
2931 :233,007,176,001,136,133,033
2937 :071,132,072,201,000,152,237
2943 :233,014,144,021,160,006,193
2949 :056,177,071,145,073,249,136
2955 :087,000,136,016,246,144,000
2961 :006,165,071,164,072,208,063
2967 :214,160,006,185,087,000,035
2973 :145,073,136,016,248,024,031
2979 :165,077,105,007,133,077,215
2985 :144,002,230,078,032,054,197
2991 :011,104,096,096,165,075,210
2997 :164,076,133,071,132,072,061
3003 :056,165,077,233,000,141,091
3009 :060,003,165,078,233,014,234
3015 :141,061,003,013,060,003,224
3021 :240,227,024,173,060,003,164
3027 :101,071,133,075,133,073,029
3033 :173,061,003,101,072,133,248
3039 :076,133,074,032,039,012,077
3045 :165,077,056,233,007,164,163
3051 :078,176,001,136,133,077,068
3057 :132,078,201,000,152,233,013
3063 :014,144,184,165,073,164,223
3069 :074,056,233,007,176,001,032
3075 :136,133,073,132,074,160,199
3081 :006,056,177,071,145,073,025
3087 :241,077,136,016,247,144,108
3093 :006,032,039,012,076,250,180
3099 :011,160,006,177,077,145,091
3105 :073,136,016,249,048,190,233
3111 :165,071,164,072,056,233,032
3117 :007,176,001,136,133,071,057
3123 :132,072,162,009,181,069,164
3129 :157,080,003,202,016,248,251
3135 :096,162,009,189,080,003,090
3141 :149,069,202,016,248,162,147
3147 :004,134,087,032,201,255,020
3153 :169,000,160,015,133,071,117
3159 :132,072,160,004,185,087,215
3165 :000,209,071,208,005,136,210
3171 :016,246,048,034,169,013,113
3177 :032,210,255,169,032,032,067
3183 :210,255,160,000,177,071,216
```

```
3189 :153,087,000,032,210,255,086
3195 :200,192,005,144,243,169,052
3201 :058,032,210,255,169,000,085
3207 :133,073,230,073,165,073,114
3213 :197,190,144,022,169,013,108
3219 :032,210,255,169,032,032,109
3225 :210,255,160,005,169,032,216
3231 :032,210,255,136,016,248,032
3237 :048,222,160,005,177,071,080
3243 :133,093,200,177,071,133,210
3249 :092,032,225,255,240,031,028
3255 :165,203,010,010,144,245,192
3261 :032,215,012,024,165,071,196
3267 :164,072,105,007,144,001,176
3273 :200,133,071,132,072,197,238
3279 :075,165,072,229,076,144,200
3285 :131,096,169,000,162,002,005
3291 :157,060,003,202,016,250,139
3297 :120,248,160,015,006,092,098
3303 :038,093,162,002,189,060,007
3309 :003,125,060,003,157,060,133
3315 :003,202,016,244,136,016,092
3321 :235,216,088,162,000,169,095
3327 :048,133,074,134,077,189,142
3333 :060,003,072,074,074,074,106
3339 :074,009,048,032,039,013,226
3345 :104,041,015,009,048,224,202
3351 :002,208,002,198,074,032,027
3357 :039,013,166,077,232,224,012
3363 :003,144,220,096,197,074,001
3369 :208,004,169,032,208,002,152
3375 :198,074,076,210,255,000,092
3381 :000,000,000,000,000,000,053
3387 :078,079,078,069,032,000,139
3393 :000,013,013,013,013,013,130
```

©

C. Regena

File Processing
Part 3

This month C. Regena concludes her three-part discussion on creating data files.

A Birthday List

Program 1 prints a birthday list of the students in a class. The same data file is used, and the information is arranged in order by birthdate. Line 180 is the OPEN statement for the printer (use your own printer configuration). Line 190 is the OPEN statement for the disk drive to read in information.

Line 210 reads in the date—again, in the same order that the items were saved. We will ignore some of the information, but all the items must be read in order. Line 250 combines several of the items into one variable T$. The birthday BD and T$ are actually arrays, so the items may be sorted. Lines 280–350 contain the sorting procedure to sort by birthday.

Line 360 and lines 510–560 print the header. Lines 370–480 print the information. Lines 380–400 print the month and day from the BD number that was saved. Line 410 prints a blank line between months. Lines 420–450 use POS and SEG$ to separate the T$ item back into its parts, then line 460 prints the information in columns using the IMAGE statement of line 200.

The Report Writer

Program 2 generates reports using the data saved in Program 1 of Part 2 (April 1984). Lines 160–200 present the option to print the report for one of the reading groups or for the whole class.

These reports will use a 132-column line, or *compressed print* (16.5 characters per inch). Line 210 OPENs device #1 for the printer. The previous reports used an 80-column line, which is the default value for most printers. VARIABLE 132 is used to designate a longer line before a carriage return. Line 230 sets *my* printer (TI 825, which is like the TI 840) to use compressed print. You will probably need a different command.

Some printers can use a certain CHR$ number. Other printers may require you to set certain hardware switches. I have used compressed print and the 132-column line so more can fit on the one line. The other two reports in this program may be printed with the regular printing.

Line 240 is the OPEN statement to read the data from the data file created by Program 1 (Part 2, April). Again, the variables are in the same order as they were saved. Line 280 checks for the end of the file. Lines 290–300 check to see if a particular group was chosen or if the whole class is to be printed. Lines 310–480 then print the first report. The student's R$ tally is separated using SEG$. Line 360 and line 410 are used to print information if only part of the ten weeks is used. If you have a different number of weeks in your report, you can change the 10 in lines 130, 410, 520, 560, 600, and 670, and the titles in lines 140 and 930–950.

Total Values

Variable names starting with T are total values. Lines 440–450 print total presentations divided by total possible weeks and the individual's percentage. Lines 500–630 print the totals for each week.

A bar graph report is printed in lines 640–700. Each asterisk represents a report, and the appropriate number of asterisks is printed for each week as a graph.

The final report in this program is to rank the students from high score to low score by percentage. Lines 720–780 contain the sort routine. The percentages were stored in the P array with the corresponding names in NN$. Lines 790–850 print the percents and names. Line 810 and the subroutine in lines 1000–1150 alphabetize the names of all students who have a zero score.

Console BASIC

You can, in fact, do file processing without Extended BASIC and all the peripherals. I used Extended BASIC mainly because of the ease in formatting the printing—lining up the columns. In regular console BASIC you can use subroutines to line up columns of numbers and the TAB function to start the columns right. See my January 1984 column in COMPUTE! for some suggestions on formatting and screen scrolling.

To use a printer you need the RS-232 interface plus the printer. A number of different name brands of printers can be used with the TI-99/4A. The printer manuals should tell you what features the printer has and how to control different features, such as the number of characters per inch and form feeds. Using the printer and RS-232 manuals, you can determine the appropriate printer configuration necessary for the OPEN statement. Without a printer, you can print on the screen—just keep within the 28 print columns and print a screen at a time or use a scrolling delay method so you can read the information as it is printed.

To use a disk drive you also need the disk controller or disk controller card for the Peripheral Expansion box. The disk controller or card comes with a command module and a manual that describes disk procedures. To use a cassette, simply change the "DSK1.---" statements to "CS1" and change the VARIABLE to FIXED. The cassette system works fine—it just takes longer than the disk system.

Program 1: Birthday List

```
80 REM TI EXTENDED BASIC
90 REM DISK, PRINTER
100 REM BIRTHDAY LIST
110 CALL CLEAR
120 DISPLAY AT(12,5):"BIRTHDAY LIST
    "
130 OPTION BASE 1
140 DIM T$(140),BD(140),M$(12)
150 FOR I=1 TO 12 :: READ M$(I):: N
    EXT I
160 DATA JAN,FEB,MAR,APR,MAY,JUN,JU
    L,AUG,SEP,OCT,NOV,DEC
170 L=0 :: I=1 :: L$="---"
180 OPEN #1:"RS232.BA=600"
190 OPEN #3:"DSK1.SAMPLE",INTERNAL,
    INPUT ,VARIABLE 192
200 IMAGE "{5 SPACES}### ##
    {3 SPACES}#####################
    ##  ########{3 SPACES}#########
    ################"
210 INPUT #3:G,N$,F$,A$,P$,BD(I),R$
    ,C$
220 IF C$="MOVED" THEN 210
230 IF N$="ZZZ" THEN 270
240 IF P$="" THEN P$="{4 SPACES}"
250 T$(I)=F$&" "&N$&"/"&P$&A$
260 I=I+1 :: GOTO 210
270 I=I-1 :: CLOSE #3
280 DISPLAY AT(23,1):"SORTING"
290 B=1
300 B=2*B :: IF B<=I THEN 300
310 B=INT(B/2):: IF B=0 THEN 360
320 FOR J=1 TO I-B :: C=J
330 D=C+B :: IF BD(C)<=BD(D)THEN 35
    0
340 AA=BD(C):: TT$=T$(C):: BD(C)=BD
    (D):: T$(C)=T$(D):: BD(D)=AA ::
     T$(D)=TT$ :: C=C-B :: IF C>0 T
    HEN 330
350 NEXT J :: GOTO 310
360 GOSUB 510
370 FOR J=1 TO I
380 IF BD(J)=0 THEN B$="---" :: D=0
     :: GOTO 420
390 BD$=STR$(BD(J)):: M=VAL(SEG$(BD
    $,1,LEN(BD$)-2)):: D=VAL(SEG$(B
    D$,LEN(BD$)-1,2))
400 B$=M$(M):: IF B$=L$ THEN 420
410 L=L+1 :: PRINT #1 :: L$=B$
420 P=POS(T$(J),"/",8)
430 N$=SEG$(T$(J),1,P-1)
440 P$="586-"&SEG$(T$(J),P+1,4)
450 A$=SEG$(T$(J),P+5,LEN(T$(J))-P+
    4)
460 PRINT #1,USING 200:B$,D,N$,P$,A
    $
470 L=L+1 :: IF L=48 THEN PRINT #1:
    CHR$(12):: L=0 :: GOSUB 510
480 NEXT J
490 PRINT #1:CHR$(12)
500 STOP
510 PRINT #1:TAB(34);"SAMPLE CLASS"
520 PRINT #1: :TAB(34);"BIRTHDAY LI
    ST"
530 PRINT #1: :TAB(33);"APRIL 15, 1
    984"
540 PRINT #1: : :TAB(5);"BIRTHDAY";
    TAB(15);"NAME";TAB(41);"PHONE";
    TAB(54);"ADDRESS"
550 PRINT #1:TAB(5);"--------";TAB(
    15);"----";TAB(41);"-----";TAB(
    54);"-------": : :
560 RETURN
570 END
```

Program 2: Report Writer

```
80 REM TI EXTENDED BASIC
90 REM DISK, PRINTER
100 REM REPORT WRITER
110 OPTION BASE 1
120 DIM D$(10),T(10),TT(10),NN$(140
    ),P(140)
130 FOR I=1 TO 10 :: READ D$(I):: N
    EXT I
140 DATA JAN 1,JAN 8,JAN 15,JAN 22,
    JAN 29,FEB 5,FEB 12,FEB 19,FEB
    26,MAR 4
150 DISPLAY AT(4,6)ERASE ALL:"REPOR
    T WRITER"
160 DISPLAY AT(7,3):"CHOOSE:" :: DI
    SPLAY AT(8,5):"1 GROUP 1" :: DI
    SPLAY AT(9,5):"2 GROUP 2"
170 DISPLAY AT(10,5):"3 GROUP 3" ::
     DISPLAY AT(12,5):"4 COMPLETE C
    LASS"
180 CALL KEY(0,KEY,ST)
190 IF KEY<49 OR KEY>52 THEN 180
200 G1=KEY-48 :: CALL HCHAR(7,3,32,
    192)
210 OPEN #1:"RS232.BA=600",VARIABLE
     132
220 REM SET FOR COMPRESSED PRINT
230 ESC$=CHR$(27):: PRINT #1:ESC$&"
    P"&"D"&ESC$&"\"
240 OPEN #3:"DSK1.SAMPLE",INTERNAL,
    INPUT ,VARIABLE 192
```

```
250 I=0 :: L$="A"
260 GOSUB 880 :: GOSUB 930
270 INPUT #3:G,N$,F$,A$,P$,BD,R$,C$
280 IF N$="ZZZ" THEN 490
290 IF G1=4 THEN 310
300 IF G1<>G THEN 270
310 IF SEG$(C$,1,5)="AUDIT" THEN 27
    0
320 C$=SEG$(N$,1,1):: IF L$<>C$ THE
    N L$=C$ :: PRINT #1 :: L=L+1
330 PRINT #1:TAB(10);N$;", ";F$;TAB
    (44);
340 TA=0 :: TP=0
350 IF R$="" THEN R$="0000000000"
360 FOR J=1 TO LEN(R$)
370 A$=SEG$(R$,J,1):: IF A$="1" THE
    N TA=TA+1

380 IF A$="1" OR A$="0" THEN TP=TP+
    1 :: T(J)=T(J)+VAL(A$):: TT(J)=
    TT(J)+1
390 PRINT #1:A$;"{4 SPACES}";
400 NEXT J
410 FOR JJ=J TO 10 :: PRINT #1:"
    {5 SPACES}";:: NEXT JJ
420 I=I+1 :: NN$(I)=F$&" "&N$
430 IF TP=0 THEN P(I)=0 :: GOTO 450
440 P(I)=INT(TA*100/TP)
450 PRINT #1,USING "{16 SPACES}##/##
    {5 SPACES}###":TA,TP,P(I)
460 L=L+1 :: IF L=48 THEN GOSUB 870
     :: GOSUB 930

470 IF A$="-" THEN I=I-1
480 GOTO 270
490 GOSUB 950
500 PRINT #1
510 PRINT #1:TAB(10);"REPORTS: ";TA
    B(42);
520 FOR J=1 TO 10
530 PRINT #1,USING "###  ":T(J);
540 TAT=TAT+T(J):: NEXT J
550 PRINT #1: :TAB(10);"ENROLLED: "
    ;TAB(42);
560 FOR J=1 TO 10
570 PRINT #1,USING "###  ":TT(J);
580 TE=TE+TT(J):: NEXT J
590 PRINT #1: : :TAB(10);"PERCENT R
    EPORTS:";TAB(42);
600 FOR J=1 TO 10
610 PRINT #1,USING "###  ":T(J)*100
    /TT(J);
620 NEXT J
630 PRINT #1:TAB(120);INT(TAT*100/T
    E)
640 GOSUB 870
650 PRINT #1: :TAB(10);"DATE";TAB(3
    0);"REPORTS"
660 PRINT #1:TAB(10);"----";TAB(30)
    ;"-------": :
670 FOR J=1 TO 10
680 A$=RPT$("*",T(J))
690 PRINT #1: :TAB(10);D$(J);TAB(30
    );T(J);"  ";A$
700 NEXT J
710 GOSUB 870
720 B=1
730 B=2*B :: IF B<=I THEN 730
740 B=INT(B/2):: IF B=0 THEN 790
750 FOR J=1 TO I-B :: C=J
760 D=C+B :: IF P(C)<=P(D)THEN 780
770 AA=P(C):: AA$=NN$(C):: P(C)=P(D
    ):: NN$(C)=NN$(D):: P(D)=AA ::
    NN$(D)=AA$ :: C=C-B :: IF C>0 T
    HEN 760
780 NEXT J :: GOTO 740
790 GOSUB 970
800 FOR J=I TO 1 STEP -1
810 IF P(J)=0 AND FL=0 THEN GOSUB 1
    000
820 PRINT #1:TAB(46);
830 PRINT #1,USING "###{8 SPACES}##
    #####################":P(J),N
    N$(J)
840 L=L+1 :: IF L=48 THEN GOSUB 870
     :: GOSUB 970
850 NEXT J
860 STOP
870 PRINT #1:CHR$(12)
880 PRINT #1:TAB(58);"SAMPLE CLASS"
890 IF G1=4 THEN 910
900 PRINT #1: :TAB(60);"GROUP";G1
910 PRINT #1: :TAB(53);"BOOK REPORT
    S PRESENTED"
920 PRINT #1: :TAB(57);"FIRST TERM
    1984" :: RETURN
930 PRINT #1: : :TAB(43);"JAN  JAN
     JAN  JAN  JAN  FEB  FEB  FEB
    FEB  MAR"
940 PRINT #1:TAB(10);"NAME";TAB(43)
    ;" 1{4 SPACES}8{3 SPACES}15
    {3 SPACES}22{3 SPACES}29
    {4 SPACES}5{3 SPACES}12
    {3 SPACES}19{3 SPACES}26
    {4 SPACES}4";TAB(110);"TOTAL";T
    AB(121);"%"
950 PRINT #1:TAB(10);"----";TAB(43)
    ;"---  ---  ---  ---  ---  ---
    ---  ---  ---  ---";TAB(110);"
    -----";TAB(120);"---": :
960 L=0 :: RETURN
970 PRINT #1: : :TAB(44);"PERCENT";
    TAB(57);"NAME"
980 PRINT #1:TAB(44);"-------";TAB(
    57);"----": : :
990 L=0 :: RETURN
1000 FOR K=1 TO J
1010 S=POS(NN$(K)," ",1)
1020 S1=POS(NN$(K)," ",S+1):: IF S1
     =0 THEN 1030 ELSE S=S1
1030 NN$(K)=SEG$(NN$(K),S+1,LEN(NN$
     (K))-S)&", "&SEG$(NN$(K),1,S-1
     )
1040 NEXT K
1050 B=1
1060 B=2*B :: IF B<=J THEN 1060
1070 B=INT(B/2):: IF B=0 THEN 1120
1080 FOR K=1 TO J-B :: C=K
1090 D=C+B :: IF NN$(C)>=NN$(D)THEN
      1110
1100 A$=NN$(C):: NN$(C)=NN$(D):: NN
     $(D)=A$ :: C=C-B :: IF C>0 THE
     N 1090
1110 NEXT K :: GOTO 1070
1120 FOR K=1 TO J :: S=POS(NN$(K),"
     ,",1)
1130 NN$(K)=SEG$(NN$(K),S+2,LEN(NN$
     (K))-S+1)&" "&SEG$(NN$(K),1,S-
     1)
1140 NEXT K
1150 FL=1 :: RETURN
1160 END
```

Jim Butterfield, Associate Editor

A Program Critique

Part 2

This month we continue with comments on Bud Rasmussen's program to copy files on the Commodore 64 with a single disk unit. At this point the program has obtained a filename. The filename is kept in two forms: the short form ("FILENAME") and the longer form for writing ("FILENAME,P,W"). We will use the short form when we open the file for reading.

In this session, we'll track the mnemonics that open the error channel, initialize the disk, and input the file into RAM memory.

```
                      ;
                      ; DISK I/O ROUTINE
                      ;
                      ;
C18A  A9  00          DIOR  LDA  #0         ;CLEAR
C18C  8D  60  03            STA  ISF        ;INPUT STAT FLAG
C18F  8D  61  03            STA  IEC        ;INPUT ERR CODE
                      ;
```

This is probably overkill. The flags should be zeroed close to where they are used, if necessary.

```
C192  A2  22                LDX  #IPBML  ;PRINT
C194  A0  C1                LDY  #>IPBM  ;'INPUT
C196  A9  AD                LDA  #<IPBM  ;PHASE BEGUN'
C198  20  75  C1            JSR  PR      ;MSG
                      ;
```

A Friendly Message

In keeping with the friendly style, a message is printed telling the user what's going on. We'll find the message in-line very shortly.

```
C19B  A9  0F                LDA  #15     ;SET
C19D  A2  08                LDX  #8      ;COMMAND
C19F  A0  0F                LDY  #15     ;CHANNEL
C1A1  20  BA  FF            JSR  SETLFS
C1A4  20  C0  FF            JSR  OPEN    ;OPEN COMMD CH
                      ;
```

The command channel is opened. This is quite important: We'll get all our error messages from this channel. It should always be opened before other disk activities are started.

```
C1A7  20  3F  C4            JSR  ID      ;INIT DISK
C1AA  4C  CF  C1            JMP  SNI     ;GOTO SET NAME
                                          INPUT
```

We send the initialize command to the disk over the command channel. This is not vital, but a good precaution. It's a subroutine within the program; we'll meet it much later.

We need to jump over the message to continue with the program. Here's the message:

```
                      ;
                      ;
                      ; INPUT PHASE BEGUN MESSAGE
                      ;
                      ;
C1AD  0D  0D  12  IPBM    .BYTE$0D,$0D,$12
C1B0  2A  2A  2A          .ASC "*** INPUT PHASE BEGUN ***"
C1CD  0D  0D              .BYTE$0D,$0D
                  IPBML   =  *-IPBM
                      ;
```

Now we're ready to open the input file in preparation for reading it. We use the short name, since the last four characters (,S,W) aren't needed or wanted for an input file.

```
                      ;
                      ; OPEN INPUT
                      ;
                      ;
C1CF  AD  AA  02  SNI   LDA  IFNL    ;LOAD INPUT
                                      FNAME LEN
C1D2  A2  40            LDX  #<FNA   ;LOAD FILENAME LO
C1D4  A0  03            LDY  #>FNA   ;LOAD FILENAME HI
C1D6  20  BD  FF        JSR  SETNAM
                      ;
```

We're doing things backwards from the equivalent BASIC coding. If we code OPEN 2,8,2,"HOTDOG" in BASIC, we've now placed the "HOTDOG" part of the command. Now let's put in the 2,8,2 sequence:

```
                      ;
                      ; SET LOGICAL FILE (INPUT)
                      ;
C1D9  A9  02      SLFI  LDA  #2      ;LOAD LOGICAL
                                      FILE #
C1DB  A2  08            LDX  #8      ;LOAD DEVICE
```

```
                                          ADDRESS
C1DD A0 02             LDY  #2          ;LOAD SEC.
                                          ADDRESS
C1DF 20  BA  FF        JSR  SETLFS
                  ;
```

And finally, the OPEN itself:

```
                  ;
                  ; OPEN FILE (INPUT)
                  ;
                  ;
C1E2  20  C0  FF  OFI  JSR  OPEN
                  ;
```

Error Check

Now we'll check to see if the OPEN took place without error:

```
C1E5  A5  90           LDA  IOS         ;TEST
C1E7  F0  0B           BEQ  OCI         ;STATUS
C1E9  8D  60  03       STA  ISF         ;STORE STATUS
                                          FLAG
C1EC  A9  01           LDA  #1          ;SET/STORE
C1EE  8D  61  03       STA  IEC         ;ERROR CODE
C1F1  4C  4F  C2       JMP  IE          ;INPUT ERROR
                  ;
```

Location $90—called IOS here—is the familiar BASIC ST flag. If it's zero, we are OK and can proceed to read the file. If not, we must advise, abort, or take other appropriate action.

But this flag is not enough. ST, or hex 90, tells us only if the transfer of information (in this case, filename) has been passed to the disk correctly. After the information gets to the disk, there may be other problems.

If the file does not exist, or for any other reason cannot be opened, the disk will know there's an error; but the computer will not. The computer must ask the disk to deliver information on possible errors over its command channel. The command channel is open and ready to receive this data (we opened 15, remember), but we must ask for it.

To do the job right, we must think about coding along the following lines:

```
LDX  #15      ; command channel
JSR  $FFC6    ; input
JSR  $FFE4    ; get a character
PHA           ; stash it
JSR  $FFCC    ; close channel
PLA           ; unstash character
CMP  #$30     ; is it 0?
BNE  ERROR    ; nope, we have problem
```

A Better Way

The above is minimum coding. It would be better to create a more elaborate subroutine which brings in the whole message from the error channel and stores it in memory. (The message would end with $0D, the Return character.) Then we could check the first character for $30 (ASCII zero, start of the OK message); if not, we'd be able to print the whole error message.

Here comes the coding for a good OPEN:

```
                  ; OPEN CHANNEL (INPUT)
                  ;
                  ;
C1F4  A2  02      OCI  LDX  #2          ;OPEN
C1F6  20  C6  FF       JSR  CHKIN       ;CHANNEL #2
                  ;
C1F9  A5  90           LDA  IOS         ;TEST
C1FB  F0  0B           BEQ  LBSA        ;STATUS
C1FD  8D  60  03       STA  ISF         ;STORE STATUS
                                          FLAG
C200  A9  02           LDA  #2          ;SET/STORE
C202  8D  61  03       STA  IEC         ;ERROR CODE
C205  4C  4F  C2       JMP  IE          ;INPUT ERROR
                  ;
```

I wish the comments said "connect channel" rather than "open channel." The OPEN (as we know it in BASIC) has been performed successfully. Now, we're establishing a connection to the input file preparatory to reading.

```
                  ;
                  ; LOAD BUFFER START ADDRESS
                  ;
                  ;
C208  A9  00      LBSA LDA  #0          ;LOAD BFR
C20A  85  FB           STA  BAL         ;ADDR LO
C20C  AD  3D  C4       LDA  SP          ;LOAD BFR
C20F  85  FC           STA  BAH         ;ADDR HI
                  ;
C211  A0  00           LDY  #0          ;BUFFER INDEX=0
                  ;
```

Just before reading, we set up the memory address into which we will start to read. The low part of the address is zero; the high part is stored as a constant in the program (SP undoubtedly stands for Start Page). Immediate addressing could be used to set the start page if preferred.

```
                  ;
                  ; INPUT LOOP
                  ;
                  ;
C213  20  CF  FF  IL   JSR  CHRIN       ;GET CHARACTER
C216  91  FB           STA  (BAL),Y     ;STORE CHARACTER
C218  E6  FB           INC  BAL         ;INCR LO BYTE
C21A  D0  0C           BNE  TIS         ;IF NOT 0, TEST STAT
C21C  E6  FC           INC  BAH         ;INCR HI BYTE
C21E  A5  FC           LDA  BAH         ;LOAD HI BYTE AND,
C220  CD  3E  C4       CMP  EP          ;CHECK FOR END
                                          ADDR
C223  90  03           BCC  TIS         ;IF LO, TEST STAT
C225  4C  3B  C2       JMP  DSP
```

CHRIN Or CHRGET

Rasmussen uses the CHRIN routine ($FFCF) to get from the file. I prefer CHRGET ($FFE4), but the difference is minor with files. Either call gets from the file rather than keyboard/screen because we have switched the input channel with our call to CHKIN ($FFC6).

Some programmers would prefer to step the Y register through its range rather than change the indirect address each time. In principle, the Y register technique is faster; but in this case, it's doubtful that the speed difference could be observed. Timing of this whole section is governed almost totally by disk speed.

The program checks carefully to make sure that the data does not overrun the memory space available.

```
                    ;
                    ; TEST INPUT STATUS
                    ;
                    ;
C228  A5 90         TIS   LDA  IOS       ;LOAD STATUS
C22A  F0 E7               BEQ  IL        ;IF 0, CARRY ON
C22C  C9 40               CMP  #EOFI     ;TEST FOR
C22E  F0 23               BEQ  EOF       ;EOF
C230  8D 60 03            STA  ISF       ;STORE STATUS
                                          FLAG
C233  A9 03               LDA  #3        ;SET/STORE
C235  8D 61 03            STA  IEC       ;ERROR CODE
C238  4C 4F C2            JMP  IE        ;INPUT ERROR
                    ;
```

Again we test the ST status byte (IOS); in this case, we're primarily interested in an end-of-file signal which would be flagged by a value of hex 40 (decimal 64) in ST.

Once again, the error routines are quite elaborate. It's my opinion that there is little need to check the disk error channel during the read phase; error notices will wait until we ask for them at end of file.

Opening The File

If we run out of memory, we come to DSP:

```
                    ;
                    ; DECREMENT START PG BY HEX 10
                    ; AND TRY AGAIN,
                    ; TO GIVE YOU 16 MORE BLKS.
                    ;
                    ;
C23B  38            DSP   SEC
C23C  AD 3D C4            LDA  SP        ;LOAD START PG
C23F  E9 10               SBC  #H10      ;SUBT HEX 10
C241  8D 3D C4            STA  SP        ;STORE IT BACK
C244  20 CC FF            JSR  CLRCHN    ;CLEAR CHANNEL
C247  A9 02               LDA  #2        ;SET CH 2
C249  20 C3 FF            JSR  CLOSE     ;FOR CLOSE
C24C  4C CF C1            JMP  SNI       ;START ALL OVER
```

I'm not sure what is going on here. The coding intention is this: If it doesn't fit, allocate an extra 4K and try again.

An Endless Loop

This is puzzling. If the 4K was available, why not make it available in the first read and save the trouble?

There's also a pitfall here. Suppose we allocate the extra 4K, and the program still doesn't fit into memory. We'll end up in an endless loop, since we will come back to DSP, do it again, and so on, and so on.

I'd prefer to allocate as much memory as possible right away, and quit if the program doesn't fit.

```
                    ;
                    ; INPUT ERROR
                    ;
                    ;
C24F  20 E7 FF      IE    JSR  CLALL     ;CLOSE ALL FILES
C252  00                  BRK
                    ;
```

This is a programmer's error termination. The program will stop and break to the monitor, if there is a monitor in place. The programmer can then examine memory locations to see what the trouble is.

If there is not a monitor in the machine, the program will terminate with a READY statement and no other explanation.

Extra Work

For general use, the program would benefit from additional work in this area so that the user would see a meaningful message. This is almost out of character: The messages are so well presented in other parts of the program that their absence here is very noticeable indeed.

```
                    ;
                    ; END OF FILE
                    ;
                    ;
C253                EOF   =    *
                    ;
C253  A5 FB               LDA  BAL       ;SAVE
C255  85 FD               STA  EAL       ;LAST
C257  A5 FC               LDA  BAH       ;ADDRESS
C259  85 FE               STA  EAH       ;OF FILE
C25B  20 CC FF            JSR  CLRCHN    ;CLEAR CHANNEL
                    ;
```

Wrapping It Up

The end address (plus one, of course) is stored away, and the file disconnected. I would check the disk error channel at this point. Any errors that may have accumulated during the input phase will be waiting.

Now we may close the file and print an advisory message:

```
C25E  A9 02               LDA  #2        ;SET CH 2
C260  20 C3 FF            JSR  CLOSE     ;FOR CLOSE
                    ;
C263  A2 88               LDX  #IPFML    ;PRINT
C265  A0 C2               LDY  #>IPFM    ;'INPUT
C267  A9 6F               LDA  #<IPFM    ;PHASE FINISHED'
C269  20 75 C1            JSR  PR        ;MSG
                    ;
C26C  4C F7 C2            JMP  SOP       ;GOTO START OUT
                                          PHASE
                    ;
                    ;
                    ; INPUT PHASE FINISHED MESSAGE
                    ;
                    ;
C26F  12            IPFM  .BYTE$12
C270  20 20 49            .ASC " INPUT PHASE FINISHED.   "
C28F  0D 0D 12            .BYTE$0D,$0D,$12
C292  20 20 52            .ASC " REMOVE INPUT DISKETTE. "
C2B1  0D 0D 12            .BYTE$0D,$0D,$12
C2B4  20 20 49            .ASC " INSERT OUTPUT DISKETTE. "
C2D3  0D 0D 12            .BYTE$0D,$0D,$12
C2D6  20 20 50            .ASC " PRESS RETURN KEY WHEN
                           READY. "
C2F5  0D 0D               .BYTE$0D,$0D
C2F7                IPFML =    *-IPFM
                    ;
                    ;
                    ; START OUTPUT PHASE
                    ;
C2F7                SOP   =    *
```

The input phase is complete. Next time, we'll take a look at output. ©

Atari Softkey

Thomas A. Marshall

This utility allows you to GOTO any line in a program while it's running, simply by pressing a console key. See the "Automatic Proofreader" article on page 180 before typing in programs.

To access the OPTION, SELECT, and START keys on the Atari keyboard console, you can use the following BASIC program:

```
MK 0 GOTO 10
GC 1 ? "OPTION":GOTO 20
EK 2 ? "SELECT":GOTO 20
BJ 3 ? "START ":GOTO 20
FG 10 ? "This is a demonstration of th
      e"
GP 11 ? "use of Atari's console keys."
HK 20 IF PEEK(53279)=3 THEN GOTO 1
HO 30 IF PEEK(53279)=5 THEN GOTO 2
IB 40 IF PEEK(53279)=6 THEN GOTO 3
AA 50 GOTO 20
```

However, this requires that the computer be tied up in a loop, lines 20 to 50.

A much better way to accomplish the same thing is for a machine language program to check the console keys during the *vertical blank period*. (This is the time that the television's electron beam ends at the lower right corner of the screen until it begins again at the top left corner of the screen.) If a console key is pressed, the machine language program will execute a "GOTO line number" where the line number corresponds to the following keys pressed:

GOTO 1 for OPTION
GOTO 2 for SELECT
GOTO 3 for START
GOTO 4 for SHIFT & OPTION
GOTO 5 for SHIFT & SELECT
GOTO 6 for SHIFT & START

Note that we have doubled the effective number of console keys by adding the SHIFT key. Using this technique, the BASIC programmer can go directly to any portion of his program without stopping the program and typing GOTO *line number*.

An Automatic RUN

If you are really lazy, you can have the BASIC line, 3 RUN, so that your BASIC program will RUN when the START key is pressed, regardless of whether the BASIC program was running beforehand or not.

Program 1 creates an AUTORUN.SYS file. Note that this file resets the memory location, MEMLO, that points to the beginning of a BASIC program. Thus, the vertical blank machine language routine resides safely below the BASIC program. The drawback to this technique is that the machine language program will be erased when you go to DOS.

Also Autoruns

An additional feature included in the disk version of "Atari Softkey" is the ability to autorun any BASIC program saved on the disk. Program 2 is a demonstration program which will be RUN automatically by the AUTORUN.SYS file. So, Program 2 should be saved on the disk with the filename as in the AUTORUN.SYS file. Program 2 currently has the filename GOTO.BAS, defined in line 40 of Program 1 by F$="RUN D:GOTO.BAS".

The Tape Version

For Atari owners who do not have a disk drive, Program 3 POKEs Softkey into page 6. You need to initialize the machine language (ML) routine with the USR statement in line 120. Program 3 is essentially the same as Program 1, but with the autorun feature removed. Again, whenever the console keys are pressed, lines 1–6 in Program 2 will be executed as described above.

However, remember that if there is no line number in the BASIC program corresponding to the console key pressed, an "ERROR 12", line not found, will occur.

The ML program is initialized by placing the low and high address of the start of the ML program into memory addresses 736–737 (RUNAD $2E0–$2E1). Upon completion of DOS.SYS load, the computer will run the ML program pointed to by this address. After resetting several vectors, the ML program sets the Vertical Blank Interrupt (VBI) vector using the deferred mode.

The Deferred Mode

I have used the deferred mode (accumulator = 7), since there are about 20,000 machine cycles available versus about 3800 cycles in the immediate mode (accumulator = 6). Thus, the ML routine checks whether the SHIFT and the console keys are pressed during the vertical blank period. Once the keys are pressed, the ML program jumps to the subroutine that sounds the keyboard click and resets the pointer to the editor routine so that the ML can perform the GOTO *line number* input. It then simulates a press of the BREAK key so that the editor buffer is emptied and the new editor pointers are executed. Once the BASIC G.*line number* is in the editor buffer, the editor pointer is reset. A RETURN, CHR$(155), is placed in the editor buffer to execute the GOTO *line number* statement.

Softkey has many applications. I have found it most useful in a program that required the modification of DATA statements. You can RUN the BASIC program simply by pressing the START key. Another application is to go directly to subroutines without going through a menu selection.

Program 1: Atari Softkey

```
BL 10  REM Atari Softkey
EF 20  GRAPHICS 0:? "Insert a DOS 2.0S
       diskette":? "with DOS.SYS in dri
       ve 1"
FN 30  ? :? "Press RETURN when you have
        done this"
ND 40  DIM F$(18):B=0:F$="RUN D:GOTO.BA
       S":F$(4,4)=CHR$(34):REM 34=ASCII
        FOR "
EN 50  IF PEEK(764)=12 THEN POKE 764,25
       5:GOTO 70
AE 60  GOTO 50
NH 70  ? :? "Now writing the AUTORUN.SY
       S file"
DE 80  TRAP 100:CLOSE #1
KC 90  OPEN #1,8,0,"D:AUTORUN.SYS":TRAP
        4:GOTO 110
FM 100 CLOSE #1:? :? "Can't open AUTOR
       UN.SYS file":END
JH 110 FOR I=1 TO 292:TRAP 180:READ A:
       B=B+A:TRAP 210:PUT #1,A:NEXT I:
       TRAP 40000
BD 120 IF A<>96 THEN 170
KE 130 IF B<>30720 THEN 190
OA 140 FOR I=1 TO 18-LEN(F$):PUT #1,32
       :NEXT I
LA 150 FOR I=LEN(F$) TO 1 STEP -1:PUT
       #1,ASC(F$(I)):NEXT I:CLOSE #1
FN 160 ? :? " DATA ok, write successfu
       l.":END
OE 170 ? :? "There are too many DATA e
       ntries":GOTO 200
DM 180 ? "There are not enough DATA en
       tries":GOTO 200
FP 190 ? :? "Bad number in DATA statem
       ents"
HL 200 CLOSE #1:? "RECHECK the entries
       !":END
BG 210 ? :? :? "Error-";PEEK(195);" wh
       en attempting disk write.":CLOS
       E #1:END
HI 220 REM
FH 230 REM The following is the decimal
KD 240 REM equivalent of the machine
GA 250 REM language.  It must be typed
CA 260 REM perfectly in order to
BB 270 REM function.
HD 280 REM
FI 290 DATA 255,255,0,30,243,30
GP 300 DATA 165,12,141,57,30,165,13,14
       1,58,30,169,56,133,12,169,30,13
       3,13,32,63,30,169,244,141,231,2
       ,169,30,141,232
LF 310 DATA 2,173,243,30,240,10,169,20
       5,141,89,30,169,6,141,90,30,160
       ,105,162,30,169,7,32,92,228,96,
       32,64,21,32
FN 320 DATA 10,30,96,169,85,141,33,3,1
       69,30,141,34,3,96,169,0,141,33,
       3,169,228,141,34,3,96,251,243,5
       1,246,220
HA 330 DATA 30,163,246,51,246,60,246,7
       6,228,243,51,46,71,0,7,169,8,14
       1,31,208,173,31,208,205,104,30,
       240,100,141,104
CI 340 DATA 30,201,7,240,93,141,104,30
       ,173,103,30,208,85,173,104,30,2
       01,3,208,19,169,49,141,100,30,1
       73,15,210,41,8
GD 350 DATA 208,51,169,52,141,100,30,2
       08,44,201,5,208,19,169,50,141,1
       00,30,173,15,210,41,8,208,28,16
       9,53,141,100,30
PF 360 DATA 208,21,201,6,208,32,169,51
       ,141,100,30,173,15,210,41,8,208
       ,5,169,54,141,100,30,169,3,141,
       103,30,32,216
MK 370 DATA 252,32,63,30,169,0,133,17,
       76,98,228,172,103,30,240,9,185,
       99,30,206,103,30,160,1,96,32,74
       ,30,169,155
NJ 380 DATA 160,1,96,18
AK 390 DATA 224,2,225,2,0,30,206,6,255
       ,6
FC 400 DATA 172,243,30,240,9,185,237,6
       ,206,243,30,160,1,96,32,74,30,1
       69,220,141,89,30,169,30,141,90,
       30,169,155,160
HL 410 DATA 1,96
```

Program 2: Atari Softkey Test Program

```
MK 0  GOTO 10
DJ 1  ? " {TAB}OPTION{UP}":END
CB 2  ? " {TAB}SELECT{UP}":END
PA 3  ? " {TAB}START {UP}":END
OH 4  ? " {TAB}SHIFT-OPTION{UP}":END
MP 5  ? " {TAB}SHIFT-SELECT{UP}":END
JO 6  ? " {TAB}SHIFT-START {UP}":END
EO 10 ? "This is a test of"
NM 11 ? "Atari Softkey!"
```

Program 3: Atari Softkey (ML) For Tape Drive Users

```
EA 100 FOR I=0 TO 204:READ A:B=B+A:POK
       E 1536+I,A:NEXT I
```

```
GC 110 IF B<>19990 OR I<>205 THEN ? "R
       echeck DATA statements.":? "The
       y do not correctly total":END
CL 120 A=USR(1536)
EG 200 DATA 104,169,1,133,2,169,6,133,
       3,165,9,9,2,133,9,160,67,162,6,
       169,7,32,92,228,96,169,47,141,3
       3,3
KD 210 DATA 169,6,141,34,3,96,169,0,14
       1,33,3,169,228,141,34,3,96,251,
       243,51,246,182,6,163,246,51,246
       ,60,246,76
CE 220 DATA 228,243,49,46,71,0,7,169,8
       ,141,31,208,173,31,208,205,66,6
       ,240,100,141,66,6,201,7,240,93,
       141,66,6
DM 230 DATA 173,65,6,208,85,173,66,6,2
       01,3,208,19,169,49,141,62,6,173
       ,15,210,41,8,208,51,169,52,141,
       62,6,208
FF 240 DATA 44,201,5,208,19,169,50,141
       ,62,6,173,15,210,41,8,208,28,16
       9,53,141,62,6,208,21,201,6,208,
       32,169,51
JA 250 DATA 141,62,6,173,15,210,41,8,2
       08,5,169,54,141,62,6,169,3,141,
       65,6,32,216,252,32,25,6,169,0,1
       33,17
HH 260 DATA 76,98,228,172,65,6,240,9,1
       85,61,6,206,65,6,160,1,96,32,36
       ,6,169,155,160,1,96
NL 270 ? " Now type in program listing
       "
GB 280 ? " number 2 to demonstrate"
CC 290 ? " Atari Softkey."
```

©

64 EXPLORER

Larry Isaacs

In this month's column we will complete our look at line drawing in the 64's bitmapped graphics mode. We will deal with both hi-res and multicolor bitmapped graphics. Fortunately, the same general principles apply to both. Last month we saw how a routine to draw lines might look in BASIC. Actually executing the routine would show that BASIC is much too slow to be of much use for this task. At the end of last month's article we took the first step in putting together a set of machine language routines. This month we will complete the set.

First, here is a summary of the features of these drawing routines. The range of coordinates supported is 0 to 319 for X, and 0 to 199 for Y, when in hi-res mode. For multicolor mode, the range is 0 to 159 for X, and 0 to 199 again for Y. It is up to the user to insure that coordinates are within these ranges. Using coordinates which are too far out of range could cause the 64 to crash. In both hi-res and multicolor mode, the location of 0,0 is found at the lower left corner of the display.

Saving Memory For BASIC

The bitmap memory is placed at 57344 ($E000), underneath the operating system ROM. This avoids taking memory away from BASIC. Since this makes the bitmap data difficult to PEEK directly from BASIC, a routine is provided to perform this function. The screen memory is placed at 51200 ($C800), just below where the DOS Wedge loads. Use of these graphics routines should not conflict with the DOS Wedge, but may conflict with other BASIC enhancement software.

Last month we began by writing four of the required routines. This month we are going to upgrade two of those to accept arguments, and add six more. As was mentioned last time, we will execute these routines via a jump table at the beginning of the machine code. This will provide us fixed locations to SYS to, even if modifications or additions are made later. The following is a list of the routines found in the jump table:

Loc.	Description
JT+0	Save screen parameters
JT+3	Restore saved screen parameters
JT+6	Enable graphics screen
JT+9	Clear graphics screen
JT+12	Move graphics cursor to X,Y
JT+15	Plot pixel at X,Y
JT+18	Draw line to X,Y
JT+21	Set drawing mode
JT+24	Set drawing color (multicolor)
JT+27	Read bitmap byte (a function)

The jump vector location of these routines is shown as the variable JT plus an offset. To obtain the actual address, JT should be set to the base of the jump table, which is 49152 or $C000. The following table gives the syntax for using each of the routines in the jump table.

```
SYS JV                    :REM SAVE SCREEN
SYS JV+3                  :REM RESTORE SCREEN
SYS JV+6,MODE             :REM ENABLE GRAPHICS
      MODE: 0=HI-RES, 1=MULTICOLOR
SYS JV+9,C0,C1            :REM CLEAR SCREEN
      C0="OFF" COLOR, C1="ON" COLOR
      USE IF HI-RES BITMAP MODE
SYS JV+9,C0,C1,C2,C3      :REM CLEAR SCREEN
      C0=BACKGROUND, C1=FOREGROUND 1
      C2=FOREGROUND 2, C3=FOREGROUND 3
      USE IF MULTICOLOR MODE
SYS JV+12,X,Y             :REM MOVE
SYS JV+15,X,Y             :REM PLOT
SYS JV+18,X,Y             :REM DRAW
SYS JV+21,DM              :REM SET DRAWING MODE
      DM: 0=FLIP, 1=DRAW, 2=ERASE
SYS JV+24,C               :REM SELECT COLOR
      WORKS ONLY FOR MULTICOLOR MODE
```

The last routine in the jump table (offset =27) is handled differently because it should be called by the USR function. To set it up as the USR function, execute the statement:

```
POKE 785,PEEK(JV+28) :POKE 786,PEEK(JV+29)
```

Once this is done, you may read bytes from the bitmap memory with the statement

```
BYTE=USR( OFFSET )
```

where OFFSET is the offset from the base address of the byte you wish to fetch.

A Graphics Cursor

The philosophy behind this is that these graphics commands differ slightly for other graphics enhancements to BASIC. Typically, enhancements will add a line-drawing command which always requires both end points. In the routines above, an internal graphics cursor is maintained. Lines are drawn from this graphics cursor to a specified end point. Whenever a line is drawn, the new end point becomes the graphics cursor location. Thus, successive executions of the DRAW routine will create a series of connected lines.

Also, you have a choice of three drawing modes, flip, draw, and erase. The draw mode causes points along the lines to be set to the on state, or to the selected color if in multicolor graphics. Erasing causes dots to be set to the off state or background color. The flip mode involves switching the pixels to their opposite state In the case of multicolor mode, pixels of the selected color are flipped to the background color, and vice versa. Pixels not of the selected color are flipped to the other nonselected color.

To provide a simple example of how to put these routines to use in a program, the following program draws an interesting circular pattern in hi-res mode. Once the pattern is drawn, the program will wait for you to press a key

```
10 JT=49152:SYS JT:REM SAVE SCREEN
20 SYS JT+6,0:SYS JT+9,1,2:REM INIT SCREE
   N
30 SYS JT+21,0:REM FLIP MODE
40 FOR I=0 TO 6.24 STEP .035
50 X=50*COS(I):Y=50*SIN(I)
60 SYS JT+12,160+X,100+Y:REM MOVE
70 SYS JT+18,160-X,100-Y:REM DRAW
80 NEXT
90 GET Z$:IF Z$="" THEN 90
100 SYS JT+3:REM RESTORE TEXT SCREEN
```

To put the required machine code into memory, run the BASIC program shown below.

Next month we'll explore some of the more interesting aspects of the machine language source code listing.

BASIC Program

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

```
1 READ LN,SA,EA:LN=LN+30              :rem 146
10 FOR I=0 TO EA-SA                   :rem 232
20 READ BY:POKE SA+I,BY:SUM=SUM+BY
                                      :rem 120
30 IF INT((I+1)/8)*8<>(I+1) THEN 60
                                      :rem 242
40 READ CS:IF CS<>SUM THEN 80         :rem 123
50 SUM=0:LN=LN+10                     :rem 254
60 NEXT                               :rem 165
70 PRINT "SUCCESSFUL LOAD":END        :rem 105
80 PRINT "ERROR IN LINE";LN:END       :rem 104
500 DATA 500                          :rem 68
510 DATA 49152                        :rem 181
520 DATA 50087                        :rem 181
530 DATA 76,47,192,76,72,192,76,9,740
                                      :rem 57
540 DATA 193,76,90,193,76,156,193,76,1053
                                      :rem 255
550 DATA 59,194,76,192,194,76,101,195,108
    7                                 :rem 53
560 DATA 76,115,195,76,137,195,0,0,794
                                      :rem 99
570 DATA 0,0,0,0,0,0,255,128,383      :rem 11
580 DATA 0,7,248,0,0,0,0,173,428      :rem 21
590 DATA 0,221,141,43,192,173,24,208,1002
                                      :rem 212
600 DATA 141,44,192,173,17,208,141,45,961
                                      :rem 230
610 DATA 192,173,22,208,141,46,192,96,107
    0                                 :rem 25
620 DATA 173,43,192,141,0,221,173,44,987
                                      :rem 182
630 DATA 192,141,24,208,173,45,192,141,11
    16                                :rem 68
640 DATA 17,208,173,46,192,141,22,208,100
    7                                 :rem 19
650 DATA 96,72,173,14,220,41,254,141,1011
                                      :rem 218
660 DATA 14,220,165,1,41,253,133,1,828
                                      :rem 69
670 DATA 104,96,72,165,1,9,2,133,582
                                      :rem 242
680 DATA 1,173,14,220,9,1,141,14,573
                                      :rem 225
690 DATA 220,104,96,164,254,240,13,160,12
    51                                :rem 65
700 DATA 0,145,251,200,208,251,230,252,15
    37                                :rem 53
710 DATA 198,254,208,243,164,253,240,10,1
    570                               :rem 123
720 DATA 136,240,5,145,251,136,208,251,13
    72                                :rem 67
730 DATA 145,251,96,32,97,192,160,0,973
                                      :rem 144
740 DATA 132,251,160,200,132,252,160,232,
    1519                              :rem 153
750 DATA 132,253,160,3,132,254,32,131,109
    7                                 :rem 12
760 DATA 192,44,40,192,16,20,160,0,664
                                      :rem 75
770 DATA 132,251,160,216,132,252,160,232,
    1535                              :rem 161
780 DATA 132,253,160,3,132,254,138,32,110
    4                                 :rem 11
790 DATA 131,192,169,0,133,251,169,224,12
    69                                :rem 84
800 DATA 133,252,169,64,133,253,169,31,12
    04                                :rem 72
810 DATA 133,254,169,0,32,131,192,76,987
                                      :rem 192
820 DATA 114,192,32,253,174,32,158,173,11
    28                                :rem 75
830 DATA 32,170,177,170,152,96,32,234,106
    3                                 :rem 24
840 DATA 192,141,34,192,142,35,192,32,960
                                      :rem 234
850 DATA 234,192,141,36,192,142,37,192,11
    66                                :rem 82
860 DATA 96,32,234,192,240,2,169,128,1093
                                      :rem 241
870 DATA 141,40,192,173,0,221,9,3,779
                                      :rem 34
880 DATA 73,3,141,0,221,173,24,208,843
                                      :rem 76
890 DATA 41,7,9,8,9,32,141,24,271     :rem 92
900 DATA 208,173,17,208,9,32,141,17,805
                                      :rem 131
910 DATA 208,44,40,192,16,12,173,22,707
                                      :rem 125
920 DATA 208,9,16,141,22,208,169,3,776
                                      :rem 90
930 DATA 208,10,173,22,208,41,239,141,104
    2                                 :rem 10
940 DATA 22,208,169,7,141,41,192,73,853
                                      :rem 141
950 DATA 255,141,42,192,169,255,141,38,12
    33                                :rem 82
960 DATA 192,96,32,246,192,44,40,192,1034
                                      :rem 241
970 DATA 48,21,173,36,192,10,10,10,500
                                      :rem 65
980 DATA 10,141,36,192,173,34,192,41,819
                                      :rem 188
```

```
990 DATA 15,13,36,192,76,163,192,173,860
                                 :rem 199
1000 DATA 36,192,10,10,10,10,141,36,445
                                 :rem 96
1010 DATA 192,32,234,192,41,15,13,36,755
                                 :rem 171
1020 DATA 192,141,36,192,32,234,192,170,1
     189                         :rem 121
1030 DATA 173,34,192,141,33,208,173,36,99
     0                           :rem 24
1040 DATA 192,76,163,192,32,246,192,162,1
     255                         :rem 129
1050 DATA 3,189,34,192,157,30,192,202,999
                                 :rem 243
1060 DATA 16,247,96,56,169,199,237,32,105
     2                           :rem 43
1070 DATA 192,72,74,74,74,133,252,160,103
     1                           :rem 20
1080 DATA 0,132,251,74,102,251,74,102,986
                                 :rem 220
1090 DATA 251,101,252,133,252,173,30,192,
     1384                        :rem 161
1100 DATA 174,31,192,45,42,192,44,40,760
                                 :rem 172
1110 DATA 192,16,6,10,72,138,42,170,646
                                 :rem 122
1120 DATA 104,24,101,251,133,251,138,101,
     1103                        :rem 133
1130 DATA 252,133,252,104,41,7,24,101,914
                                 :rem 207
1140 DATA 251,144,2,230,252,24,105,0,1008
                                 :rem 198
1150 DATA 133,251,165,252,105,224,133,252
     ,1515                       :rem 207
1160 DATA 173,30,192,45,41,192,170,96,939
                                 :rem 242
1170 DATA 169,0,168,44,39,192,16,7,635
                                 :rem 94
1180 DATA 80,2,177,251,77,38,192,44,861
                                 :rem 145
1190 DATA 40,192,48,10,61,47,194,133,725
                                 :rem 183
1200 DATA 97,189,47,194,208,8,61,55,859
                                 :rem 161
1210 DATA 194,133,97,189,55,194,73,255,11
     90                          :rem 94
1220 DATA 49,251,5,97,145;251,96,128,1022
                                 :rem 234
1230 DATA 64,32,16,8,4,2,1,192,319
                                 :rem 126
1240 DATA 48,12,3,32,156,193,32,97,573
                                 :rem 85
1250 DATA 192,32,171,193,32,0,194,76,890
                                 :rem 186
1260 DATA 114,192,169,1,149,106,169,0,900
                                 :rem 228
1270 DATA 149,107,56,189,34,192,253,30,10
     10                          :rem 73
1280 DATA 192,149,98,189,35,192,253,31,11
     39                          :rem 98
1290 DATA 192,149,99,16,20,169,255,149,10
     49                          :rem 99
1300 DATA 106,149,107,56,169,0,245,98,930
                                 :rem 238
1310 DATA 149,98,169,0,245,99,149,99,1008
                                 :rem 4
1320 DATA 96,21,98,208,4,149,106,149,831
                                 :rem 192
1330 DATA 107,96,165,99,74,133,103,165,94
     2                           :rem 39
1340 DATA 98,106,133,102,24,169,0,229,861
                                 :rem 230
1350 DATA 98,133,104,169,0,229,99,133,965
                                 :rem 250
1360 DATA 105,96,24,165,102,101,100,133,8
     26                          :rem 56
1370 DATA 102,170,165,103,101,101,133,103
     ,978                        :rem 151
1380 DATA 197,99,144,19,208,4,228,98,997
                                 :rem 224
1390 DATA 144,13,138,56,229,98,133,102,91
     3                           :rem 32
1400 DATA 165,103,229,99,133,103,56,96,98
     4                           :rem 40
1410 DATA 32,246,192,32,97,192,162,0,953
                                 :rem 184
1420 DATA 32,74,194,162,2,32,74,194,764
                                 :rem 137
1430 DATA 165,98,197,100,165,99,229,101,1
     154                         :rem 137
1440 DATA 144,62,32,130,194,36,107,16,721
                                 :rem 221
1450 DATA 10,32,159,193,56,169,0,229,848
                                 :rem 194
1460 DATA 108,133,108,32,171,193,32,0,777
                                 :rem 227
1470 DATA 194,230,104,208,4,230,105,240,1
     315                         :rem 103
1480 DATA 102,238,30,192,208,3,238,31,104
     2                           :rem 11
1490 DATA 192,32,154,194,144,9,24,173,922
                                 :rem 241
1500 DATA 32,192,101,108,141,32,192,32,83
     0                           :rem 3
1510 DATA 171,193,32,0,194,76,241,194,110
     1                           :rem 15
1520 DATA 162,1,181,98,180,100,149,100,97
     1                           :rem 17
1530 DATA 148,98,202,16,245,32,130,194,10
     65                          :rem 74
1540 DATA 36,107,16,10,32,159,193,56,609
                                 :rem 184
1550 DATA 169,0,229,108,133,108,32,171,95
     0                           :rem 23
1560 DATA 193,32,0,194,230,104,240,31,102
     4                           :rem 1
1570 DATA 24,173,32,192,101,108,141,32,80
     3                           :rem 10
1580 DATA 192,32,154,194,144,8,238,30,992
                                 :rem 246
1590 DATA 192,208,3,238,31,192,32,171,106
     7                           :rem 27
1600 DATA 193,32,0,194,76,60,195,32,782
                                 :rem 137
1610 DATA 159,193,76,114,192,32,234,192,1
     192                         :rem 132
1620 DATA 41,3,73,3,106,106,106,141,579
                                 :rem 120
1630 DATA 39,192,96,32,234,192,41,3,829
                                 :rem 144
1640 DATA 170,189,133,195,44,40,192,16,97
     9                           :rem 45
1650 DATA 3,141,38,192,96,0,85,170,725
                                 :rem 88
1660 DATA 255,32,170,177,170,152,24,105,1
     085                         :rem 121
1670 DATA 0,133,251,138,105,224,133,252,1
     236                         :rem 109
1680 DATA 32,97,192,160,0,177,251,32,941
                                 :rem 187
1690 DATA 114,192,168,169,0,108,5,0,756
                                 :rem 139
```

©

Atari Line Check Utility

Ed Sisul

"Atari Line Check" lets you use a joystick to perform a line-by-line search for program bugs.

Quite often, the most effective way to debug a program is to check each line, one at a time, for mistakes. For those of us who are not fortunate enough to own a printer, this can be a very tedious task. The lines can be examined using LIST and CTRL-1 to scroll through the program, but it is difficult to find minor mistakes while staring at a whole screen filled with GRAPHICS 0 text. The lines can be displayed one at a time using the sequence LIST *line number*, SHIFT CLEAR, LIST *line number*, SHIFT CLEAR, etc.; but this approach is too slow and cumbersome.

Scrolling With A Joystick

This program will step through a listing and display each line, one at a time, in large GRAPHICS 2 print. The best part is that the scrolling is controlled with a joystick. Pulling back on the stick advances through the listing, and pushing forward on the stick backtracks through the listing. With the stick centered, the displayed line stays on the screen for scrutiny. If a mistake is spotted, press the trigger button, and the line containing the mistake is redisplayed in the normal screen editing mode so it can be corrected. Once the error is dispatched, typing CONT will resume the line-by-line check, or typing RUN will terminate the line check and execute the main program. After typing in "Atari Line Check," LIST it to disk or cassette. Then, using the ENTER command, append it to the program to be checked. Plug a joystick into Port 1 and type GOTO 32000 to start checking lines.

Array Storage

The heart of the program is lines 32010–32030. Lines 32010–32020 retrieve the program line numbers stored in memory and store them in the array LINUM. A complete explanation of the PEEKs used to do this can be found in Larry Isaacs' article "Inside Atari BASIC" in *COMPUTE!'s First Book of Atari*. Line 32025 opens the screen editor for input and output, lists a line on the screen, then retrieves the entire line, including its line number, and stores it in the variable LINE$. The POKEs in line 32025 blank the screen during these operations. Line 32030 then reprints LINE$ on the screen in GRAPHICS 2 text in black letters on a white background.

Lines 32035–32055 contain the joystick controller routines to increment or decrement the subscript of the line number array or to redisplay a line for editing. Line 32000 initializes the variables, dimensions LINE$ to the maximum number of characters in a logical line, and dimensions the LINUM array to accommodate a 200-line program. The POKE in line 32000 standardizes the left-hand margin on all systems. Line 32005 initially sets all elements of the LINUM array to zero. Should you encounter a program with more than 200 lines, simply change the dimensioned size of LINUM in line 32000 and the maximum increment of the loop in line 32005 accordingly.

Storage Characters

Because each line is displayed in graphics mode 2, which uses the internal character set, some characters won't be displayed as originally typed. For instance, the special graphics characters will be displayed as numeric or punctuation symbols, and lowercase letters will be displayed as green uppercase letters. Also, the CLEAR character, CHR$(125), will cause the screen to clear when it is printed. When this happens, just press the trigger button to see the characters in their original form.

Atari Line Check

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

```
DM 32000 POKE 82,0:ST=0:Z
         Z=1:TRAP 32005:D
         IM LINE$(120),LI
         NUM(200):TRAP 40
         000
IC 32005 FOR N=O TO 200:L
         INUM(N)=0:NEXT N
ML 32010 AD=PEEK(136)+256
         *PEEK(137)
JB 32015 LINUM(ZZ)=PEEK(A
         D)+256*PEEK(AD+1
         ):IF LINUM(ZZ)=3
         2000 THEN END
OE 32020 IF LINUM(ZZ)=0 T
         HEN AD=AD+PEEK(A
         D+2):GOTO 32015
HL 32025 OPEN #1,13,0,"E:
         ":POKE 709,8: POK
         E 710,8:POKE 712
         ,8:LIST LINUM(ZZ
         ):POSITION 0,1:I
         NPUT #1;LINE$:CL
         OSE #1
BF 32030 GRAPHICS 18:POKE
         708,2:POKE 712,8
         :POSITION 0,2:?
         #6;LINE$
IC 32035 IF STRIG(0)=0 TH
         EN ST=1:GRAPHICS
          0:LIST LINUM(ZZ
         ):STOP
EP 32040 IF ST=1 THEN ST=
         0:GOTO 32025
MH 32045 IF STICK(0)=13 T
         HEN ZZ=ZZ+1:GOTO
          32020
MA 32050 IF STICK(0)=14 A
         ND ZZ>0 THEN ZZ=
         ZZ-1:GOTO 32025
DF 32055 GOTO 32035
```

©

Commodore Word Wizard

Joe W. Rocke

"Word Wizard" improves your writing skills by checking the readability of any written material. For the VIC-20, Commodore 64, and PET/CBM computers.

The term *foggy writing* was originated by Robert Gunning. Seeking ways to improve the readability of written text, he developed a *fog index* formula. The formula is based on counting the number of words and sentences in a sample paragraph of text. Long words and long sentences produce a high index number. This type of writing is called foggy because it can be harder to read and understand. Writing that is easy to read (and understand) should have a low fog index.

The fog index formula uses a 100- to 200-word sample of text. Words of three syllables or more are considered "long." Dividing the word count by the number of sentences provides the average sentence length. Adding the number of long words and performing a simple computation produce the fog index. Although the index number is rather arbitrary, it does provide a standard for measuring text readability.

Researchers have since learned that people prefer to read below their educational level. Thus the fog formula has been expanded to produce a reading level index number. The result is a number that represents the approximate grade level at which written material can be read and understood.

People are comfortable reading text that has a reader index ranging from 6 to 8. Most of the writing in popular magazines and newspapers has an index in this range. People are capable of reading at a higher level, but the concentration required can make such writing tedious. Even college professors find it uncomfortable to read something with an index of 12 or higher.

Computerized Word Check

The computer is an ideal tool for checking text for readability. Large companies have developed programs of this type to check their product manuals. When used with word processing systems, this checking process takes little additional time.

Using "Word Wizard" is as simple as typing text onto a video screen instead of on paper, as with a typewriter. A 100-word sample is all that is required. Almost all text-reading analysis is based on this sample size.

The program begins with a prompt. There is no cursor, but whatever is typed appears on the screen. The left arrow can be used to correct a typo without affecting the program. Use the RETURN key only when you are finished entering the sample. The screen then clears, and the text that has been typed to memory will begin to march across the screen. The text display will then be formatted to improve readability.

Type in the text sample without worrying how it looks on the video display. The text will wrap around the screen, causing some words to be broken midway and to continue on the next line. The display is primarily for reference so you can see what was originally typed.

The Display Phase

Next, during its display phase, the program counts characters, words, and sentences. It also counts the number of words containing more than nine characters, which are presumed to consist of three or more syllables. Word groups ending with a semicolon or colon are counted as one sentence. This prevents a compound sentence from being counted as a single sentence. Naturally, any word group ending with a period, question mark, or exclamation mark is counted as a sentence.

The word-checking data is stored in simple variables and is then used to compute the reading index at the end of the display cycle. A continuation prompt concludes the display cycle to permit you to read the last display page.

Finally the word, sentence, and long word counts are displayed. The reading index, rounded to two decimal places, completes the text analysis. The program then asks you to repeat the analysis or exit the program.

An index of 6–9 indicates a good readability level. A higher index indicates that the text might benefit from some editing. You may want to use two shorter sentences which carry the same thought as a long one, or try to find shorter words. For example, it is easier to read *city* than the word *metropolis*.

Variables

A$	The input string is confined to one character.
BE	Beginning address of the memory storage area.
C	ASCII value of A$, and the character counter.
C$	Character string used for the display cycle.
ID	Reading index. L is the display line length counter.
LW	Long word count storage.
MS	Memory storage ending address.
P	PEEK value of MS contents.
S	Sentence count storage.
T	Display cycle loop counter.
W	Word count storage.
WC	Input cycle word count.
Z & Z$	Prompts.

Housekeeping Chores

Lines 10–30: Housekeeping chores are performed at the beginning of the program. The formula used to round the reading index is defined in line 10. Major variables are set to zero to prevent errors if the program is rerun. Variable MS in line 20 denotes the beginning memory storage address. A second variable is set to the same value for use in the display loop.

The value currently in the program works with an unexpanded VIC-20. Use MS=2300 in line 20 if you have a PET/CBM or a 3K expanded VIC. (Ignore the color commands if you have a PET.) For a VIC with 8K or more of expansion memory, use MS=5900. Try MS=3300 for the Commodore 64. For other systems you will have to use an address above the BASIC program area.

Lines 35–150: The input cycle begins at line 60 with the GET A$ keyboard scan for a key input. When a key is pressed, the input is checked for a backspace (left cursor). If it is a backspace, the invisible cursor moves one space to the left, and the memory storage is decreased by one. This is to prevent counting the backspace as part of the text. The program then loops back for a new key input.

If the key pressed is a text character, the key is displayed and converted to its ASCII equivalent. The ASCII value is then POKEd in memory address MS for storage. The input is then tested for a carriage return (CR); if not a CR, storage address MS is incremented by one, and the program loops back for another key input. Note that a CR breaks the input loop, jumping program flow to the continuation GOSUB.

The Word Count

Line 110 performs a word count during the input cycle. The count value of 125 in line 120 limits input to a maximum of 125 words. These two lines are optional, but do insure keeping the input within sample limits. A smaller number of words can be used for a sample, of course.

Lines 160–300: The display and checking cycle begins upon user response to the continuation prompt. Variables used to accumulate word-checking data are set to zero to prevent errors if the program is repeated. A FOR-NEXT loop is used for the display cycle, since storage beginning address BE and ending address MS were established during the input cycle.

The stored ASCII data is PEEKed from each memory address, converted to a string, and temporarily stored in string variable C$ for display. C$ now represents the keyboard character entered during the input cycle. The individual characters are counted and the count is stored in C. L is used to count characters for line display formatting.

Word-checking functions are performed by IF statements. These lines check for the space character that denotes a word end, or punctuation indicating a sentence end. A space increments the word count, W. A sentence end increments the sentence count stored in S and decreases the character count by one. The decrease prevents the punctuation from being counted as a word character. If the character count in C is equal to or greater than 9, and a space indicates a word, then long word counter LW is incremented. The character counter is returned to zero value whenever a space or sentence end is encountered.

Screen Formatting

Line 220 formats the text to reduce word wraparound.

Lines 320–400: The text analysis is performed in this portion of the program. The reading index is computed in line 320. Text data accumulated during the word-check cycle are displayed, followed by the reading index (ID). The rounding function is performed by the FNA(ID) formula which was established at the beginning of the program.

Lines 410–480: The remaining lines contain the user prompts. Conventional INPUT statements are used to keep the program short. END is used between the REPEAT prompt and the continuation GOSUB to prevent an error message when exiting the program. Line 470 prints the word input count and returns control to the continuation prompt of line 150.

Word Wizard

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

```
5 REM... * WORD CHECK *                    :rem 145
10 DEF FNA(B)=INT(B*100+.5)/100             :rem 92
20 MS=5300: BE=MS                          :rem 165
30 C=0:L=0:LW=0:S=0:W=0:WC=0               :rem 137
35 REM...INPUT CYCLE                       :rem 214
50 PRINT"{CLR}{7}BEGIN INPUT ...":PRINT
                                           :rem 169
60 GETA$:IFA$=""THEN 60                    :rem 239
70 IFA$=CHR$(157)THEN PRINTA$;:MS=MS-1:GO
   TO60                                    :rem 209
80 PRINT A$;                               :rem 149
90 C=ASC(A$)                               :rem 118
100 POKE MS,C                              :rem 207
110 IFA$=" "THEN WC=WC+1                     :rem 3
120 IFWC=>125 THEN 470                     :rem 153
130 IFA$=CHR$(13)THEN 150                   :rem 64
140 MS=MS+1: GOTO 60                        :rem 71
150 GOSUB 440                              :rem 174
155 REM...DISPLAY CYCLE                    :rem 143
160 C=0:L=0:LW=0:S=0:W=1                   :rem 125
170 PRINT"{CLR}"                           :rem 252
180 FOR T=BE TO MS                         :rem 219
190 P=PEEK(T)                              :rem 241
200 C$=CHR$(P)                             :rem 216
210 C=C+1:L=L+1                             :rem 29
220 IFC$=" "AND L=>15THEN GOSUB 460
                                            :rem 84
230 PRINTC$;                               :rem 196
240 IFC$=" "THEN W=W+1:C=C-1               :rem 222
250 IFC$="."ORC$="!"ORC$="?"ORC$=":"ORC$=
    ";"THEN S=S+1:C=C-1:C$=" "              :rem 32
260 IFC$=" "ANDC=>9 THEN LW=LW+1 :rem 231
270 IFC$=" "THEN C=0                       :rem 239
280 IFC$=CHR$(13)THEN 310                   :rem 70
290 NEXT                                   :rem 218
300 PRINT                                   :rem 32
310 GOSUB 440                              :rem 172
315 REM...* ANALYSIS *                     :rem 191
320 ID=.4*(W/S+LW*100/W)                    :rem 36
330 PRINT"{CLR}"                           :rem 250
340 PRINTSPC(4)"** ANALYSIS **":PRINT
                                           :rem 166
350 PRINT"WORDS{2 SPACES}=";W              :rem 199
360 PRINT"SENTENCES{2 SPACES}=";S:rem 221
370 PRINT"AVG.WD/SENT =";INT(W/S)            :rem 8
380 PRINT"LONG WORDS{2 SPACES}=";LW
                                            :rem 70
390 PRINT                                   :rem 41
400 PRINT"READER INDEX =";FNA(ID):rem 187
410 PRINT:INPUT"REPEAT (Y/N)";Z$ :rem 209
415 IFZ$<>"N"ANDZ$<>"Y"THEN410             :rem 223
420 IFZ$="Y"GOTO 20                         :rem 24
430 PRINT"{BLU}{CLR}":END                   :rem 43
440 INPUT"PRESS <RETURN>";Z                :rem 232
450 RETURN                                 :rem 121
460 PRINTC$;CHR$(13):L=0:RETURN            :rem 140
470 PRINT:PRINT"WORDS INPUT=";WC           :rem 100
480 GOTO150                                :rem 107
```

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
```

CAPUTE!

Modifications Or Corrections To Previous Articles

Atari Super Directory

The character which appears as a grave (`) in lines 5010 and 5020 of this program from the April issue (p. 176) should actually be {.}, CTRL-period. You may find it easier to replace these lines with the lines below, which build M$ from DATA statements.

```
OH 5000 DIM M$(40):RESTORE 5040
NJ 5010 FOR I=1 TO 40:READ A:M$(I)=C
        HR$(A):NEXT I
KI 5030 RETURN
FG 5040 DATA 104,201,2,240,9,170,240
        ,5,104,104,202,208,251,96,10
        4,133,204,104,133,203,104
IK 5050 DATA 104,133,205,160,0,177,2
        03,9,128,145,203,200,196,205
        ,208,245,96,0,0
```

Roader For Atari And Color Computer

The Atari version of this game from the March issue (p. 66) may stop with an ERROR 141 message. To prevent this, Edward Rybczyk suggests the following corrections:

```
380 IF A=43 THEN CLR: RUN
390 POKE 764,255: END
```

The Color Computer version requires Extended BASIC to run as published. Ron Crail suggests changes to allow the program to run in standard Color BASIC: Change the value of XLOC to 304 in line 220 and to 308 in line 230, and change COS to SIN in lines 260 and 310. Also, adding the line 245 N$="X" will prevent an OS error.

VIC Barrier Battle

A testing loop was inadvertently left in line 200 of this game program from the March issue (p. 84). Troy Pibus points out that the line should read:

```
200 DD=37154:P1=37151:P2=37152
```

64 MLX And Trident

There is an error in the version of the "MLX" machine language editor from the March issue (p. 182). In line 765, K=S+1 should be replaced with K=S. This error will prevent the "Trident" game (p. 100), published in MLX format, from working properly. Fortunately, the problem is quite easy to fix. First, load and correct MLX and save the corrected version. Then run MLX and use the MLX Load option to load in Trident. Use the starting and ending addresses given in the Trident article. Retype the first line of Trident (49152), then use the MLX Save option to create a new copy of the game, which should now work properly.

Atari Trident

Reader Jim Davis suggests the following improvement to this game from the March issue (p. 94):

```
105 Z=USR(ADR(M$),M,M+1,128):FOR I=15 TO
    0 STEP -0.08:SOUND 0,10,8,I:NEXT I:Z
    =USR(ADR(A$),48+C,1,144,51)
```

This adds an explosion sound when an incoming missile is destroyed.

Commodore Floating Subroutines

Programs 1, 2, and 3 for this article from the March issue (p. 164) will print a range of hex address values which is one greater than the correct range, as shown in decimal. To correct this, Paul Montognese suggests changing the H=C in line 63994 to H=C–1.

Chopperoids

Some readers tried to create a binary file (MLX option F) for this Atari machine language program (December 1983, p. 122). As stated in the article, "Chopperoids" must be put on a boot disk or boot tape. If you made a binary file, follow these steps to create a boot disk from your work:

1. Load the MLX program and make the following *temporary* changes:

```
750 IF  NOT READ THEN 1040
850 TRAP 40000:CLOSE #2:? "Finished.
    ":LET READ=0:BUFFER$(FIN-BEG+31)
    =CHR$(0):BUFFER$(31)=BUFFER$(61)
    :GOTO 360
1000 H=INT(ADR(BUFFER$)/256):L=ADR(B
     UFFER$)-H*256:L=L+30:POKE ICBAD
     R+X,L:POKE ICBADR+X+1,H
```

2. Run the modified MLX and use the addresses given in the original article. Specify the boot disk option.

3. Use the MLX Load command to load your binary file. All the data will be moved up five lines, as described in the February "CAPUTE!" corrections.

4. Use the MLX New Address command to begin typing at line 6092 and enter the additional lines from February "CAPUTE!" (p. 181). Insert a new disk in the drive and use the MLX Save option to create a boot disk with the corrected data.

MLX Machine Language Entry Program For Commodore 64

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. *You need to know nothing about machine language to use MLX—it was designed for everyone.*

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file. You can then use the LOAD command to read the program into the computer:

LOAD "filename",1,1 (for tape)
LOAD "filename",8,1 (for disk)

To start the program, you enter a SYS command that transfers control from BASIC to machine language. The starting SYS number appears in the article.

Using MLX

Type in and save MLX for your 64 (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

You'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the comma, SPACE bar, or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

	U	I	O		7	8	9
H	J	K	L	become 0	4	5	6
	M	,	.		1	2	3

MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

SHIFT-S: Save
SHIFT-L: Load
SHIFT-N: New Address
SHIFT-D: Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

What if you forgot where you stopped typing? Use the Display command to scan memory from the beginning to the end of the program. When you reach the end of your typing, the lines will contain a random pattern of numbers. When you see the end of your typing, press any key to stop the listing. Use the New Address command to continue typing from the proper location.

MLX: Machine Language Entry

```
10 REM LINES CHANGED FROM MLX VERSION 2.0
   0 ARE 750,765,770 AND 860        :rem 50
100 PRINT"{CLR}[6]";CHR$(142);CHR$(8);:
    POKE53281,1:POKE53280,1         :rem 67
101 POKE 788,52:REM DISABLE RUN/STOP
                                    :rem 119
110 PRINT"{RVS}{39 SPACES}";        :rem 176
120 PRINT"{RVS}{14 SPACES}{RIGHT}{OFF}
    [*]£{RVS}{RIGHT} {RIGHT}{2 SPACES}
```

```
     [*]{OFF}[*]£{RVS}£{RVS}
     {14 SPACES}";                         :rem 250
130 PRINT"{RVS}{14 SPACES}{RIGHT} [G]
     {RIGHT} {2 RIGHT} {OFF}£{RVS}£[*]
     {OFF}[*]{RVS}{14 SPACES}";            :rem 35
140 PRINT"{RVS}{41 SPACES}"                :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} MACHINE LANG
     UAGE EDITOR VERSION 2.01{5 DOWN}"
                                           :rem 237
210 PRINT"[5]{2 UP}STARTING ADDRESS?
     {8 SPACES}{9 LEFT}";                  :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+119*F)
                                           :rem 166
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
     THENGOSUB3000:GOTO210                 :rem 235
225 PRINT:PRINT:PRINT                      :rem 180
230 PRINT"[5]{2 UP}ENDING ADDRESS?
     {8 SPACES}{9 LEFT}";:INPUTE:F=1-F:C$=
     CHR$(31+119*F)                        :rem 20
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
     THENGOSUB3000:GOTO230                 :rem 183
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
     {2 SPACES}":GOSUB1000:GOTO 230
                                           :rem 176
260 PRINT:PRINT:PRINT                      :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                           :rem 225
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),
     2),5);":";                            :rem 33
315 FORJ=ATO6                              :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                           :rem 228
390 IFN=-211THEN 710                       :rem 62
400 IFN=-204THEN 790                       :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
     EW ADDRESS";ZZ                        :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
     {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                           :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                           :rem 238
420 IF N<>-196 THEN 480                    :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
     ";:INPUTT                             :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
     T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
     0                                     :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
     0"+MID$(STR$(I),2),5);":";            :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
     "+MID$(STR$(N),2),3);",";             :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                           :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
     T:GOTO310                             :rem 50
480 IFN<0 THEN PRINT:GOTO310               :rem 168
490 A(J)=N:NEXTJ                           :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
     SUM=(CKSUM+A(I))AND255:NEXT           :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
     );                                    :rem 94
511 IFN=-1THENA=6:GOTO315                  :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
                                           :rem 122
520 PRINT:PRINT"LINE ENTERED WRONG : RE-E
     NTER":PRINT:GOSUB1000:GOTO310:rem 176
530 GOSUB2000                              :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
     272,0:POKE54273,0                     :rem 227
550 AD=AD+6:IF AD<E THEN 310               :rem 212
560 GOTO 710                               :rem 108
570 N=0:Z=0                                :rem 88
580 PRINT"[£]";                            :rem 81
581 GETA$:IFA$=""THEN581                   :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*(A$=".")-4*
     (A$="J")-5*(A$="K")-6*(A$="L"):rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
     ):IFA$="H"THENA$="0"                  :rem 134
584 IFAV>0THENA$=CHR$(48+AV)               :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
     ORA=32THEN670                         :rem 229
590 IFA>128THENN=-A:RETURN                 :rem 137
600 IFA<>20 THEN 630                       :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
     {OFF}{LEFT} {LEFT}";:GOTO690          :rem 62
620 GOTO570                                :rem 109
630 IFA<48ORA>57THEN580                    :rem 105
640 PRINTA$;:N=N*10+A-48                   :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                           :rem 229
660 Z=Z+1:IFZ<3THEN580                     :rem 71
670 IFZ=0THENGOSUB1000:GOTO570             :rem 114
680 PRINT",";:RETURN                       :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                           :rem 149
691 FORI=1TO3:T=PEEK(S%-I)                 :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                           :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                           :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                           :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF}
     ALONE TO CANCEL SAVE){DOWN}":rem 106
720 F$="":INPUT"{DOWN} FILENAME";F$:IFF$=
     ""THENPRINT:PRINT:GOTO310             :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"                :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
                                           :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$:
     OPEN15,8,15,"S"+F$:CLOSE15            :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
     ):POKE782,ZK/256                      :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
     T$):SYS65469                          :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
     66                                    :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
     *256:POKE780,253                      :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
     2)*256:SYS65496                       :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
                                           :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":GOTO310
                                           :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
     RY AGAIN.":IFDV=1THEN720              :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
     ;E2$:CLOSE15:GOTO720                  :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                           :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF}
     ALONE TO CANCEL LOAD)"                :rem 82
800 F$="":INPUT"{2 DOWN} FILENAME";F$:IFF
     $=""THENPRINT:GOTO310                 :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"                :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
                                           :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                           :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
     ):POKE782,ZK/256                      :rem 2
```

```
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                  :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                            :rem 70
850 POKE780,0:SYS65493            :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
                                  :rem 111
865 PRINT"{DOWN}DONE.":GOTO310    :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
                                  :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800          :rem 102
1000 REM BUZZER                   :rem 135
1001 POKE54296,15:POKE54277,45:POKE54278,
     165                          :rem 207
1002 POKE54276,33:POKE 54273,6:POKE54272,
     5                            :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
     273,0:POKE54272,0:RETURN     :rem 202
2000 REM BELL SOUND               :rem 78
2001 POKE54296,15:POKE54277,0:POKE54278,2
     47                           :rem 152
2002 POKE 54276,17:POKE54273,40:POKE54272
     ,0                           :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
                                  :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1000                     :rem 89
```

©

NEWS&PRODUCTS

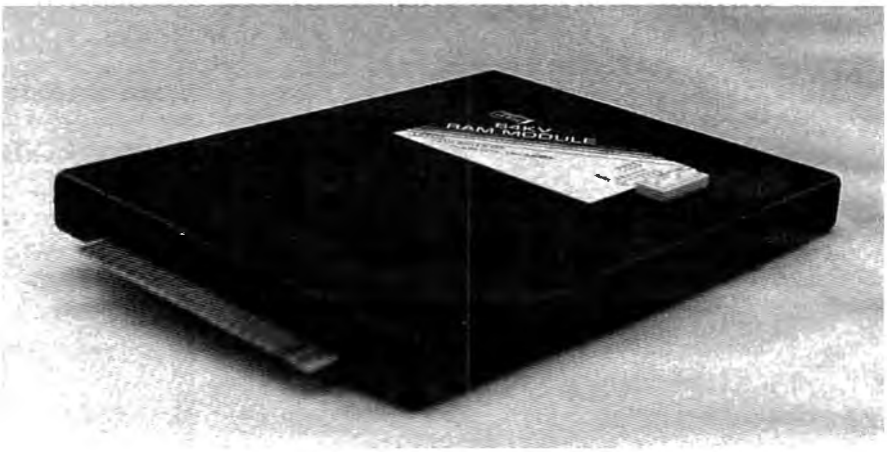
Memory Expander For VIC-20

Letco has announced the 64KV Memory Module, which adds more than 64K of memory to your VIC-20.

The 64KV houses 8K in each of the VIC's blocks 1, 2, and 3. Block 3 can also be paged, or swapped, under program control, with five other separate 8K sections of memory. Each block has a separate enable switch and a write-protect switch, and there is a switch to make block 3 respond as though it is block 5 (the normal game block).

The module is priced at $109.95

Letco
7310 Wells Road
Plain City, OH 43064
(614) 873-4410



The Letco 64KV Memory Module adds more than 64K RAM memory to your VIC-20.

Authoring System And Teaching Tool

CLAS, a teaching tool and authoring system for educators, has been released by Touch Technologies for the Apple II+ and IIe, the IBM PC and PCjr, and the Commodore 64.

The software package functions as a teaching tool for any subject. Authoring procedures allow instructors to create lessons in their own teaching style. Up to 30 problem sets can be offered with each lesson. Questions take the form of true/false, multiple choice, short answer, or matching.

If desired, the questions can be presented in a different order each time the lesson is used.

Sound is used to give feedback when a response is made to a question. A help mode is provided for the student, along with a review of problem areas and a summary of performance at the end of the lesson.

Memory requirement for Apple computers is 48K. The IBMs must use DOS 2.0/2.1. *CLAS* is available for $89.95.

Touch Technologies
609 S. Escondido Blvd.
Ste. 101
Escondido, CA 92025
(619) 743-0494

Interface For TI-99/4A

Mikel Laboratories, Inc., has announced an RS-232-C interface system for the TI-99/4A.

The $145.95 system is a free-standing unit which allows the TI-99/4A to use a printer and modem without a peripheral expansion unit.

The company also offers cassette interface systems ($49.95), TI cassette cables ($11.95), and printers and monitors. A line of personal computer accessories for the TI-99/4A will soon be available from Mikel Laboratories.

Mikel Laboratories
3341 W. El Segundo Blvd.
Hawthorne, CA 90250
(213) 679-2542

Life Insurance Program For Atari, Commodore

Advanced Financial Planning has released *Life Insurance Planning*, a software package for the Atari 400 and 800 computers and the Commodore 64.

The program will calculate the inflation rate applicable to a user's budget; the user's total estate needs reduced into terms of today's dollars (such as future living expenses for the family, college expenses, and funeral expenses); the total estate provided by all sources of income and assets; and the total shortfall needed to be provided by life insurance.

Life insurance needs can be calculated for any year over the planning period in order to help the user select the proper type of insurance policy.

Life Insurance Planning supports virtually any printer, and requires a disk drive. The Atari version requires the Atari BASIC cartridge and 32K RAM. The package is priced at $29.95. When purchased with Advanced Financial Planning's *Retirement Planning* program, the total price is $49.95 (shipping prices are included in this total).

Advanced Financial Planning
20922 Paseo Olma
El Toro, CA 92630
(714) 855-1578

Music Adventure Games For Apple II

Syntauri Corporation has introduced Musicland, an advanced set of musical games for the Apple II.

The package is built from four basic games—*Sound Factory*, *Timbre Painting*, *Music Doodles*, and *Music Blocks*. The four games are integrated. Musicland is a foundation program from which advanced musical concepts and structures may be taught to young children.

The system attempts to maintain the interest of young students, while providing musical challenges for adult musicians as well. Aimed at musically untrained children, Musicland lets youngsters use joystick controls to discover musical form, timbre, orchestration, composition, and transposition.

Children can compose, edit, and play music as well. Interactive graphics aid exploration, from sketching a simple melody to inverting a complex musical passage. Multipart pieces can be composed, orchestrated, and played back in stereo. Each of the four games covers a different range of musical learning experiences.

Musicland requires a 64K Apple II computer system with one disk drive, plus synthesizers (the Mountain Computer Music-System) which plug into the Apple. The Musicland set with manuals sells for $150. The synthesizers are available for under $400.

Syntauri Corporation
4962 El Camino Real
Suite 112
Los Altos, CA 94022
(415) 966-1273

Three Learning Programs For Atari, Commodore

Three learning programs from Carousel Software have been released for the Commodore 64 and Atari computers on disk or cassette.

Telly Turtle is an introduction to computer programming which uses drawing routines and emphasizes logical thinking, problem solving, numbers sequencing, and visual discrimination.

Brain Strainers includes three learning games for from one to four players: *Clef Climber*, a multilevel, animated note recognition game; *Finders Keepers*, a multiscreen and multilevel concentration game; and *Follow the Leader*, a music and graphic pattern recognition game with up to 44 levels of difficulty.

Simulated Computer is an animated simulation of a computer in operation. Programs written by the user can be seen and heard flowing through the component parts of the computer. The program serves as a teaching tool about the way a computer works.

Telly Turtle (34.95) and *Brain Strainers* ($29.95) are meant for ages five to adult. *Simulated Computer* ($29.95) is directed toward ages 12 to adult.

Carousel Software, Inc.
877 Beacon Street
Boston, MA 02215
(617) 437-9419

Games, Tutorial For Commodore 64

Advanced Microware has introduced two new software products for the Commodore 64.

Casino Pac includes four games—Blackjack, Poker, Keno, and Slot Machine. Each simulates the new videogaming machines being used in gambling centers such as Las Vegas and Atlantic City. The games let you practice your betting strategy, try your own betting systems, or play for fun.

Casino Pac sells on tape or disk for $39.

64Tour is a tour of the features and capabilities of the Commodore 64, with demonstrations of all the graphics modes, as well as music and sound effects. The package is priced at $12.

Advanced Microware
P.O. Box 6143
Santa Ana, CA 92706
(714) 554-6470

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.*

COMPUTER MAIL ORDER

NEC PRINTERS
NEC 2050 .... $999.00
NEC 3550 .... $1679.00

PERCOM/TANDON DRIVES
5¼ 320K Floppy $229.00
5 Meg Hard w/Controller .... CALL
10 Meg Hard w/Controller .... CALL
15 Meg Hard w/Controller .... CALL
20 Meg Hard w/Controller .... CALL

AMDEK
310A Amber Monitor $169.00
DXY 100 Plotter $599.00
Color II $399.00

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

MICROPRO
WordStar/MailMerge $349.00
InfoStar $299.00
SpellStar $159.00
CalcStar $99.00

MICROSTUF
Crosstalk $105.00

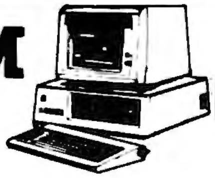
MICROSOFT
Multiplan $159.00

ASHTON TATE
dBASE II $389.00
Friday! $185.00

IUS
EasyWriter II $249.00
EasySpeller $119.00
EasyFiler $229.00

CONTINENTAL SOFTWARE
1st Class Mail/Form Letter...$79.00
The Home Accnt. Plus $88.00

IBM



VISICORP

	IBM	APPLE
VisiCalc		159.00
VisiCalc 4	159.00	
VisiCalc-Advanced		269.00
VisiWord/Spell	249.00	
Visitrend/Plot	199.00	199.00
VisiLink		169.00
VisiFile	199.00	169.00
VisiSchedule	199.00	199.00
Visidex		159.00
VisiPlot		135.00
VisiTerm		75.00
Desktop Plan	199.00	169.00
Bus. Forecast Model	75.00	75.00
Stretch Calc	75.00	75.00
VisiTutor Calc	59.00	59.00
VisiTutor-Advanced	75.00	75.00
VisiTutor Word	59.00	59.00
Vision Calc	249.00	
Vision Graph	129.00	
Vision Mouse	159.00	
Vision Host	319.00	

pfs

	APPLE	IBM
Write:	79.00	89.00
Graph:	79.00	89.00
Report:	79.00	79.00
File:	79.00	89.00
Solutions*: as low as	16.00	16.00

*Call On Titles

LOTUS
1-2-3 .... $329.00

PROFESSIONAL SOFTWARE
PC Plus/The Boss $349.00

SYNAPSE
File Manager $89.00

MBC-550 PC ... CALL
MBC-555 PC ... 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

APPLE/FRANKLIN DISK DRIVES
MICRO-SCI
A2 .... $219.00
A40 .... $299.00
A70 .... $319.00
C2 Controller $79.00
C47 Controller $89.00

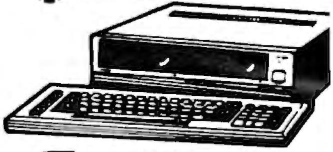
RANA
Elite 1 .... $279.00
Elite 2 .... $389.00
Elite 3 .... $569.00

APPLE IIe STARTER PACK
64K Apple IIe. Disk Drive & Controller. 80 Column Card. Monitor II & DOS 3.3
COMPLETE .... CALL

TERMINALS
914 .... $569.00
924 .... $689.00
925 .... $739.00
950 .... $929.00
970 .... $1039.00

COMPUTERS
Teleport Portable .... CALL
800A $1099.00
802 .... $2699.00
803 .... $1949.00
802H $4695.00
806/20 .... $4999.00
816/40 .... $9199.00
1602 .... $3399.00
1603 .... CALL

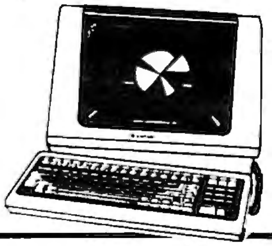
SANYO



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"

TeleVideo



PRINTERS

AXIOM
AT-100 Atari Interface $239.00
GP-100 Parallel Interface $199.00
AT-846 Interface $89.00

AXIOM 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
8600P $929.00
F10-40 .... $999.00
F10-55 .... $1499.00

COMREX
ComWriter II Letter Quality ... $499.00

DIABLO
620 Letter Quality $949.00
630 Letter Quality $1749.00

DAISYWRITER
2000 .... $999.00
Tractor Feed $109.00

EPSON
MX-80FT, MX-100, RX-80, RX-80FT, FX-80, FX-100 .... CALL

IDS
Prism 80...For Configurations ... CALL
Prism 32...For Configuration ... CALL

MANNESMAN TALLY
160L $589.00
180L $799.00
Spirit 80 .... $309.00

NEC
2010/2030 .... $899.00
8023 Dot Matrix $379.00
8025 Dot Matrix $669.00
3510 Serial/Letter Quality ... $1449.00
3530 Parallel/Letter Quality ... $1499.00
7710/7730 Serial/Parallel ... $1949.00

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

SMITH CORONA
TP-2 .... $399.00
Tractor Feed $119.00

SILVER REED
500 Letter Quality $469.00
550 Letter Quality $699.00

STAR
Gemini 10X $299.00
Gemini P15X $399.00
Delta 10 .... $559.00
Serial Board $75.00

TOSHIBA
1350 .... $1599.00
TRANSTAR .... CALL

PRINTER CABLES
Available for Atari, Commodore, IBM, Apple, Epson, Kaypro, Televideo, Franklin, Eagle, Sanyo, Osborne, NEC, Zenith and many others. We supply all your computer needs!

PAPER SUPPLIES
1000 shts. 8½x11 Tractor Paper ... 19.99
1000 shts. 14½x11 Tractor Paper .. $24.99
1 or 2" Address Labels $9.99

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 .... $399.00
Color 2 Plus $419.00
Color 3 .... $349.00
Color 4 .... $699.00

BMC
12" Green $88.99
12" Green HI-RES $119.99
9191-13" Color $249.99

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
JC 1215 Color $299.99
JC 1216 RGB $429.00
JC 1460 Color RGB $359.00

PRINCETON GRAPHICS
HX-12 RGB $519.00

SAKATA
100 .... $269.00

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

USI
Pi 1, 9" Green $99.99
Pi 2, 12" Green $119.99
Pi 3, 12" Amber $149.99
Pi 4, 9" Amber $139.99
1400 Color $269.99

QUADRAM
Quadchrome 8400 .... $579.00

ZENITH
ZVM 122 Amber $109.00
ZVM 123 Green $89.99
ZVM 135 Color/RGB $469.99

MODEMS

ANCHOR
Mark I (RS-232) $79.00
Mark II (Atari) $79.00
Mark III (TI-99) $109.00
Mark IV (CBM/PET) $125.00
Mark V (Osborne) $95.00
Mark VI (IBM-PC) $169.00
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 .... $219.00
Smartmodem 1200 .... $509.00
Smartmodem 1200B $459.00
Micromodem II $265.00
Micromodem II Plus $299.00
Micromodem IIE $269.00
Micromodem 100 .... $299.00
Smart Com II $89.00
Chronograph $199.00

NOVATION
J-Cat $99.99
SmartCat 103 .... $179.00
SmartCat 103/212 .... $399.00
AutoCat $219.00
212 AutoCat $549.00
Apple Cat II $249.00
212 Apple Cat $569.00
Apple Cat 212 Upgrade $309.00
Cat $139.99
D-Cat $149.00
PC-Cat $399.00

ZENITH
ZT 1 .... $309.00
ZT 10 .... $339.00
ZT 11 .... $369.00

APPLE INTERFACE CARDS & BUFFERS
Choose from PKASO, Orange Micro, MPC, MicroMax, Tymac, Quadram & Practical Peripherals .... CALL

WEST — 800-648-3311
In NV call (702)588-5654, Dept. 0506
Order Status Number: 588-5654
P.O. Box 6689, Stateline, NV 89449

CANADA — 800-268-4559
In Toronto call (416)828-0866, Dept. 0506
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 1B
Mississauga, Ontario, Canada L5L1T1

EAST — 800-233-8950
In PA call (717)327-9575, Dept. 0506
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. Third St., Williamsport, PA 17701

VISA

No risk, no deposit on C.O.D. orders. Pre-paid orders receive free shipping within the UPS Continental United States with no waiting period for certified checks or money orders. Add 3% (minimum $5.00) shipping and handling on all C.O.D. and credit card orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability and price change. We stock manufacturer's and third party software for most all computers on the market. Call today for our new catalog.

COMPUTER MAIL ORDER

Koala Technologies Corporation

KOALA PADS

Atari (Disk)	$75.00
Atari (ROM)	$82.00
C-64 (DISK)	$75.00
C-64 (ROM)	$82.00
IBM	$99.00
AP/Franklin	$85.00

commodore

CBM 8023...$599

MSD SD1 Disk Drive	$349.00
MSD SD2 Disk Drive	$599.00
CBM 4032	$599.00
CBM 8096	$869.00
CBM 9000	$999.00
B128-80	$769.00
CBM 64K Memory Board	$269.00
8032 to 9000 Upgrade	$269.00
2031LP Disk Drive	$299.00
8050 Disk Drive	$949.00
8250 Disk Drive	$1199.00
4023 Printer	$379.00
8023 Printer	$569.00
6400 Printer	$1399.00
Z-RAM	$499.00
Silicon Office	$699.00
The Manager	$199.00
Soft ROM	$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

SX-64 PORTABLE $839.00

VIC 20 .... CALL

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
Simon's Basic	$19.00
Word Pro 64 Plus	$59.00
Parallel Printer Interface	$49.00
Calc Result 64	$129.00
Codewriter 64	$75.00
Quick Brown Fox	$49.00
Word Pro 64 Plus	$59.00
MCS 801 Color Printer	$499.00
DPS 1101 Daisy Printer	$459.00
Magic Voice Speech Module	$54.00
Desk Organizer Lock	$49.00

We stock a full inventory of software for Commodore, such as: Artworx, Broderbund, Commercial Data, Creative Software, Epyx, HES, MicroSpec, Nufekop, Romox, Sirius, Synapse, Thorn EMI, Tronix, UMI, Victory, Spinnaker, Rainbow & Timeworks!

ATARISOFT

Games for IBM, Apple, C64 & VIC 20

	IBM/Apple	C64/VIC 20
Pac Man	29.99	37.99
Centipede	29.99	37.99
Dig Dug	29.99	37.99
Donkey Kong	29.99	37.99
Defender	29.99	37.99
Robotron	29.99	37.99
Star Gate	29.99	37.99

CMO'S PORTABLE CORNER

HP 71B $499.99

41 CV........$129.99

41 CX........$199.99

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 Cass. or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

TIMEX/SINCLAIR

Timex/Sinclair 1000	$24.99
Timex/Sinclair 2086	CALL
16K Memory	$25.00
2040 Printer	$99.99
VuCalc	$17.99
Mindware Printer	$99.99

PC-8201 $599

NEC

PC-8221A Thermal Printer	$149.00
PC-8281A Data Recorder	$99.00
PC-8201-06 8K RAM Chips	$105.00
PC-8206A 32K RAM Cartridge	$329.00

SHARP

PC-1500A...$165.99

PC-1250A....$88.99

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

ATARI HOME COMPUTERS

600XL $99*

*With Purchase of 1010 Program Recorder, E.T. Phone Home & C1311 Joysticks

TOTAL SYSTEM PRICE $259

1010 Recorder	$74.00
1020 Color Printer	$249.00
1025 Dot Matrix Printer	$339.00
1027 Letter Quality	$309.00
1030 Direct Connect Modem	$119.00
1050 Disk Drive	$339.00
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 Assorted Education	$47.00
4011 Star Raiders	$33.00
4012 Missile Command	$29.00
4013 Asteroids	$29.00
5049 VisiCalc	$159.00
7097 Logo	$79.00
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

600XL......$189

800XL......$299

1200XL ....CALL

1400XL ....CALL

PERCOM

AT 88-S1	$369.00
AT 88-A1	$259.00
AT 88-S1PD	$449.00
AT 88-DDA	$119.00
RFD 40-S1	$449.00
RFD 40-A1	$269.00
RFD 40-S2	$699.00
RFD 44-S1	$539.00
RFD 44-S2	$869.00

TEXAS INSTRUMENTS

TX 99-S1	$279.00

RANA

1000	$329.00

TRAK

AT-D2	$389.00

INDUS

GT-Drive	$379.00

MEMORY BOARDS

Axlon 32K	$59.00
Axlon 48K	$99.00
Axlon 128K	$299.00
Intec 32K	$59.00
Intec 48K	$85.00
Intec 64K	$99.00
Intec Real Time Clock	$29.00

DISKETTES

MAXELL

5¼ MD-1	$29.00
5¼ MD-2	$39.00
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

ELEPHANT

5¼" SS/SD	$18.49
5¼" SS/DD	$22.99
5¼" DS/DD	$28.99

HEAD

5¼ Disk Head Cleaner	$14.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-n-File 10	$3.99
Flip-n-File 50	$17.99
Flip-n-File (400/800 ROM) Holder	$17.99

LJK ENTERPRISES

Atari Letter Perfect-Disk (40/80)	$79.99
Atari Letter Perfect-ROM (40 col)	$79.99
Atari Letter Perfect-ROM (80 col)	$79.99
Atari Data Perfect-ROM (80 col)	$79.99
Atari Spell Perfect-DISK	$59.99
Atari Utility/MailMerge	$21.00
Apple Letter Perfect	$99.00
Apple Data Perfect	$75.00
Apple LJK Utility	$21.00
Apple Lower Case Generator	$19.00

ALIEN VOICE BOX

Atari	$119.00
Apple	$149.00

CONTROLLERS & JOYSTICKS

WICO

Joystick	$21.99
3-way Joystick	$22.99
Famous Red Ball	$23.99
Power Grip	$21.99
BOSS Joystick	$17.99
ATARI/VIC Trak Ball	$34.99
Apple Trak Ball	$54.99
Apple Adapter	$15.99
Apple Analog	$37.99

KRAFT

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

AMIGA

3100 Single	$13.99
3101 Pair	$19.99
Joyboard	$37.99

TG

Atari Trak Ball	$47.99
Apple Joystick	$47.99
Apple Trak Ball	$47.99

WEST 800-648-3311

In NV call (702)588-5654 Dept. 0506
Order Status Number: 588-5654
P.O. Box 6689 Stateline, NV 89449

CANADA 800-268-4559

In Toronto call (416)828-0866 Dept. 0506
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 1B
Mississauga, Ontario, Canada L5L1T1

EAST 800-233-8950

In PA call (717)327-9575 Dept. 0506
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. Third St. Williamsport, PA 17701

MasterCard

CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada.
INTERNATIONAL ORDERS: All shipments outside the Continental United States must pre-pay by certified check only. Include 3% (minimum $5.00) shipping and handling. EDUCATIONAL DISCOUNTS: Additional discounts are available to qualified Educational Instutions.
APO & FPO: Add 3% (minimum $5.00) shipping and handling.

Lyco Computer Marketing & Consultants

TO ORDER CALL US — **TOLL FREE 800-233-8760**

In PA 1-717-327-1824

PERCOM

FOR ATARI COMPUTERS

AT88S1 ....$299.00
AT88S2 ....$535.00
AT**S1PD...$439.00
RFD40S1...$399.00
RFD40S2...$675.00
RFD44S1...$449.00
AT88 doubler

HARD DISK DRIVES for APPLE IBM-PC

5MEG.....$1349.00
10MEG ...$1599.00
15MEG ...$1999.00
20MEG ...$2359.00

*Add $ 80.00 for TRS 80 D. ves

TEXAS INSTRUMENT

Disk Drive...$245.00

TRAK DISK DRIVES

AT-D1 ....$379.00
AT-D2 ....$399.00
PRINTER CABLE $22.95
Software for ATD-2 ....$22.95

RANA DISK DRIVE

COMPUTER CARE

BIB
5¼ DISK DRIVE CLEANER............$12.75
COMPUTER CARE KIT $19.75

BLANK DISKETTES

ELEPHANT

Single Side SD (10).......$17.75
Single Side DD (10).......$21.75
Double Side DD (10)......$26.75

MAXELL

MD I (10)...................$28.75
MD II (10) $38.75

CERTRON CASSETTES

CC-10 12 for $15.99
CC-20 12 for $17.99

INNOVATIVE CONCEPTS

Disk Storage (holds 10)....$4.95
Disk Storage (holds 15)....$9.95
Disk Storage (holds 50)....$26.95

apple®

SSI

Battle of Shilo $26.75
Tigers in the Snow.... $26.75
Cosmic Balance $26.75
Knights of the Desert . $26.75
Battle for Normandy .. $26.75
Germany 1985 .... $36.75

RANA DISK DRIVES

Elite 1.............$295.00
Elite 2.............$449.00
Elite 3.............$559.00

MICRO-SCI

A2 ....call
A40.........................call
A70.........................call

MUSE

Castle Wolfenstein $20.75
Caverns of Frietag $20.75
Robot War.............$26.75

CONTINENTAL

Home Accountant $51.75
Book of Apple Software . $16.75

BRODERBUND

Bank Street Writer $49.75
AE $24.75
LODE RUNNER D $24.75
Choplifter $24.75
David's Midnight......... $24.75

SPINNAKER

Kindercomp................ $21.75
Story Machine $23.75
FaceMaker $23.75
Snooper Trooper......... $29.75
Delta Drawing $34.75

EPYX

Temple of Apshai..........$26.95
Star Warrior $26.95
Crush, Crumble & Chomp . $22.75

ADVENTURE

Saga#1 Adventureland ...$29.95
Saga#2 Pirate Adventure.$29.95
Saga#3 Secret Mission...$29.95
Stone of Sisyphus $24.95

ALIEN GROUP

Atari Voice Box $99.00
Apple Voice Box......... $129.00

commodore

1212 Programmers Ad.$44.75
1213 Vicmon...........$44.75
Vic 20 dust cover........$6.99
Vic 64 dust cover....... $6.99

TIMEWORKS

INVENTORY..............$59.75
ACCOUNTS REC. $59.75
ACCOUNTS PAY. $59.75
GENERAL LEDGER.......$59.75
PAYROLL..................$59.75
CASH FLOW $59.75
SALES ANALYSIS.........$59.75
ELEC. CHECKBOOK......$59.75
MONEY MANAGER $59.75
DATA MANAGER..........$59.75
WALL STREET $59.75

HES 64

Sound Box $9.95
64Forth $55.75
Hesmon................$25.75
Turtle Graphics........$37.75
Heswriter $28.75
Gridrunner.............$19.75
Attack of M C $22.75
Turtle Trainer..........$22.75
Turtle Tutor............$22.75
Paint Brush $22.75
Benji $25.75
Home Manager $28.75
Time Money Mgr $44.97
OmniCalc $33.75
Sword Point.............$19.95
Hes Modem.............$52.75

CARDCO

Cardprinter / LQ1....... $499.00
Cardprint DM1........... $109.00
5 Slot Expansion 64 ..... $54.00
64 Write NOW............ $39.00
64 Mail NOW............. $29.00
20 Write NOW............ $29.00
64 Keypad $29.00
Universal Cass. Int....... $29.75
Printer Utility............. $19.75
6 Slot Expansion......... $79.95
3 Slot Expansion......... $24 95
PRINTER INTERFACE........ $39.75
PRINTER INTERFACE with full graphics $65.75
LIGHT PEN $29.75

PARKER 20

Frogger (ROM)...........$33.75
QBert (ROM) $33 75
Tutankham (rom) $33 75

SPINNAKER 64

Kindercomp................$21.75
Story Machine $23.75
Face Maker $23.75
Snooper Trooper...........$29.75
Delta Drawing...............$34.75
Shamus II c/d.............. $24.95
Pinhead c/d $22.95

QUICK BROWN FOX

QBF Word Processor $49.95

LJK

Letter Perfect............ $105.00
Data Perfect..... .... $95.00

ADVENTURE INTERNATIONAL

S. Adams Adventure..... $28.75

VIC-64

Household Finance C/D .. $24.75

VIC 20

King Arthurs Heir Cass $24.75
Monster Maze Rom $24 75

ATARI — Computers for people.

600XL ...$CALL
800XL.........for
1400XL...Lowest
1450 .....Prices

1020 PRINTER...........NOW
1025 PRINTER..............IN
1027 PRINTER ....STOCK
1050 DISK DRIVE ....$SAVES
1010 RECORDER......$74.75

SYNAPSE

BLUE MAX C/D $24.75
Ft. APOCALYPSE C/D...$24.75
PHAROAH'S CURSE C/D...$24.75

FIRST STAR

ASTRO CHASE C/D....$22.75
BRISTOLS C/D.........$22.75
FLIP FLOP C/D $22.75

ALIEN GROUP

Voice Box 2 . $99.75

DON'T ASK

Sam $41.75
Abuse $15.95
Teleatri......$27.95
Poker Sam .. $24.95

APX

3R Math $19.95
Typo Attack $24.95
Family Budget $19.95
F. Cash Flow................ $19.95

BRODERBUND

Bank Street Writer D...... $49.75
AE D...................... $24.75
Apple Panic D............ $23.75
Choplifter ROM $32.75
David's Midnight..........$24.75
Stellar Shuttle C/D....... $18.75
Ft. Apocalypse........... $24.75

EPYX

GATEWAY TO ASPHI R..............$28.75
JUMPMAN JR R $28.75
PIT STOP R $28.75
GATEWAY TO

SSI

Battle of Shilo C/D.. ......$26.75
Tigers in the Snow C/D....$26.75
Battle for Normandy C/D ..$26.75
Knights of the Desert C/D .$26.75
Cosmic Balance C/D $26.75

ON-LINE

Frogger $24.95
Wizard & Prin $26.95

ROKLAN

Wizard of War.............$29.75
Gorf $29.75
Delux Invader $27.95

BIG 5

Miner 2049................ $32.75

EASTERN HOUSE

Monkey Wrench 2 ....$52.75

BRODERBUND

LODE RUNNER D......$24.75
OPERATION WIRLWIND D $29.75
DROL D $24.75

PARKER BROTHERS

Tutankham R............$33.75
Super Cobra R $33.75
Astro Chase R...........$33.75
Frogger R $33.75
QBert R $33.75
Popeye R.................$33.75
Risk R $42.75
Chess R $42.75

SPINNAKER

Story Machine R $26.75
Face Maker R...........$24.75
Kinderomp R............$20.75
Fraction Fever R $24.75
Delta Drawing R.........$26.75

YOU CAN'T TELL A DISK DRIVE BY ITS COVER!!

WITH A **HAPPY ENHANCEMENT** INSTALLED THESE ARE THE MOST POWERFUL DISK DRIVES FOR YOUR ATARI COMPUTER

WARP SPEED SOFTWARE DISK READING AND WRITING 500% FASTER

HAPPY BACKUP — Easy to use backup of even the most heavily protected disks
HAPPY COMPACTOR — Combines 8 disks into 1 disk with a menu
WARP SPEED DOS — Improved Atari DOS 2.0S with WARP SPEED reading & writing
SECTOR COPIER — Whole disk read, write and verify in 105 seconds
1050 ENHANCEMENT — Supports single, 1050 double, and true double density
810 ENHANCEMENT — Supports single density

SPECIAL SUGGESTED RETAIL PRICE: Get the HAPPY ENHANCEMENT 810 or 1050 version with the HAPPY BACKUP PROGRAM, plus the multi drive HAPPY BACKUP PROGRAM, plus the HAPPY COMPACTOR PROGRAM, plus the HAPPY DRIVE DOS, plus the HAPPY SECTOR COPY, all with WARP DRIVE SPEED, including our diagnostic, a $350.00 value for only $249.95, for a limited time only! Price includes shipping by air mail to U.S.A. and Canada. Foreign orders add $10.00 and send an international money order payable through a U.S.A. bank. California orders add $16.25 state sales tax. Cashiers check or money order for immediate shipment from stock. Personal checks require 2-3 weeks to clear. Cash COD available by phone order and charges will be added. No credit card orders accepted. ENHANCEMENTS for other ATARI compatible drives coming soon, call for information. Specify 1050 or 810 ENHANCEMENT, all 1050s use the same ENHANCEMENT. Please specify -H model for all 810 disk drives purchased new after February 1982, call for help in 810 ENHANCEMENT model selection. Dealers now throughout the world, call for the number of the dealer closest to you.

ATARI is a registered trademark of Atari Computer Inc

HAPPY COMPUTERS, INC. • P. O. Box 1268 • Morgan Hill, California 95037 • (408) 779-3830

PRODUCT MART

DUST COVERS

For Personal Computers, Peripherals, Game Units–Protective, Long-Lasting Vinyl Resists Both Dust and Liquids.

— CHOICE OF COLORS —

Amdek	IBM PC
Apple	Mattel
Atari	Rana Systems
BMC	Sanyo
Commodore	Star Micronics
Coleco	TI 99/4
Epson	TRS 80
Franklin Ace	*PLUS OTHERS*

GROUP/VOLUME DISCOUNTS AVAILABLE

FOR **FREE BROCHURE** WRITE:

ENCHANTED FOREST
P.O. Box 5261, Newport Beach, CA 92662
(1129 W. Balboa Blvd.)

Dealer Inquiries Invited

RENT SOFTWARE

1-2-3 Lotus 99.00*
Home Accountant 15.00*
PFS: Write 28.00*
VisiCalc 50.00*

FREE CATALOG

*Ask about our membership

Games / Business / Utilities / Education

1-800-221-1031 California
1-800-221-4568 All Others
619-481-0559 San Diego

SOFTRENT™

Advertisers Index

COMPUTE!'s GAZETTE Back Issues 183
COMPUTE!'s PC & PCjr Subscription 66

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 7/31/84 CO584

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 7/31/84 CO584

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 7/31/84 CO584

COMPUTE! Reader Service
P.O. Box 11747
Philadelphia, PA 19101

Place
Postage
Here

Place
Postage
Here

COMPUTE! Reader Service
P.O. Box 11747
Philadelphia, PA 19101

COMPUTE! Reader Service
P.O. Box 11747
Philadelphia, PA 19101


"Commodore-ready",
. . . and ready for you NOW!
"Cardcorder" DC/1, Data Cassette Recorder/Player
Introducing the "CARDCORDER", Model DC/1, the Computer Cassette that is "Commodore-ready," designed for storage and retrieval of computer data efficiently, economically; with consistent performance. Yet, this fine CARDCO product is priced lower than any similar product with special quality features.
Includes standard connector which is "Commodore-ready"; LED "save" indicator light which confirms data recording on to the tape; handles up to 120 minutes (60 minutes on each side) of any standard tape including existing pre-recorded commercial as well as personal data tapes intended for use with Commodore Personal Computers; ready to go . . . just plug it in and record efficiently.
CARDCO's "CARDCORDER" COMPUTER CASSETTE is a quality data cassette recorder/player in an attractive polystyrene case, with all the standard cassette functions: record . . . play . . . rewind . . . fast forward . . . stop and eject . . . pause. A solid-state designed product of the finest components with auto-stop.
The "CARDCORDER" DC/1 carries a 90 day warranty to original owners.
All CARDCO products are available at your local dealers.
AUTO STOP
COUNTER
SAVE
RECORD
PLAY
REWIND
F FWD
STOP/EJT
PAUSE
cardco, inc.
DC/1
cardco, inc.
313 Mathewson Wichita, Kansas 67214 (316) 267-6525
"The world's largest manufacturer of Commodore accessories."
Commodore™ is a registered trademark of Commodore Business Systems, Inc.