A Beginner's Guide To Programming Languages

COMPUTE!

$3.00
June
1987
Issue 85
Vol. 9, No. 6
$4.25 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

Laser Chess™

A fascinating, futuristic two-player strategy game—our best ever. For the Commodore 64, Apple II, Amiga, and Atari XL/XE.

64 RAMdisk
Access data instantly with this "disk drive in your computer"

Full-Screen Editor For Applesoft
Easy editing with this feature-packed programmer's aide

Fast Fractal Landscapes For IBM

Atari XL/XE Super Editor
Add four powerful new editing features

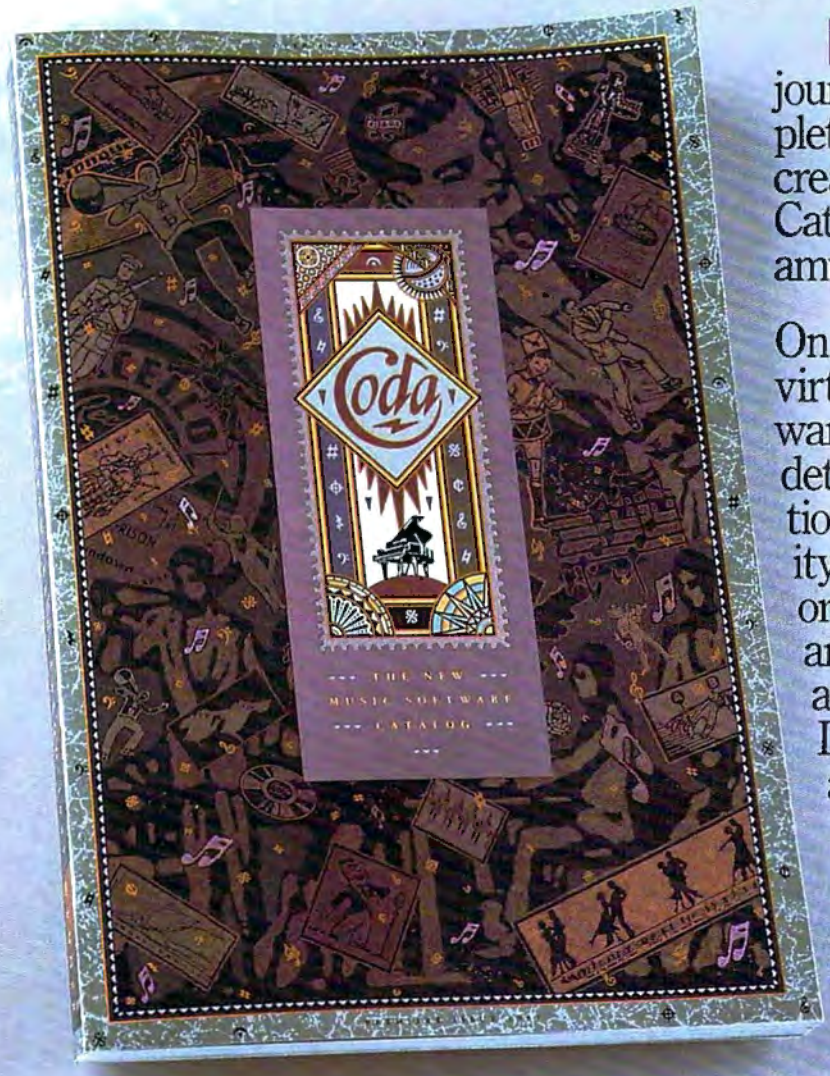
Super Hi-Res Graphics And Sound On The Apple IIGS

Amiga Disk-Based Character Sets

College Fund Planner For The IBM PC
How to calculate the funds needed for your child's education

06
71486 02193 3

How To Find Singing Frogs, Bathtub Reverb, Secrets Of Analog And Other Information Beethoven Would Have Killed For.



You're about to embark on a journey through the most complete music software catalog ever created. It's called the Coda Catalog. 160 pages of intrigue, amusement, and information.

On over 600 products, including virtually every piece of music software that exists today. Coda is detailed with whimsical illustrations. And written with a simplicity you'll appreciate. Use it to order software, books, videos, and equipment. All at the guaranteed lowest price. For Apple, IBM, Macintosh, Amiga, Atari, and Commodore computers.

Quite simply, Coda is the best source of music software in the world. Or as one critic so eloquently put it, "Beethoven would have killed for this information." Only $4. Order by calling toll free 1-800-843-1337. Or collect 612-854-9554. Oh, by the way, the singing frog is on page 114.

Coda™

Wenger Corporation, Music Learning Division, 1401 East 79th Street, Minneapolis, MN 55420-1590

We'll pay you to take the most exciting classes anywhere.

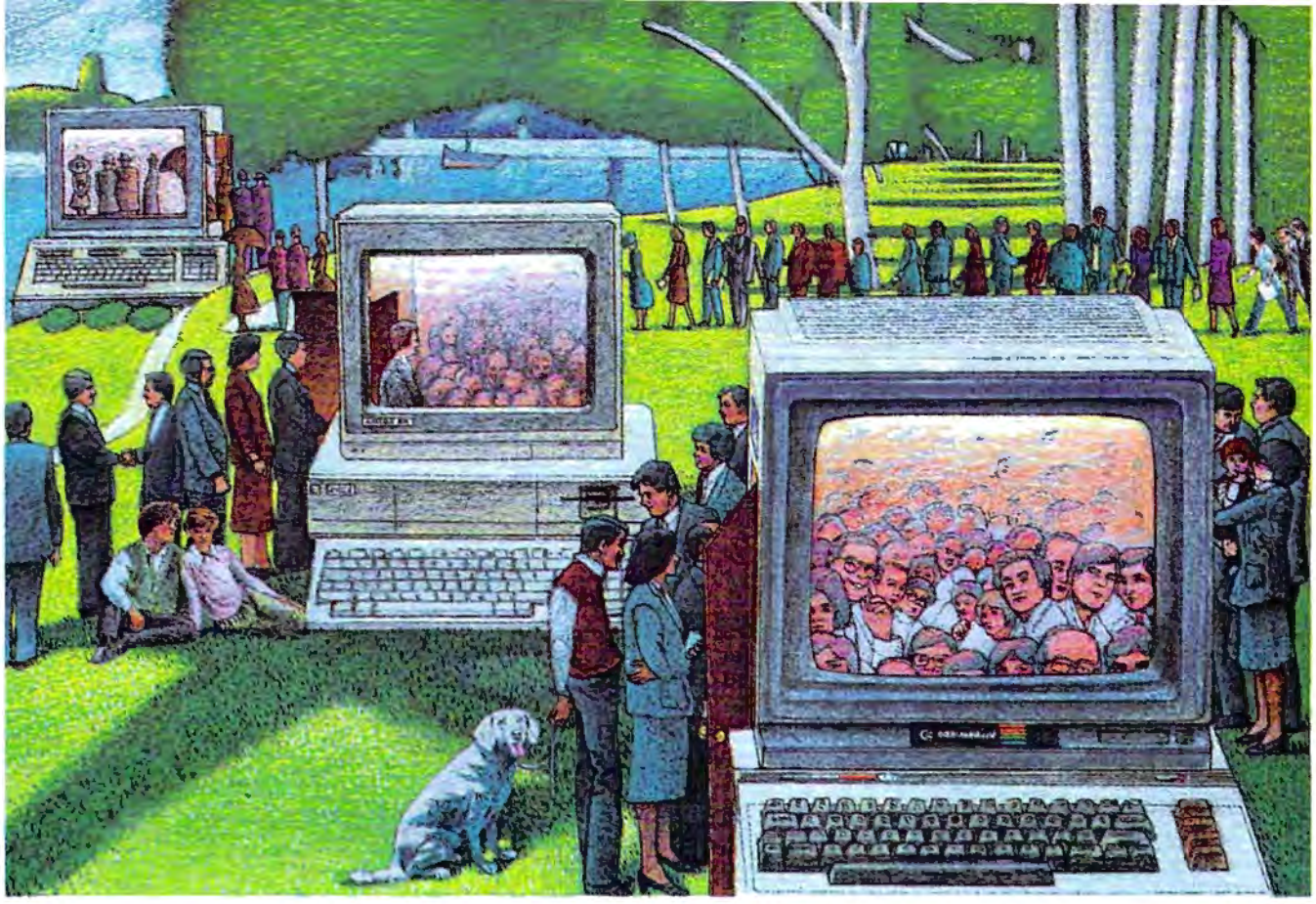
You'll learn electronics, avionics, aircraft maintenance, health care sciences, management or logistics—the Air Force will train you in one of more than 200 technical specialties America needs today.

You'll get hands-on experience with the latest equipment, and we'll pay 75% of your tuition for off-duty college courses, to get you even further.

Whatever your goals, the Air Force will equip you with the skills to get where you want to be.

If you're looking seriously into your future, Aim High to a future in the Air Force. Visit your Air Force recruiter today or call toll-free 1-800-423-USAF (in California 1-800-232-USAF).

AIR FORCE



USE THE BRAINS YOUR COMMODORE WASN'T BORN WITH.

Right at Your Fingertips in CompuServe's Commodore® Forums

Our Commodore Forums involve thousands of Commodore users worldwide. These Forums show you just how easy and fun it is to get the most from your Commodore Computer.

The Commodore **Communications Forum** provides the latest news on communications software and advice on how to effectively use your machine for online computing.

The Commodore **Programming Forum** supports programmers and developers of Commodore 8-bit computers.

The Commodore **Arts and Games Forum** is for all Commodore 8-bit computers. Compose or play music, create or retrieve colorful graphics, and download games software.

The Commodore **Amiga® Forum** is the national resource for all business and entertainment applications in the Amiga community.

Easy access to free software.

- *Download first-rate, non-commercial user-supported software and utility programs.*
- *Take advantage of CompuServe's inexpensive weeknight and weekend rates (when Forums are most active, and standard online charges are just 10¢ a minute).*
- *Go online in most major metropolitan areas with a local phone call.*
- *Receive a* ***$25.00 Introductory Usage Credit*** *when you purchase your CompuServe Subscription Kit.*

Information you simply can't find anywhere else.

Use the Forum *Message Board* to exchange mail with fellow members. Join ongoing, real-time discussions in a Forum *Conference*—with Commodore luminaries like Jim Butterfield, Jim Gracely, Bob Landwehr and Steve Punter. Scan Forum *Data Libraries* for free software, documentation and contributions from Commodore enthusiasts.

Enjoy other useful services, too. Like electronic editions of your favorite magazines, newsletters and articles, including Family Computing, OMNI Online and the Electronic Gamer.™

All you need is your Commodore computer and a modem...or almost any other personal computer.

To buy your Subscription Kit, see your nearest computer dealer. Suggested retail price is $39.95. To receive our free brochure, or to order direct, call 800-848-8199 (in Ohio, call 614-457-0802). If you're already a CompuServe subscriber, type GO CBMNET (the Commodore Users Network) at any ! prompt to see what you've been missing.

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220

800-848-8199

In Ohio, call 614-457-0802

An H&R Block Company

COMPUTE!

JUNE 1987
VOLUME 9
NUMBER 6
ISSUE 85

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 122 before typing in programs.

AP Apple, **Mac** Macintosh, **AT** Atari, **ST** Atari ST, **64** Commodore 64, **128** Commodore 128, **PC** IBM PC, **PCjr** IBM PCjr, **AM** Amiga
*General interest.

COMPUTE! Publications, Inc.
Part of ABC Consumer Magazines, Inc.
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., 825 7th Ave., New York, NY 10019 USA. Phone: (212) 265-8360. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. POSTMASTER: Send address changes to: **COMPUTE!** Magazine, P.O. Box 10955, Des Moines, IA 50950. Second class postage paid at Greensboro, NC 27403 and additional mailing offices.

ISSN 0194-357X.

Editor's Notes

In this issue we're proud to present *Laser Chess*™, a fascinating strategy game. Inspired by—but far more futuristic than—traditional chess, this captivating game can be played only on a computer. The winning entry in our $10,000 programming contest for COMPUTE!'s Atari ST Disk & Magazine, it has now been translated into Amiga, Commodore 64, Apple II, and Atari XL/XE versions.

Elsewhere you'll find a variety of useful utilities and applications. For Commodore 64 programmers, "64 RAMdisk" is highly useful in program development, and "ML Runner" turns machine language programs into files that load and run like BASIC files. "Full-Screen Editor for Applesoft" makes it far easier to program Apple BASIC. "Atari XL/XE Super Editor" significantly improves the screen editor in these machines, and "Atari NoDOS" can restore programs which seemed hopelessly lost. "Fast Fractal Landscapes" for the IBM PC and compatibles offers impressive graphics detail and speed, and "College Planner" and "Car Payments" are compact, easy-to-use programs that help with financial planning. "GraphiDemo for Amiga" is a fast and colorful program which shows off the tremendous processing power and graphics of the Amiga. And "Amiga Disk-Based Fonts" is a helpful utility with a tutorial on how to load custom text fonts from disk and install them from BASIC.

Readers who have been with us for some time are likely to agree that the quality of the articles and programs in this issue does not come as a surprise; rather it's part of a tradition established by Robert Lock, the founder of COMPUTE! Publications.

This past December Robert withdrew from the daily management of the company and accepted the position of Editorial Consultant. He will continue to provide the company with guidance and will assist, in particular, in the development of new products and services.

This change in editorial management gives us our first opportunity to publicly acknowledge his contributions to the growth and success of the company he founded and to the personal computing industry as a whole.

In all these years he has never given an interview or in any fashion moved his personality into public view. While some other industry leaders appeared more concerned at times with their personal image than with the health of their companies, Robert has always worked quietly behind the scenes. We can, however, with this editorial, recognize his contribution and thank him in a small way for his efforts.

He was in his early thirties when he had the idea of starting a magazine devoted to consumer computing. And in the past seven years, COMPUTE!—under Robert's guidance—has grown into a highly successful publishing group: four magazines, over 150 books in print, and over 1,000,000 readers a month. COMPUTE! Publications has become a major contributor to the ongoing introduction of computing into the homes, schools, and businesses of America and elsewhere in the world.

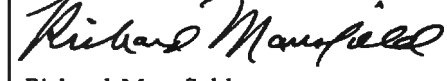
He saw early on that computers would have an immense impact. Starting in a storefront in Greensboro, with a handful of employees, he began working on the early issues of COMPUTE!. At that point, he was personally involved in every aspect of his young magazine: He pasted up galleys, called advertisers, contacted authors, and edited copy.

As the years went by, he continued to directly supervise the growing company in both its editorial and business activities. He has an intuitive grasp of business and finance combined with strong editing and writing skills. This combination of talents is as rare as it is powerful. And perhaps even more rare: He is an extremely clear thinker. Those of us who work closely with Robert have always been aware that he is remarkably accurate in his assessments on a wide range of topics.

We at COMPUTE! were not concerned during the notorious industry shakeout of 1984–85. We knew we would be among the survivors. In fact, COMPUTE!'s Gazette was introduced in the face of the shakeout, and became one of the strongest startups in magazine publishing history. We have gone on to publish a number of bestselling books and continue to feature some of the best programs, columns, and articles available on the subject of home and recreational computing.

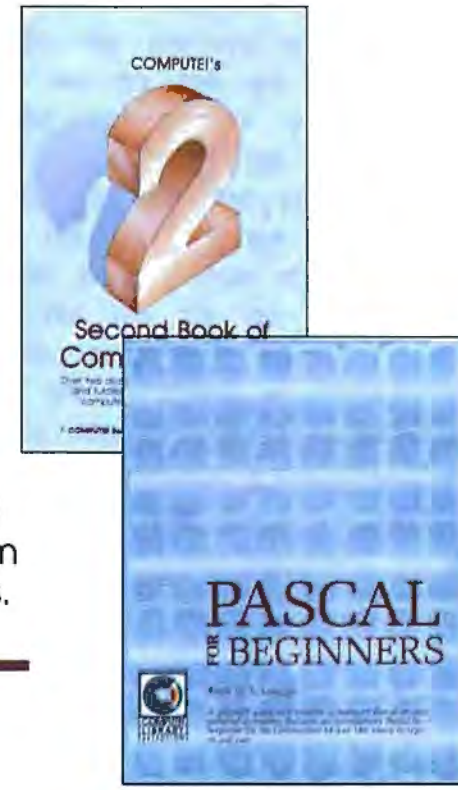
It all began with a single idea, Robert's vision of how best to engage, entertain, and inform the reader about a powerful emerging technology that he saw would eventually affect every aspect of our lives. But a single idea, however accurate, rarely leads to the creation of a major publishing house. Thousands of others also began to see the importance of consumer computing by the end of the 1970s, and hundreds of publications were introduced. By 1983 there were 150 computer magazines competing for the attention of the reading public. Today there are only a few. That COMPUTE! Publications survived and flourished was largely due to Robert's strong leadership. The staff at COMPUTE!, and the readers who enjoy our efforts, are fortunate that he will continue to play a vital role, contributing to the direction of our publications as we grow.

In the coming years we will, with his help, preserve the traditions and values Robert established here. And we will expand, offering more comprehensive coverage as well as maintaining the quality of programs, tutorials, and features you've come to expect from COMPUTE! Publications.



Richard Mansfield
Editorial Director

New books from COMPUTE!



COMPUTE! Books is bringing you a brand new line up of books for your Commodore 64 and 128. These recent releases offer you everything from programming hints to exciting games, from educational to home and business applications.

Pascal for Beginners

$14.95 0-87455-068-8
Book/disk combination for the Commodore 64
$29.95 ISBN 0-87455-069-6

This introductory text to standard Pascal on any computer is an ideal tutorial for anyone who wants to learn this powerful computer language. It includes everything you need, including an introductory Pascal interpreter* for the Commodore 64 and 128 in 64 mode, ready to type in and use. Written in plain English and offering numerous program examples, it gently and clearly explains standard Pascal and structured programming. Latter sections include discussions of advanced topics such as files and dynamic data storage. *There is also an optional disk available for $12.95 for the Commodore 64 which includes most of the programs in the book. 688BDSK.*

*The Commodore 64 Pascal interpreter is not full-featured, but still a powerful implementation of Pascal which suits the needs of most beginners.

COMPUTE!'s Music System for the Commodore 64 and 128

Book/disk combination only
$24.95 ISBN 0-87455-074-2

Sidplayer, the feature-packed, popular music player and editor program, is now more versatile and more impressive than before. *Enhanced Sidplayer* for the Commodore 128 and 64 includes two new versions—one for the Commodore 128 running in 128 mode and another for the Commodore 64. Take advantage of every feature the SID chip (the sound chip in the 128 and 64) has to offer. Just like the original, *Enhanced Sidplayer* is easy to learn and use, with many powerful new features. The accompanying disk contains the editor, player programs (including a Singalong program), utilities, and sample music that you can enjoy immediately or change. The new *Sidplayer* plays any songs created by the original *Sidplayer* for the Commodore 64.

User's Guide to *GEOS*: geoPaint and geoWrite

$18.95 ISBN 0-87455-080-7

Learn the ins and outs of *GEOS*, the new icon-based operating system for the new Commodore 64C and the 64, with this step-by-step guide. Everything from creating simple letters with *geoWrite* and pictures with *geoPaint* to merging text and graphics and using desk accessories is clearly and concisely explained.

COMPUTE!'s Second Book of the Commodore 128

$16.95 ISBN 0-87455-077-7

The editors at COMPUTE! Publications have collected some of the best games, programs, and tutorials for the Commodore 128 (in 128 mode) from *COMPUTE!* magazine and *COMPUTE!'s Gazette*. Like *COMPUTE!'s First Book of the Commodore 128*, this book offers a variety of programs and articles for every 128 user. Each program has been fully tested and is ready to type in and use on the Commodore 128 running in 128 mode. *There is also a disk available for $12.95 which includes the programs in the book. 777BDSK.*

Mapping the Commodore 64, Revised

$16.95 ISBN 0-87455-082-3

An update of the bestselling memory map and programming guide. It's a necessity for intermediate and advanced programmers. This definitive sourcebook has been expanded and now covers the new icon-based *GEOS* (Graphics Environment Operating System) with clear descriptions of how to make it work for you. For BASIC and machine language programmers of both the Commodore 64 and 64C.

Look for COMPUTE! Books at your local computer or book store.
Or, to order directly from COMPUTE!, call toll free 1-800-346-6767 (in NY 212-887-8525) or write COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please include shipping and handling: $2.00 per book in U.S. and surface mail; $5.00 airmail.
NC residents add 5 percent sales tax and NY residents add 8.25 percent sales tax.
Please allow 4–6 weeks for delivery.

COMPUTE! Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies

COMPUTE! Books are available outside the United States from subsidiaries of McGraw-Hill International Book Company.

Publisher	James A. Casella
Editorial Director	Richard Mansfield
Managing Editor	Kathleen Martinek
Associate Publisher	Selby Bateman
Editor, COMPUTE! & COMPUTE!'s GAZETTE	Lance Elko
Assistant Editor, COMPUTE!	Philip I. Nelson
Production Director	Tony Roberts
Editor, COMPUTE!'s Atari ST Disk & Magazine	Tom R. Halfhill
Technical Editor	Ottis R. Cowper
Assistant Technical Editors	George Miller, Dale McBane
Assistant Editor, COMPUTE!'s Atari ST Disk & Magazine	Todd Heimarck
Assistant Editor, COMPUTE!'s GAZETTE	Rhett Anderson
Assistant Editor	John Shadle
Assistant Features Editor	Kathy Yakal
Programming Supervisor	Patrick Parrish
Editorial Programmers	Tim Victor, Tim Midkiff, William Chin
Copy Editors	Karen Uhlendorf, Karen Siepak
Editorial Assistant	Caroline Hanlon
Submissions Reviewer	David Hensley
Programming Assistants	David Florance, Troy Tucker
Executive Assistant	Debi Nash
Administrative Assistants	Julia Fleming, Iris Brooks, Mary Hunt, Sybil Agee
Receptionist	Anita Armfield
Associate Editors	Jim Butterfield Toronto, Canada Fred D'Ignazio Birmingham, AL David Thornburg Los Altos, CA
Contributing Editor	Bill Wilkinson
COMPUTE!'s Book Division Editor	Stephen Levy
Assistant Editors	Gregg Keizer, Tammie Taylor
Director of National Sales	Joseph W. Hatcher
Production Manager	Irma Swain
Art & Design Director	Janice R. Fary
Assistant Editor, Art & Design	Lee Noel
Mechanical Art Supervisor	De Potter
Artists	Robin Case, Kim Potts
Typesetting	Terry Cash, Carole Dunton
Illustrator	Harry Blair
Director of Advertising Sales	Peter Johnsmeyer
Associate Advertising Director	Bernard J. Theobald, Jr.
Production Coordinator	Kathleen Hanlon
Customer Service Manager	Diane Longo
Dealer Sales Supervisor	Jose Cruz
Individual Order Supervisor	Cassandra Green

James A. Casella, President
Richard Mansfield, Vice President, Editorial Director
Richard J. Marino, Vice President, Advertising Sales
Christopher M. Savine, Vice President, Finance & Planning

Editorial Board
Richard Mansfield
Kathleen Martinek
Selby Bateman
Lance Elko
Tom R. Halfhill
Stephen Levy
Robert Lock, Founder and Editorial Consultant

COMPUTE! Publications, Inc. publishes:

COMPUTE!
COMPUTE!'s Gazette
COMPUTE!'s Gazette Disk
COMPUTE!'s Apple Applications Special
COMPUTE!'s Atari ST Disk & Magazine

Editorial offices:	324 West Wendover Avenue Suite 200 Greensboro, NC 27408 USA
Corporate offices:	825 7th Avenue New York, NY 10019 212-265-8360
Customer Service:	800-346-6767 (In NY 212-887-8525)
Hours:	9:30 A.M.–4:30 P.M. Monday–Friday

Coming In Future Issues

Weather Wizard For The IBM PC, Commodore 64, Atari, Apple II, Amiga, and Atari ST

Vanishing Directory For The IBM PC

Sprite Designer For The Commodore 64

Medium-Resolution Autorun For The Atari ST

P/M Magic For The Atari

1571 Utility Package For The Commodore 128

IBM Printer Control

Zookeeper For The Amiga

Subscription Orders & Inquiries

COMPUTE!
P.O. Box 10954
Des Moines, IA 50340

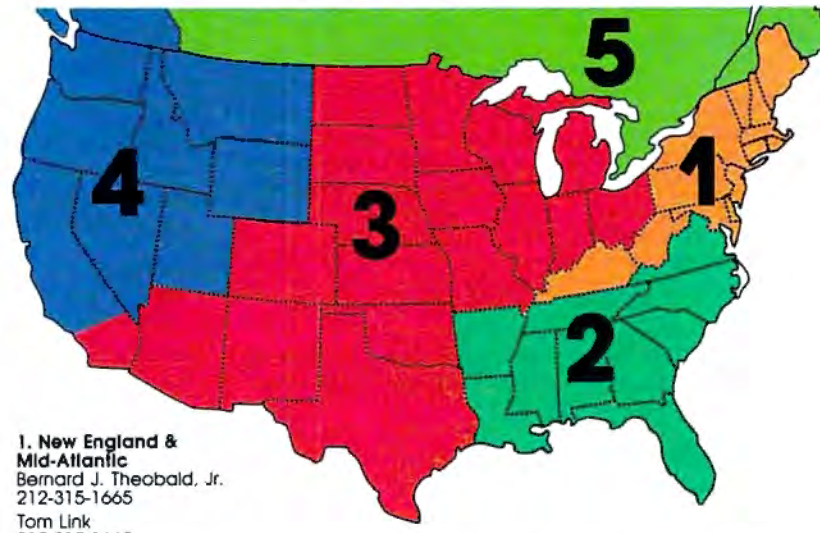
TOLL FREE Subscription Order Line
800-247-5470
In IA 800-532-1272

COMPUTE! Subscription Rates (12 Issue Year):

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

Member ABC — Audit Bureau of Circulations
MPA — Magazine Publishers Association

Advertising Sales



1. New England & Mid-Atlantic
Bernard J. Theobald, Jr.
212-315-1665
Tom Link
212-315-1665

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

3. Midwest & Southwest
Jerry Thompson
312-726-6047 (Chicago)
713-731-2605 (Texas)
303-595-9299 (Colorado)
415-348-8222 (California)
Lucille Dennis
415-348-8222

4. West, Northwest & British Columbia
Jerry Thompson
415-348-8222
Lucille Dennis
415-348-8222

5. Canada
Harry Blair
919-275-9809

Director of Advertising Sales:
Peter Johnsmeyer

Associate Advertising Director:
Bernard J. Theobald, Jr.

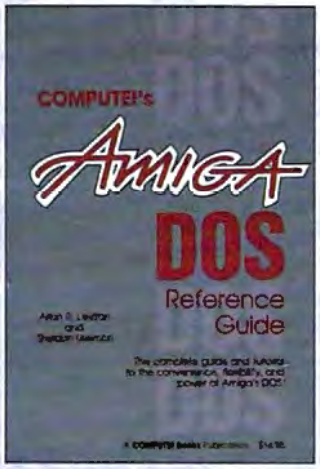
COMPUTE! Sales Office 212-315-1665

Address all advertising materials to:
Kathleen Hanlon
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408

AMIGA

SUPPORT FROM COMPUTE! BOOKS

Everything for the Amiga. From BASIC beginner's guides to advanced programming handbooks, COMPUTE! offers you information-packed tutorials, reference guides, programming examples, ready-to-enter applications, and games to help you develop your computing skills on Commodore's Amiga.

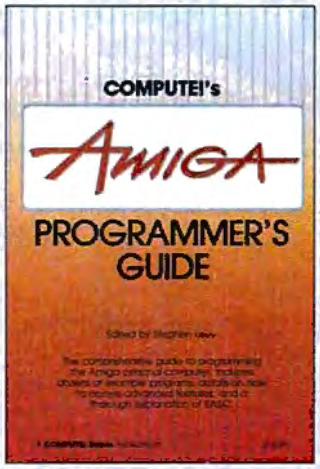


COMPUTE!'s AmigaDOS Reference Guide

Arlan R. Levitan and Sheldon Leemon
A comprehensive tutorial and reference guide to the powerful AmigaDOS—the operating system underlying the Workbench and Intuition—this book offers information useful to every Amiga owner. It defines and illustrates all DOS commands, and shows you how to create file directories, access peripherals, run batch file programs, and avoid "disk shuffle." The screen- and line-oriented text editors are explained in detail. Numerous examples and techniques explain how to use AmigaDOS to make operating your Amiga both convenient and efficient.
$16.95 ISBN 0-87455-047-5

Elementary Amiga BASIC

C. Regena
Here's your introduction to the new and powerful BASIC on the Amiga personal computer. The Amiga's impressive graphics, animation, and sound can be unlocked with the right commands, and BASIC is the place to start. Complete descriptions of Amiga BASIC's commands, syntax, and organization take you from the beginner level to a full-fledged programmer. Plus, the book offers you ready-to-type-in programs and subroutines while showing you how to write your own programs. *There is a disk available which includes the programs in the book,* $12.95. *This title is also available as a book/disk combination for $29.95* (057-2).
$14.95 ISBN 0-87455-041-6

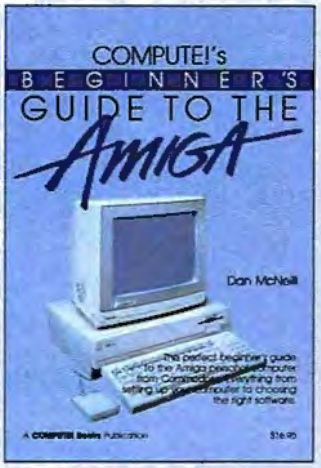


COMPUTE!'s Amiga Programmer's Guide

Edited
Your tutorial and reference manual to AmigaDOS, BASIC, Intuition, and other important software tools which accompany the new Amiga, *COMPUTE!'s Amiga Programmer's Guide* is a clear and thorough guide to the inner workings of this fascinating new-generation computer. The great speed of its 68000 microprocessor, coupled with the versatility of the Amiga-specific graphics and sound, makes the Amiga one of the most powerful computers available today.
This book is the key to accessing the Amiga's speed and power.
$17.95 ISBN 0-87455-028-9

Advanced Amiga BASIC

Tom R. Halfhill and Charles Brannon
This guide to applications programming on Commodore's new Amiga contains everything an intermediate programmer requires to begin creating sophisticated software on this powerful machine, including several ready-to-type-in programs. Clear, yet comprehensive documentation and examples cover advanced BASIC commands, designing graphic applications, generating sound and music, using the Amiga's built-in speech synthesizer, creating a user interface, and programming the computer's peripherals. *There is a disk available which includes the programs in the book,* $15.95. (June release)
$17.95 ISBN 0-87455-045-9



COMPUTE!'s Beginners Guide to the Amiga

Dan McNeill
Written in a lively and entertaining style, this book teaches you everything a beginner needs to know to get started quickly with the Amiga from Commodore. You will learn about setting up the system, all the most popular types of software, and details about the hardware.
$16.95 ISBN 0-87455-025-4

Inside Amiga Graphics

Sheldon Leemon
The Amiga, Commodore's powerful new computer, is an extraordinarily impressive graphics machine. Easy to use, the Amiga can produce color graphics and excellent animation. You'll find thorough descriptions of the computer's abilities and the hardware required to create a complete graphics system. Software, too, is central to the Amiga's power, and complete tutorials show you how to get the most from the machine. (June release)
$17.95 ISBN 0-87455-040-8

COMPUTE!'s Kids and the Amiga

Edward H. Carlson
The latest in this bestselling series written by Edward Carlson, *COMPUTE!'s Kids and the Amiga,* will acquaint you with BASIC. Over 30 sections—all with instructor notes, lessons, assignments, and lively illustrations—entertain and amuse you as you learn to program your new computer. Clear writing and concise examples make it easy for anyone—children and adults alike—to painlessly learn BASIC. (May release)
$14.95 ISBN 0-87455-048-3

Look for these books at your local book or computer store. Or order directly from COMPUTE!. Call toll-free 1-800-346-6767 (in NY 212-887-8525).

Please allow 4-6 weeks for delivery after your order is received.

COMPUTE! Publications, Inc.
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE! COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications

COMPUTE! books are available outside the United States from subsidiaries of McGraw-Hill International Book Company.

A Beginner's Guide To Programming Languages

Tom R. Halfhill, Staff Editor

How hard is programming—really? And which language is best? Here are some straight answers to some of the most commonly asked questions about programming on personal computers.

Recently a teacher whose school was being equipped with personal computers for the first time related how the faculty was introduced to the idea:

"The computer expert for the school district came down and grouped us all together in a room. None of the teachers but me had ever worked with a computer before. The expert proceeded to give us a crash course in computers—but started off by trying to teach us BASIC programming. Pretty soon he was talking about numeric variables, string arrays, and subroutines, and got everybody lost. When the session ended and we left the room, I heard one teacher mutter, 'That's the last time I'm ever going *near* one of those things!'"

These teachers weren't the first people to be discouraged by the complexities of programming. Over the past several years, thousands of students, too, have been exposed to the same approach to computer literacy. Thankfully, most school systems have now recognized the difference between computer *users* and computer *programmers* and have restructured their computer-literacy courses accordingly.

There was a time when a person had to become a mechanic to drive an automobile, become a pilot to fly, and set up a darkroom to take pictures. But as cars, airplanes, and cameras advanced beyond the primitive stage to become mature machines, it became less necessary to wrestle with their inner workings. Likewise, you don't have to be a programmer to operate a computer intelligently and usefully.

Ten years ago, programming was almost unavoidable if you wanted to use a microcomputer. Commercial software was practically nonexistent, and the handful of public domain programs available were usually considered starting points for your own modifications. But now the personal computer is a mature enough machine so that the average person shouldn't have to struggle with string arrays, subroutines, and recursive procedures. You can probably buy the software you need right off the shelf, type it in from a magazine or book, download it from a bulletin board system or information service, or obtain it from a user group library.

Still, there comes a time when nearly every computer user considers the question: Should I learn to program? Maybe now, as a knowledgeable user, you've realized that computers really aren't as mysterious as they first seemed, so you're wondering if programming might be within your grasp, too. Or maybe you're the type who *likes* to tinker with inner workings—you won't be satisfied until you have total control over the machine. Or maybe you have an unusual application in mind and just can't find the right software to do the job.

If you fit any of these descriptions—or if you're merely curious—you may want to take the plunge into programming.

Primitive Languages

First, a definition: *A program is simply a list of instructions for the computer to carry out.* Some instructions may tell the computer to load a file from disk; others may tell the computer to display something on the screen. It's your job, as a programmer, to write these instructions in the proper sequence. The "proper sequence," of course, depends on what you want the program to do.

So far so good. Now all you have to do is learn what the instructions are and how to fit them together. And this is where things start to get sticky.

Beginning programmers are sometimes dismayed when they discover that the instructions recognized by computers are much more

We've got you covered

on land

C= ● 人 $30.00

on sea

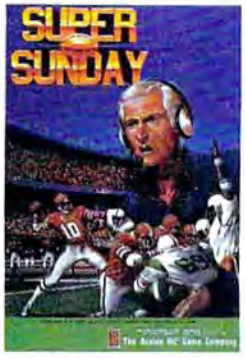
C= ● $30.00

in the air

C= $35.00

and on the court

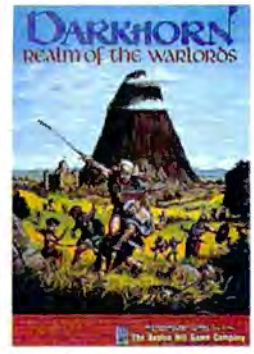
Coming Soon:
C= ● IBM



on the field

C= ● IBM $35.00

and in your mind!



C= ● $30.00

Available at leading game and computer stores
or call TOLL FREE **1-800-638-9292** for ordering information.

microcomputer games DIVISION
The Avalon Hill Game Company
A MONARCH AVALON, INC COMPANY
4517 Harford Road, Baltimore, Maryland 21214 • 301 254-9200

primitive than they anticipated. COMPUTE! has received letters that read something like this: "Can you tell me which command lets you move a sprite around the screen with a joystick and lets it fire shots?" Or, "Which command sorts a list of names in alphabetic order?"

There are answers to these questions, but they fill enough books to support a large part of the computer publishing industry.

The problem is that individual programming instructions do relatively little. It takes a fair number of them strung together to make the computer do something minimally useful or even recognizable. Imagine, for instance, sending someone to the store for a loaf of bread. Instead of simply saying, "Please go to the store and get a loaf of bread," you have to tell the person, "Walk one block to the corner and turn left. Then walk one more block and enter Joe's Market on the right. Pick up a loaf of bread and pay for it at the counter. Then walk one block back to the corner, turn right, and walk another block home again."

This kind of detail is necessary even when programming in a so-called *high-level language*, like BASIC, Pascal, Modula-2, or Logo. Programming in a lower-level language like C, Forth, or, ultimately, machine language, would be like telling the person how to walk by putting one foot in front of the other, how to open and close a door, how to turn, how to recognize a loaf of bread and pick it up, and how to pay for it.

Computer languages are a little more advanced than the grunts of primitive cavemen—but not much.

Don't let this discourage you, though. The relative simplicity of a computer language makes it much easier to learn than a modern human language. English, French, and German consist of many thousands of words and many dozens of grammatical rules. BASIC, Pascal, and Forth—even the most powerful versions—rarely have more than 100–200 words and a few dozen rules. And, just as your working vocabulary in English is only a fraction of the total language, your working vocabulary in BASIC or Pascal may amount to fewer than 30 commands about 90 percent of the time.

Someday it may be possible to program a computer in English, French, or German. Reseachers at this very moment are working on a *natural language interface*. But until computers get a lot smarter, it's going to be easier for us to learn their languages than for computers to learn ours.

Sorry, No Esperanto

Your first step, then, is to decide which computer language you want to learn. You might be surprised to discover that there are about as many computer languages as there are human languages. If you were born and raised in an English-speaking country, you probably grew up speaking English. However, that same kind of choice isn't made for you when you buy a computer, even if BASIC is built in. Your computer can run several different languages.

Why are there so many computer languages to choose from? After all, it's perfectly logical that hundreds of different human languages developed. There was no communication between different peoples for centuries, and by the time someone came up with the idea of inventing a standard language like Esperanto, it was far too late to get everyone to agree. But computers are a recent invention, so it seems like somebody could have set the standard at the beginning and saved us a lot of trouble.

It's not that simple, though. The reason why there are scores of computer languages is that each one was conceived to solve certain kinds of problems, to run on certain kinds of computers, or to reflect certain programming philosophies. For instance, BASIC—which stands for Beginner's All-purpose Symbolic Instruction Code—was invented by a pair of university professors in the mid-1960s who wanted a language easy enough for neophytes to learn in a few hours. Likewise, Pascal was invented by a Swiss professor who sought a language that was ideal for students. FORTRAN (FORmula TRANslator), one of the oldest languages, was invented for scientists and engineers. COBOL (COmmon Business-Oriented Language), another ancient computer tongue, was invented for writing business programs. The U.S. Department of Defense even has its own computer language, Ada.

Most languages have also sprouted several dialects. There are dozens of versions of BASIC, for example. Although these dialects are largely similar, there are still too many dissimilarities to allow all but the simplest programs written in one dialect to run under another. Sometimes an extensive rewrite is necessary.

Like choosing a computer, choosing a language is an emotional issue for some people. If you start asking around for advice, expect to hear conflicting arguments about which language is best, and why.

Back To BASICs

All in all, a good starting point for the beginning programmer is BASIC—if only because it's the language you probably already own. Most personal computers come with BASIC on a system disk or built into read only memory (ROM). A BASIC programming manual was probably included in the package, too. If not, a trip to a bookstore or library will turn up dozens of volumes of BASIC manuals.

Besides the fact that it's free, BASIC is a very flexible language that's ideal for experimenting and writing short programs. If you play around with BASIC for a few weeks and decide that programming isn't much fun, at least you haven't spent the $50 to $500 that another language may cost.

As an alternative, some languages are available in public domain versions for the nominal cost of a user group library disk or download charges from a bulletin board system or information service. Forth and COMAL are prime examples.

To help you get started, the following pages cover the features of some of the most popular languages available on today's personal computers. This isn't intended to be a complete list, of course, but it will introduce you to some of the pros and cons of several of the most widely-used languages. If you consider the options carefully before making your choice, odds are you won't be disappointed.

BASIC

Todd Heimarck
Assistant Editor

No matter which computer you buy, you'll almost certainly be given a version of the BASIC language with your system. It might be in Read Only Memory (ROM), on a plug-in cartridge, or in a program on the system disk, but whatever form it takes, it's there. BASIC is the *lingua franca*, the universal language, for microcomputers.

Everybody has BASIC. You can write a program, save it to disk, and give a copy to a friend or put it up on a local bulletin board system. Magazines and books publish BASIC program listings which you can study to learn new techniques. Many user groups, schools, and other organizations offer classes in BASIC programming.

If you decide at some point in the future to buy a new computer with more memory or a faster processor, you'll have a head start on programming it if you know BASIC. There are some differences between the manifold dialects of BASIC, but there's a core of commands that are the same on all computers. PRINT, FOR-NEXT, IF-THEN, and many other keywords function virtually identically in all versions of BASIC.

There are other languages available, and they have their good points. BASIC isn't perfect and you'll hear proponents of other languages making comparisons between their favorite language and BASIC. You *don't* often hear someone say, "C is a good language because it has many features that Forth lacks." That's because BASIC is the standard against which others are judged. The most popular language, it has survived for 25 years and probably will still be in use 25 years from now.

Interactive And Flexible

One of BASIC's most attractive features is that it's interactive. You can sit down at a keyboard, type in a few lines, and then type RUN. There's no need to wait while you save the source code, compile and link it, and hope that it works. You just type RUN. The program will either run or it will stop with an error message. BASIC is generally good about pointing out mistakes and telling you which line isn't working. If there's an error, you can list the line and figure out what's wrong.

BASIC encourages experiments. You can stop the program, change part of it, and see the results immediately. A weekend programmer can be creative—adding a line here, changing a variable there—much like a sculptor molding a lump of clay. When you're creating a graphics screen or playing with sound effects, BASIC's flexibility makes it easy for you to test new ideas.

Since BASIC is a high-level language with English keywords, program listings are understandable. You can usually read through a subroutine and follow its logic.

Spaghetti And Celerity

People who dislike BASIC generally offer one of two arguments against it. First, BASIC is an *unstructured* language (imposes few rules on the programmer), and programs often have so many GOTOs that reading the listing is like untangling spaghetti. Second, compared to those written in other languages, BASIC programs can run slowly.

Advocates of structured programming suggest that programs should be broken into modules of less than one page each. Variables should be declared at the beginning of each module. Loops should be indented. You should include many comments that explain what's going on. And you should never use an unconditional GOTO.

If you agree that there are advantages to structured programming, you can easily follow all of those rules in BASIC. The modules are subroutines. Variables can be defined at the start of a program. FOR-NEXT loops can be indented. You can include REMs and avoid the GOTO command. You can write highly structured programs, if that's what you want.

In other words, BASIC doesn't force you to write spaghetti code. It's possible to create a program that looks nice when it's listed. If you see a program that looks messy, blame the author—not the language.

On the other hand, you could ignore the whole idea of structure. If a program works, it works. Some very good programs don't look clean and structured on the inside, but they get the job done.

What about the speed of BASIC? If flat-out speed is what you want, then machine language (ML) is the only choice. All languages are slower than ML.

It's true that some languages are faster than BASIC. But that doesn't matter in some situations. If you're calculating a mortgage payment, for example, BASIC might take three seconds, versus one-third second in a faster language. If you had those extra seconds, what would you do with them?

In other cases, the problem isn't BASIC, but the algorithm used. For example, a bubble sort that alphabetizes a list of 500 names might run very slowly. You could rewrite the program in another language and it would run the bubble sort faster. But if you changed to a Shell sort or a quick sort, you'd notice a definite improvement in speed, even in BASIC. There are many techniques for speeding up BASIC programs.

To make comparisons fair, BASIC programs should be *compiled* (run through a special process which translates them into something resembling machine language). A compiled BASIC program often runs at the same speed as a similar program in another language. With a compiler, you can have the best of both worlds.

Machine Language

Richard Mansfield
Editorial Director

En oino aletheia, if you don't know Greek, is, as they say, Greek to you. Likewise, CMP #$FD:RTS looks pretty strange if you don't know machine language. But there's nothing inherently complicated about either language—once you know the vocabulary and the rules.

Too many BASIC programmers stay away from machine language (ML—sometimes also called *assembly language*) because they take one look at it and it appears impossibly alien. In fact, it is simply another language with about 50 words to learn and some new techniques. The fundamentals of programming in any language remain the same: data lists, loops, branching, subroutines. So, once you've learned to communicate with your computer in one language, you've got a considerable head start when learning a second language.

Machine language *does* require somewhat more exacting programming (in the sense that cooking dinner from scratch is more exacting than opening cans). But there are many tools now available to the ML programmer: single steppers, error-detecting assemblers, and a wealth of books which help the novice move quickly through a bafflement phase toward eventual mastery.

Talking Directly To The Machine

But why, if you already know another language, would you want to learn ML? One clue is that most commercial or professional programming (especially on computers such as the Commodore 64) is ML. This is because ML is the *machine's language*. Any other language requires translation before the computer can understand what to do. For example, when you use the BASIC command PRINT, the computer looks up the print subroutine. (This subroutine is, like the subroutines in virtually any computer language, written in ML.) The lookup is one level of indirection, but there's also the problem that the PRINT statement can have so many different forms (are we printing to screen? is there a TAB? and so on). All this translating takes time and, with interpreted languages like BASIC, this time is spent *while the program is running.*

The primary benefit of ML is that there is no lag time between the computer's reading the instruction and its ability to act upon the instruction. ML, the computer's own language, speaks directly to the microprocessor chip. So-called *higher-level languages* all require, to one degree or another, compromises and indirect, time-consuming, superfluous tasks. The reason for this is obvious: The computer can speak *only* one language—ML. Anything else you offer it will have to be reduced to ML before the computer can respond.

Two higher-level languages renowned for speed and power—C and Forth—illustrate this point. Both languages are interesting, enjoyable to program, and a significant improvement in efficiency over BASIC. However, after programs are written in C or Forth, they are often further modified by replacing key sections with pure ML. There are even utilities which watch an executing C program and then report where the computer is spending most of its time—where in the program are the most delays. If the report reveals that the computer is spending 40 percent of its time in lines 256–288, you rewrite that segment, replacing C with ML.

A Subtle Intimacy

The most important issue, then, is speed, and nothing can work faster than ML. For years BASIC programmers, too, have included ML sections within the BASIC program to make game animation smoother, spreadsheets calculate faster, or word processors able to keep up with quick typists. We can see this endless quest for additional speed in hardware as well: clock speeds have gone from 1 MHz to 16 MHz. In the world of computers, there's no such thing as having too much memory or being too quick.

There's an additional advantage to ML which is significant, but somewhat subtle: intimacy. With ML you are working with a *low-level language* (the only low-level language), and that means you're right down in the engine room and you've got all the tools you need to do anything you want. In other words, you're down on the computer's level, and you can see and access all its features. For example, it's often not possible from higher-level languages to test whether the BREAK or CONTROL keys are being pressed on the keyboard; to design disk directory lists which suit your needs; to redefine keys; to speed up input; or to quickly switch entire screens of information. Most of these tasks are easy to accomplish via ML; all are possible.

A machine language programmer works with instructions which, by themselves, accomplish less than many of the instructions of higher-level languages (PRINT is made up of hundreds of ML instructions). This has two primary consequences: You use more instructions per ML program, and, therefore, have more bugs to fix. It takes somewhat more time to program ML. (Your reward is that your finished program will run perhaps 100 times faster than its high-level-language equivalent.) A second consequence is that ML is freer than other languages, more uniquely the work of one individual. For this reason, higher-level languages are often chosen for team programming—the separately programmed parts are more likely to combine harmoniously into a finished product when a more rule-oriented, less individualistic language is used.

But, if you work alone, enjoy a challenge, and want to write the very best programs your computer can run, you might want to give ML a try. It's not really more difficult than other languages—just different.

C

Philip I. Nelson
Assistant Editor

You might call C a "middle-level" language, since it is easier to write and understand than machine language, but it lacks some convenience features found in high-level languages. C originated in the early 1970s at Bell Labs, where it was first used to transport the UNIX operating system from a DEC minicomputer to other computers.

Today, C is used to write everything from arcade games to operating systems for new computers. C is a favorite of commercial software houses for two main reasons: C programs compare favorably with machine language programs in performance, and C is not tied to any particular operating system, making it suitable for programs that must be translated for several different computers. And as reasonably priced C compilers have made their way into the home computer market, C has become popular with hobbyist programmers, too.

Keyword Economy

C has fewer than 30 keywords in all, and about half of those are used to declare variables and other data objects. The most important remaining keywords are

if, else, case, switch, default, for, do, while, break, continue, return, goto

If you're used to BASIC—with versions having anywhere from 50 to 200 keywords—you might wonder how any useful language can get by with so few words. Part of the answer lies in the fact that C never uses two words where one will do. For example, where BASIC uses FOR and NEXT to create a FOR loop, or WHILE and WEND to create a WHILE loop, C dispenses with NEXT in the first case and WEND in the second. The occurrence of *for*, *while*, or *do* is sufficient to identify a loop in C, and this economy of expression typifies the language in general.

Another reason why C requires few keywords is that it relies heavily on context and uses many words and operators for more than one purpose. For instance, the type identifier *int* can have quite different meanings depending on where it occurs in a program. In a variable declaration, int means *I plan to use an integer variable with the following name.* When it occurs before a function, int means *The following function returns an integer value.* And when it appears in a cast, int means *Convert the following noninteger value into an integer.*

C has a luxurious supply of operators. In a simple example, BASIC uses the statement `a = a + 1` to add 1 to the value of the variable *a*. C permits the same syntax, but it can do the job more economically, in two different ways:

```
a++;
a += 1;
```

Several variable types are available in C. You must declare every C variable—state its name and its type—before using it; this requirement, among other things, makes C programs easy to read, since variable declarations can occur in only a few places. The preferred style is to use *external* variables—those known to every function in the program—only where necessary, thus minimizing the risk that a secondary function will inadvertently garble a variable used elsewhere in the program. Most C variables are *local* to one degree or another, meaning that they are known only to a single function or related functions.

Another powerful data object is the *pointer*—a variable that points to some other object. A common use of pointers is to access individual elements of an array. However, pointers have many other uses, including direct manipulation of the computer's memory.

An array in C is a collection of data objects of the same type under one name. For example, a string is an array of objects of the type *char* (character) which ends with a zero. More complex data objects include the *structure*, which is a collection of objects of dissimilar types under one name. Each *member* of a structure can be accessed by a unique name, although the entire structure is treated as a logical unit. Not only can a structure accommodate both arrays and simple variables within the same skin, but it also can hold variables of different types—characters, integers, floating point numbers, and so on.

C data types may be combined quite freely, allowing you to create arrays of pointers, pointers which point at pointers, arrays of structures, pointers to structures, self-referential structures, and so on. This fact, combined with C's general economy of expression, permits you to create concise, elegant programs.

Easily Transportable

The C language does not provide input/output (I/O) functions of any kind (it has no keywords such as READ or OPEN, for instance). This is in keeping with the notion that C should not be shackled to any single computer or operating system. However, every C compiler includes a so-called standard library of common I/O functions which do tasks such as printing text, reading and writing files, and so forth. A program that uses nothing but standard I/O functions should be transportable, with little or no change, to almost any computer for which there is a C compiler. However, this ideal is rarely achieved except for very plain, generic programs, since most programmers will want to take advantage of special features unique to the host computer. To simplify the process of translating programs, C programmers often segregate machine-specific code in distinct, easily identifiable modules.

Forth

Rhett Anderson
Assistant Editor

Every programming language has its friends and enemies, but perhaps no language stirs up as much controversy as Forth.

Forth's notoriety is a result of a significant distinction between it and other popular computer languages. While most languages provide a set of standard commands and functions, Forth encourages you to extend the language itself to fit the task. It's been said that Forth is a "toolbox," and that you have to build every program from spare parts. Once you've written several programs in Forth, you've built your own set of tools to make writing future programs easier.

Forth was designed by one person—Charles Moore—to control astronomical instruments. Its speed and size, however, have made it a versatile language for microcomputers: It is fast and compact.

Forth is a low-level language. This means that when you program in Forth, you are very close to programming in the computer's native language. Because of the dense structure of Forth, large Forth programs may be even smaller than equivalent machine language programs. Forth allows easy access to a machine's hardware. As a matter of fact, it's as easy to read or write to a hardware port as it is to read or write to a variable.

Forth is often one of the first languages to appear on a computer because it is so easy to write a Forth interpreter. Sometimes there are good Forth implementations even before there are good BASIC implementations.

A Difference In Notation

People who are used to languages like BASIC and Pascal are usually shocked when they see their first Forth program. This is partly due to the fact that Forth uses *postfix* notation instead of *infix* notation.

Infix notation is the method of calculating that we learn in school. It is the notation used by Texas Instruments calculators and the BASIC language. Here is an example:

```
(1 + 2) * (5 + 4)
```

This expression evaluates to 27. Parentheses are used to specify which calculations are to be performed first. In infix notation, you always have to keep the order of operations in mind—for instance, multiplication and division are done before addition and subtraction (unless parentheses indicate otherwise). As you might imagine, computers are slowed down by taking all these rules into account.

In postfix notation (used in Hewlett-Packard calculators and the Forth language), the same expression would look like this:

```
1 2 + 5 4 + *
```

Postfix notation has the advantage of never requiring parentheses. The expression is evaluated from left to right—always.

Another characteristic of postfix notation is that numbers are put onto a *stack*. A stack is simply a place to hold numbers. It is a LIFO (Last In / First Out) structure. This means that if you put a 4 and then a 5 into the stack, you'll get the 5 back before the 4. Math is always done this way in Forth.

Postfix notation is much faster on a computer than infix notation. Most microprocessors have special commands to handle stacks. Some even have built-in hardware stacks. This helps make Forth very fast indeed.

Besides its heavy use of stacks, the critical difference between Forth and other languages can be found within the structure of Forth itself. New words (which are equivalent to procedures and subroutines of other languages) are defined as a series of pointers to previously defined, lower-level words. The path of execution of a Forth program follows these pointers from high-level words down through lower-level words until at last a *primitive* word is reached. (A primitive word is one which is defined in machine language instead of Forth.) Following this path is like untying a ball of kite string, but the computer manages the task with breakneck speed.

Some of the original arcade games were programmed in Forth, games that would be nearly impossible to program in BASIC.

Forth is fast as an interpreted language, but there are still a few Forth compilers available. However, compilers don't speed up Forth as much as they do BASIC.

Small, But Powerful

Forth's small size was especially important in early computers. Some versions of Forth have been written to use less than 4K of memory. Now that computers have more RAM, new Forth packages emphasize new features. Recent versions of Forth for the ST, Amiga, and IBM are very powerful languages which allow full file access, floating point numbers, and multitasking.

There are disadvantages to Forth, too. It can be difficult to read, write, and debug. Early implementations did not have floating point numbers, and Forth advocates are still debating whether or not the advantages of floating point numbers outweigh the disadvantages (floating point numbers are slower and use more space). Another problem is file storage. Earlier implementations of Forth could not use the standard file formats of the computers they ran on. But recent implementations have solved this problem, too.

Forth remains a popular language because it offers great flexibility. When you program in Forth, you play by your own rules—it's possible to customize and personalize it more than any other high-level language. Although some say that it's dangerous to program at this level, others demand the freedom that the Forth language offers.

Mindscape Aims To Please With New Hits For the Atari XE™/XL™ and ST™.

INFILTRATOR

Fasten your shoulder harness and seat-belt. *Infiltrator* by Chris Gray puts you at the controls of a powerful gunship ready for action. You are Johnny "Jimbo Baby" McGibbits with miles of enemy airspace to clear. You'll then convert to covert ground action, behind enemy lines, to destroy the mad leader and his hostile troops. Available on the **Atari XE/XL** only.

BOP 'N WRESTLE

Pow! Sock! Slammo! *Bop 'N Wrestle* is as close as you can get to professional wrestling without drawing blood. Put yourself into the ring with 10 of the biggest, baddest bruisers ever to perfect the turn-buckle fly. Available on the **Atari XE/XL** only.

TRAILBLAZER

Push yourself to the limit in this hyperspatial spherical grand prix. Leave your opponent in a cloud of dust as you leap over black holes, purple walls, blue bouncers, and the terrible cyan nasty zone. Not for the faint at heart. But surely an action arcade adventure in which you can have a ball. Available on the **Atari XE/XL and the Atari ST.**

BALANCE OF POWER

by Chris Crawford. You are the President of the United States. Tension always escalates in this global geopolitical simulation of the cold war's cruel reality. Test your savvy with what the *New York Times* called one of the most sophisticated strategic simulations in America. Available on the **Atari ST** only.

HIGH ROLLER

Strap yourself into a Harrier jet and enter the dangerous world of vertical takeoffs, barrel roll attacks, and heat-seeking sidewinders. Learn to fly the most sophisticated combat aircraft in this flight simulation software. Your mission is to destroy the enemy headquarters 500 miles away. Good luck! Available on the **Atari ST** only.

Mindscape
Software that challenges the mind.

MINDSCAPE

Infiltrator is available on Apple II family, IBM & compatibles, C64 & C128 and Atari XE/XL. *Bop 'N Wrestle* is available on Apple II family, IBM & compatibles, C64 & C128 and Atari XE/XL (128K recommended). *Trailblazer* is available on C64 & C128, Atari XE/XL and Atari ST. *Balance of Power* is available on IBM & compatibles, Atari ST, Macintosh, and Amiga. *High Roller* is available on C64 & C128 and Atari ST.

Visit your retailer or call 1-800-221-9884 (in Illinois 1-800-942-7315) for Visa or MasterCard orders. To purchase by mail, send Visa or MasterCard number with expiration date, check or money order to Mindscape, Inc., P.O. Box 1167, Northbrook, IL 60065. Add $3.00 for shipping and handling. Allow 3-5 weeks for delivery.

Pascal

Tony Roberts
Production Director

For those whose natural tendencies veer toward disorder, computer programming can degenerate from a pleasant occupation into a chaotic nightmare in as little as a few short GOSUBs.

Pascal, with its structure and its rules, can help keep programmers out of this trouble, as well as make their jobs more rewarding and their programs more useful.

Pascal, developed by Niklaus Wirth in the 1970s as a teaching language, is named in honor of the seventeenth-century philosopher and mathematician Blaise Pascal. Wirth reasoned that if a language's source code were clear and understandable, and if its compiler could identify errors, the language would be an efficient teaching tool. Programmers who learned Pascal would have a solid understanding of programming and could easily pick up other languages as needed.

As it turned out, Pascal grew beyond the classroom and has become a development tool for many programmers who need a language that's easier to work with than machine language, but faster than BASIC.

Anyone familiar with BASIC will see many similarities between it and Pascal. Pascal programs manipulate variables and arrays and use program-control structures such as FOR-NEXT, IF-THEN-ELSE, and WHILE-WEND loops.

Pascal differs from BASIC, however, in that it imposes specific rules about how a program is to be constructed. A Pascal program consists of the following items, which must appear in the order indicated:

- Program identifier
- Constant, type, and variable declarations
- Definition of functions and procedures
- The main program

Help Or Hindrance?

Pascal's detractors complain that the rigid rules and set structure stifle creativity; those who have grown fond of the language lean heavily on its structure for support. Using the framework required by Pascal, proponents claim, they can spend more of their time deciding how the program will *work* rather than how it will be *written*. In addition, the rules help the compiler detect and flag errors, which in turn helps the programmer debug the program.

For example, Pascal requires that all variables be *declared* before being used. Declaring a variable simply involves stating the variable name and the type of data the variable can hold. The following variable declarations might be used in an adventure game:

```
VAR
  score     : integer;
  level     : integer;
  strength  : real;
```

Now, assume that the program contains the following lines:

```
score := strength;
lever := level +1;
```

When the program is compiled, both lines will generate errors. In the first case, the variables are of different type. You cannot assign the value held in *strength* (a real variable) to the variable *score* (an integer variable) because the types are incompatible. In the second line, notice that the programmer has accidentally typed *lever* instead of *level*. Because the variable *lever* has not been declared, the compiler will report that *lever* is an unknown variable. Upon seeing this, the programmer immediately recognizes the typographical error and corrects it. This type of inconspicuous error can be elusive in BASIC.

Pascal programs are usually written as a series of subroutines that are called, as needed, by the program's main body. Essentially the goal of a Pascal programmer is to break down every task into a series of subtasks, writing a separate subroutine to handle each of the subtasks.

These subroutines, called *procedures* and *functions* in Pascal, are designed to be relatively independent. That is, they can be tested and debugged separately from the rest of the program. Making a change in a procedure or function should only affect the small task being performed by that procedure or function.

Subroutines are just like mini-Pascal programs: They consist of a name; constant, type, and variable definitions; function and procedure definitions; and a main body.

Extending The Language

Procedures and functions are the building blocks with which you both construct a program and extend the language itself. Once a subroutine works in *Program A*, you can use it in *Program B* as well. Let's say you wrote the procedure to convert a string to uppercase and print it on the screen. You might call it Print_ Up_String. This procedure could then be compiled into and used in any future program you might write. It's the equivalent of adding a new keyword to BASIC.

A mathematician might develop a set of subprograms that efficiently handle certain calculations. Someone who works extensively with data files would develop procedures for opening, closing, and reading files.

This ability to reuse completed and debugged subprograms saves time and builds the confidence of any programmer. Once you've built up a library of subroutines, you'll be more confortable attacking larger, more sophisticated projects.

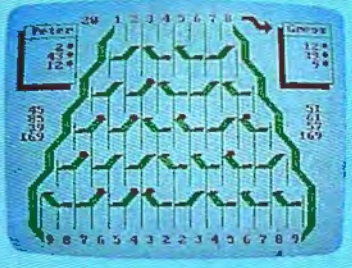
Scope Of Variables

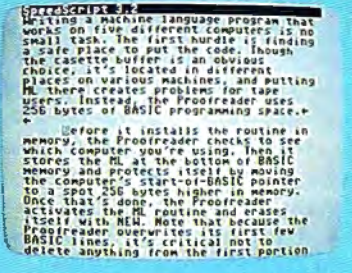
Another way Pascal differs from BASIC is in *scope*, which is the range in which a variable is effective. In BASIC, variables are *global*; that is, they can be referenced from anywhere in the program. Pascal can use global variables, but the language also makes heavy use of variables with lesser scope, and this contributes to the ease with which procedures and functions can be used as program building blocks.

When a variable is declared at the top of a Pascal program, it is a global variable, and it can be referenced or manipulated anywhere in the program. When a variable is

GET UP TO 200 FUN-FILLED PROGRAMS EACH YEAR—

when you subscribe now to COMPUTE!





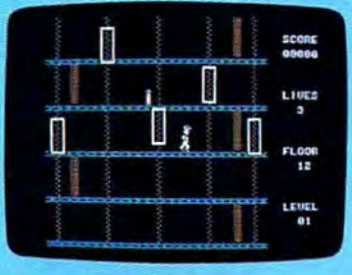
Subscribe to COMPUTE! today through this special introductory money-saving offer, and you'll be getting a lot more than just another computer magazine. That's because each issue of COMPUTE! comes complete with up to 20 all-new, action-packed programs.

Subscribe now and you can depend on a steady supply of high quality, fun-filled programs like Hickory Dickory Dock, Switchbox, TurboDisk, Home Financial Calculator, Turbo Tape, SpeedScript, SpeedCalc, and hundreds of other educational, home finance, and game programs the entire family can use all year long.

The superb programs you'll find in each issue are worth much, much more than the low subscription price.

And there's more to COMPUTE! than just exciting new programs. Month after month, COMPUTE!'s superb articles deliver the latest inside word on everything from languages to interfaces...from programming to disk drives.

Whether you're a novice or an experienced user, COMPUTE! is the magazine for you. So subscribe today. Return the enclosed card or call 1-800-247-5470.



Do it now.

ACT NOW AND SAVE!

COMPUTE!™ Publications, Inc. abc

One of the ABC Publishing Companies

If attached order card is missing, write: COMPUTE! P.O. Box 10955, Des Moines, IA 50950

declared within a subprogram, its scope is limited to that subprogram. When the subprogram is called, Pascal creates the variable and tracks it, but when the subprogram is exited, the variable is deleted and the space it occupied in memory is made available for other uses.

The same variable—the traditional loop counter *i*, for example—can be used in several subprograms without conflict. As long as the scope of a Pascal variable is limited to one subprogram, you need not worry about using unique variable names.

When To Use Pascal

Pascal can be used for any programming project, but it is most useful for programs that are regularly used. Because they're written in a compiled language, Pascal programs run many times faster than those written in an interpreted language like BASIC. However, it usually takes longer to write a Pascal program than to write a similar BASIC program.

For short utility programs that will be run once—or twice—you're probably better off sticking to BASIC, but if you're writing a program to be used daily in your business, Pascal is a good language to use. The increased speed will be appreciated by those who use the program, and you, the programmer, will appreciate the ease with which you can make minor changes as needs dictate.

Learning the elements of Pascal is relatively easy, but some of the language's more advanced features, such as the ability to create new data types, to manipulate complex data structures, and to manage large amounts of memory through the use of pointers, require a bit more study. Once you've mastered these topics, however, you'll find Pascal able to handle most programming problems.

Modula-2

Tim Midkiff
Editorial Programmer

Modula-2, introduced in 1980 by Niklaus Wirth, the creator of Pascal, was designed to provide a clear and natural way to solve common programming problems. Wirth also wanted the language to be suitable for large-scale software development and system-level programming (teams of people working on a single program).

Modula-2 provides facilities for looping, branching, and procedures that closely resemble their Pascal counterparts. The languages are very similar, so Pascal programmers can easily adapt to Modula-2.

Compiler Requirements

Modula-2 compilers require that all variables be declared. The standard data types include arrays, enumeration, subrange, sets, records, and pointers. *Type checking* is performed on the usage of variables. Type checking insures that assignments and expressions do not contain incompatible data types. As an example, assuming that *i* had been declared an *integer* and *x* had been declared a *real* variable (a variable which can include a decimal point and fractions), the compiler's type checking would flag this line as illegal:

```
x := i;
```

Type transfer functions are provided to allow operations that may be illegal under normal type checking. For example, using the same variables as in the above example and a type transfer function, this line would now be considered legal by the compiler:

```
x := REAL(i);
```

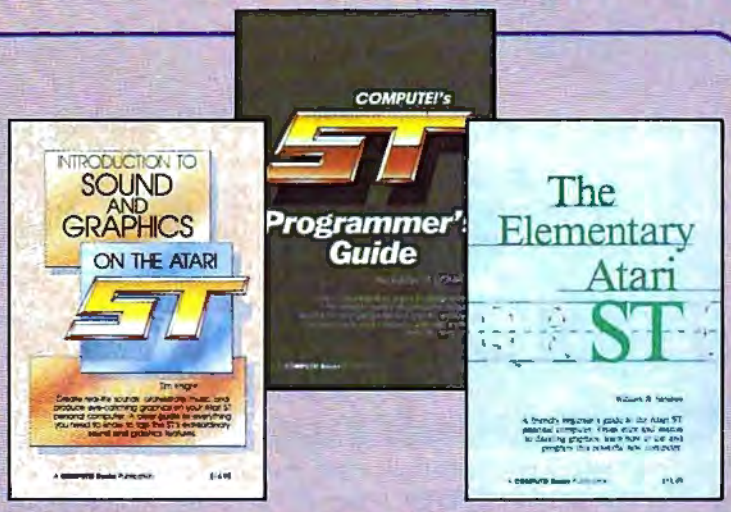
Modules and coroutines provide support for group software development. Modules are separately compiled units which allow programmers to build libraries of code that can be imported by other programs. Modules declare all procedures and data which can be imported by other programs. This facilitates *data hiding*, which prevents inadvertent changes to variables. The coroutine support also allows programs to transfer control to other programs, allowing the software to simulate multiprocessing.

The low-level support provides many functions necessary to access machine-specific features. Absolute addressing allows access to specific memory locations such as ports and system variables. There are procedures which return the address of variables or the amount of memory used to store the variable. Also, there is a procedure which allows the insertion of machine language routines by giving the opcode and operands for an instruction. This is useful to speed the execution of time-critical sections of a program. Bit manipulations are provided as part of the set data type.

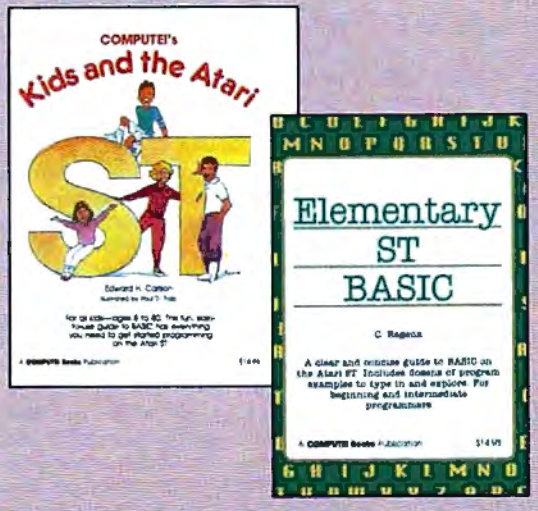
Modula-2 is a compact language with only about 40 reserved words. Much of the machine-specific programming is provided in the libraries. There are standard libraries for I/O, strings, files, coroutines, and low-level machine routines. Most implementations of the language for personal computers provide system calls as part of the libraries. This is an important consideration when you're comparing implementations, as it greatly facilitates programming. C

COMPUTE! Books' ATARI ST Collection

COMPUTE! Books offers you a superior line of titles for the new Atari ST. Packed full of useful utilities, exciting games, in-depth tutorials, and valuable applications, these clearly written books bring you fully tested information and entertainment for the whole family.



Look for these COMPUTE! books at your local book or computer store.



You can order directly from COMPUTE! by calling 800-346-6767 (in NY call 212-887-8525) or by sending your payment to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please include $2.00 per book shipping and handling for U.S. and surface mail or $5.00 for airmail. North Carolina residents add 5 percent sales tax. New York residents add 8.25 percent sales tax.

Please allow 4–6 weeks for delivery from receipt of order.

COMPUTE! books are available outside the United States from subsidiaries of McGraw-Hill International Book Company.

COMPUTE!'s First Book of the Atari ST

Edited
A valuable collection of ready-to-type-in-and-use applications, games, and utilities. Graphics utilities like "ST Doodler," games like "Switchbox" and "Tug-a-War," and educational programs like "Hickory Dickory Dock" turn your Atari ST into everything from a business graphics machine to a powerful teaching tool. Tutorials show you how to add power to ST BASIC and how to add excitement to your own creations with sound effects. *A disk is available for $15.95 which includes all the programs in the book, 203BDSK.*
$16.95 ISBN 0-87455-020-3

The Elementary Atari ST

William B. Sanders, 272 pages
A friendly, easy-to-use guide to the Atari ST, this book takes you through connecting your computer, loading programs, creating graphics and music, and writing your own programs.
$18.95 ISBN 0-87455-024-6

Elementary ST BASIC

C. Regena, 208 pages
A tutorial and reference guide to the ST's impressive graphics, animation, and sound with complete descriptions of ST BASIC's commands, syntax, and organization. *A disk is also available for $15.95 which includes programs from the book, 343BDSK.*
$14.95 ISBN 0-87455-034-3

COMPUTE!'s Kids and the Atari ST

Edward H. Carlson, 238 pages
Easy-to-understand instructor notes, lessons, assignments, and lively illustrations help both kids and adults painlessly learn to program on the Atari ST. The latest in the bestselling series by this author.
$14.95 ISBN 0-87455-038-6

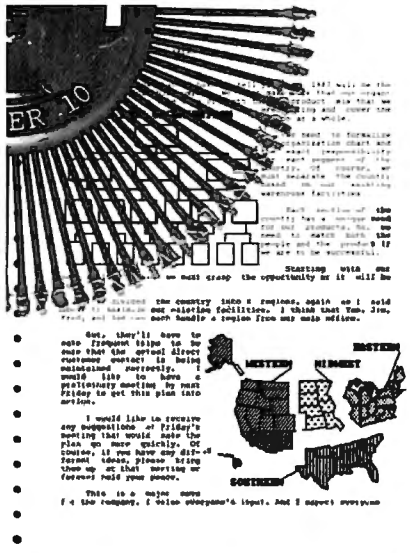
COMPUTE!'s ST Programmer's Guide

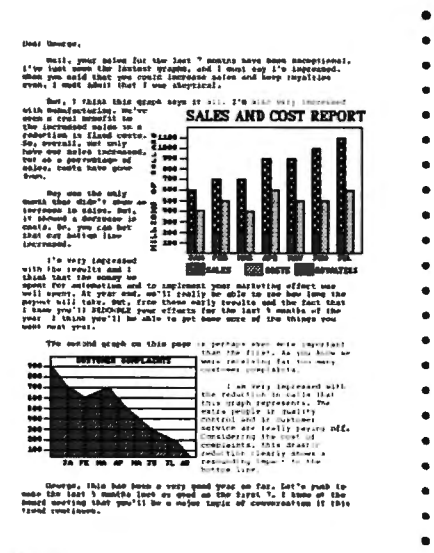
Editors of COMPUTE!, 356 pages
A comprehensive reference guide to the Atari ST, this book explores in detail Logo and BASIC, the advanced features of the ST such as GEM and TOS, and every aspect of programming from concepts to actual program writing.
$17.95 ISBN 0-87455-023-8

Introduction to Sound and Graphics on the Atari ST

Tim Knight, 197 pages
Thorough descriptions of the Atari ST's color graphics and sound abilities, plus all the information needed to create a complete sound and graphics system.
$16.95 ISBN 0-87455-035-1

COMPUTE! Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE!, COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications





For Daisy Wheel, Dot Matrix & Ink Jet Printers

$89.90 Desktop Publishing Breakthrough

Imagine using a word processing and drawing program that lets you integrate charts and pictures that you 'paint' or 'clip' into your text. Well, if you use an IBM PC or Clone, now you can have graphically dramatic documents, from business or personal letters, to proposals, to organization charts, even with a daisy wheel printer.

By Drew Kaplan

It's easy. It's impressive. And, now your thoughts can be powerfully illustrated in both words and graphics.

After all, for illustrating abstract data and thoughts, nothing beats a dramatic chart or drawing. So, let your ideas leap off the page by using integrated text and graphics. Your thoughts are sure to make an impressive impact.

Whether you write letters, bank proposals, term papers, company manuals or news letters, you can forget complicated and expensive laser printing. And, you can forget complicated expensive desktop publishing programs.

Now for just $89.90, you can use your daisy wheel, dot matrix or ink jet printer to print normal text. Plus, you can integrate simply fabulous graphs and drawings into your creations.

INCREDIBLY EASY

Savtek, a brain trust group, has developed an easy to use yet incredibly sophisticated integrated word processing and graphics program.

Just create your letters, proposals, or reports as you would with any other word processor. In fact, if you **already** have a document created in virtually any other word processor, you can 'grab' it into Savtek's instantly.

You'll produce visually powerful technical papers and manuals with drawings and charts, and dramatic marketing reports with graphs. You'll produce sales proposals with panache.

And since there's no complicated training needed (if you can run a word processor, you can run Savtek), you'll make great impressions, fast.

Anyway, once you've created the written part of your report, using Savtek's sophisticated automatic word processing features, you're ready to add pictures, charts and graphs.

Just select from the over 100 supplied changeable pictures or draw your own, using the automated ICON based drawing program.

Later, you'll learn much more about the sophisticated drawing program that lets you draw, paint, fill, expand, reduce, copy, and move your pictures.

And, you'll form squares, circles and triangles automatically. Anyone can draw with it because it's totally automated and uses arrow keys and doesn't require a mouse. But, read on.

Once you've selected a picture, the computer will produce an automatically sized box representing it. Just position the box wherever you want the picture to be in the text.

Like magic, the actual picture will appear and the text will automatically reformat itself around it.

And, speaking of reformatting, this program will automatically make page-breaks and recalculate each page as you write or edit. If you make an addition to page 1 of a 10 page report, the effect will ripple through all 10 pages.

So, whatever length you've chosen for each page (including headers, footers and automatic page numbering), will automatically be preserved.

You'll particularly like the cut and paste features of this word processing program which allow you to copy, move or delete sections of your text.

Of course, you'll have automatic Word-wrap, Hidden Hyphenation, Justified Smooth Right or Ragged Right text. Plus, you'll have Find, Replace and Search.

And look how you can format your document. There are 5 page templates called rulers which allow you to automatically set up your page.

You can select any right and/or left margins, your tabs, one, two or three line spacing, and the number of blank lines at the top and bottom of your page.

Each of the 5 rulers comes with different default settings. But, you can adjust and save them or change them and even use several at one time on a page.

HOW DO THE PRINTERS WORK?

I use a daisy wheel printer because I like my letters to look personal. I've always had to switch to a dot matrix printer for graphs and illustrations.

Unfortunately, I couldn't have my graphics on the same page as my text.

Now, because this program can use the period on the daisy wheel to create all the charts and graphic symbols you see within this ad, I don't need to switch printers any more.

And while it doesn't create the graphics as fast as a dot matrix, the quality is superb. Now my graphics can be impressively integrated into my text.

Note: Every single sample page shown in this ad, was printed out on my EXP 400 Silver Reed daisy wheel printer.

Note: This program does not produce two column news letters in a single action. Simply create a double length column and cut it when you have it printed.

No matter what printer you use, daisy wheel, dot matrix (with or without near letter quality printing) or ink jet (color or single color), you'll have powerful looking documents to really present your ideas in the most professional manner.

DESKTOP PUBLISHING

Desktop publishing is about the hottest category of computer programming. It seems that everyone has discovered the impact of combining text and graphics.

And very impressive presentations are just what Savtek's ETG Desktop publishing system provides for you.

Imagine leveraging the capabilities of your own IBM or Clone, your own printer and your own keyboard to produce the documents you see on these pages, with nothing else to buy.

THE 1000 WORD PICTURE

First a confession. I can't draw. That's why you don't see drawings in DAK's catalogs. But I've been amazed at how creative I can be with this paint program.

It's easy. You do everything with the arrow keys and the return key. By using the arrow keys you can draw in any direction with a choice of 12 brush shapes.

There's an erase function to eliminate anything you don't like. And here's my favorite function. UNDO is a function that works throughout this program.

. . .Next Page Please

. . .Publishing Continued

It simply removes the last thing you did. So, no matter what you do wrong, you're a button away from removing it.

If you don't want a solid line, just spray an area. It's like using a spray can.

Let's say you want to connect two points with a straight line. Use the Angle Line. It produces a computer generated straight line between any two points.

What if you want a circle? Just touch the return key. Then use the diagonal arrow key to enlarge or reduce the circle. If you use the up/down or right/left arrows, you'll get an ellipse.

In the same way you can create squares, rectangles or triangles. And you'll be amazed how many things, from houses to technical drawings, are made up of squares, rectangles, circles and triangles.

But, that's not all. You can choose any of 32 background patterns to fill in enclosed areas or broad lines. And if 32 isn't enough, you can design your own.

There's so much more. You can juggle a picture. Imagine, turning it over or sideways with the touch of a button.

You can copy or move a picture or even part of a picture right on the screen. So, draw it once and copy it or move it.

But, here's my favorite. You can enlarge or reduce any picture or part of a picture right on the screen. So you can change its size equally, or you can stretch it out or make it tall and thin. Wow!

There are 12 included font/sizes. So you can have large or small type in your choice of styles within a picture or integrated with your text.

And, each of the 12 font/sizes can be shown on the screen and printed normally, in bold, in italic, in outline, or in shadow. Plus, you can write normally across the page, up the page, down the page or upside down.

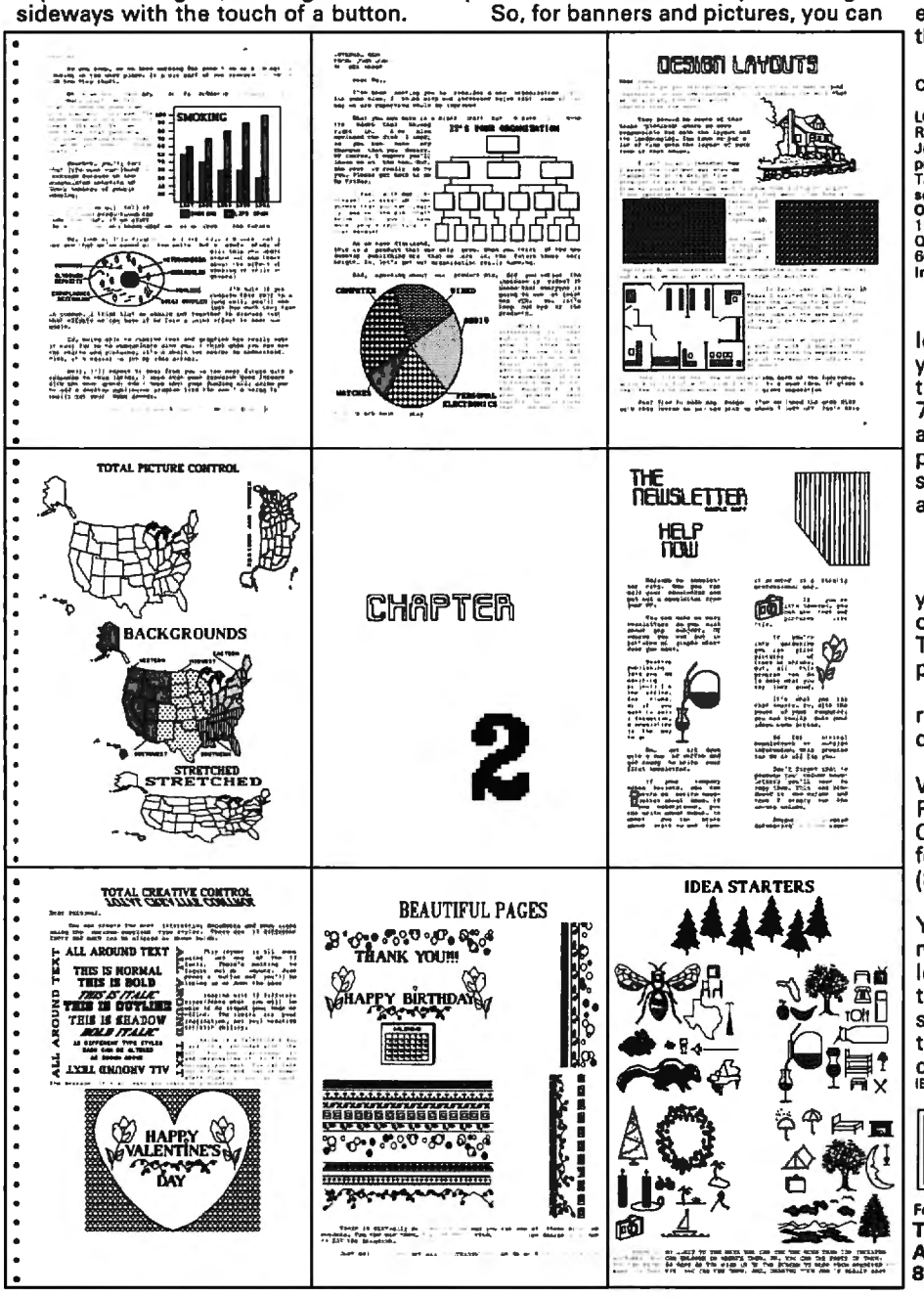
Finally, you can zoom into any small section of the screen and edit your pictures, pixel by pixel. With this kind of power, you don't need to be an artist, just have the ability to push a button.

You can operate this Paint program independently. Or, you can access any picture from within word processing.

So, for banners and pictures, you can print directly from the Paint Program. Or, for everything previously described, simply access your pictures, captions, graphs or charts through the desktop publishing section.

This program is incredibly powerful, yet you'll be comfortable using it within just a few hours.

Every picture in this ad was created with this program. And, you haven't even seen the tip of the iceberg of its capabilities. For example, if you have a picture on the screen, you can bring a second picture up and join them together.



WHO CAN USE THE SYSTEM

All you need is an IBM PC, AT, XT or 100% compatible with standard IBM CGA or EGA graphics capability. It must have at least 256K, and either two floppy disk drives or one floppy and a hard disk.

Below is a list of some of the dot matrix, ink jet and daisy wheel printers that have been tested with this program. If your printer is compatible with any of these printers, it should work too.

Special Note: Most daisy wheel printers are Diablo 620/630 compatible, so they will work with this program.

Special Note: With a color printer you can print 3 colors plus black text.

C.Itoh 8510, Epson Fx-80, FX-85, FX-185, JX-80 (color), LQ-800, LQ1500, LX80, MX80 with Graftrex Plus or Graftrax, RX-80, Hewlett Packard 2225C Think Jet or QuietJet, LaserJet, or LaserJet Plus, IBM 80CPS Graphics Printer, IBM Proprinter, IBM 3852 Jetprinter (color), Juki 6100, Mannesmann Tally Spirit 80, NEC 3500, 3510, 3520, 3530, 3550, 5500 series, 8023A, NEC Pinwriter P5XL, P6, P7, (single or color), OKIDATA Microline 92, ML92, w/IBM Plug & Play, Microline 193, 20 (color), Panasonic KX-P1091, KX-P1091i, Quadram Quadjet (color), Radio Shack DMP-200, Silver Reed EXP 400, 600, 800 and all EXP series, Star Micronics SG-10, Texas Instruments 855, 865, Xerox (Diablo) 620, 630.

FINAL FACTS

There's a pop down calculator which lets you deposit your results right into your text. A clock/timer picks up the time from your computer, and there's a 7,300 year calendar. They are all available as pop-down windows. Savtek's program is backed by a standard limited software warranty /license. It comes with a superb, easy to use reference manual.

DESKTOP PUBLISHING FOR YOU RISK FREE

Make your ideas explode in front of your readers. When you send out a letter or proposal, let it be really dramatic. They will be your ideas, but you'll be presenting them like never before.

If you're not 100% satisfied, simply return it in unused condition within 30 days for a courteous refund.

To order Savtek's ETG Integrated Word Processing and Graphics Desktop Publishing System for your IBM PC or Clone, call toll free or send your check for the breakthrough price of just **$89.90** ($4 P&H) Order No. 4801. CA add tax.

Look at the 12 sample pages I created. You'll see graphs, pictures and charts mixed into my text. I even designed a logo for my newsletter. Just think about the impact you'll make when you present your ideas with a combination of text and graphics. And oh, it's so incredibly easy to use.

IBM is a registered trademark of International Business Machines.

DAK INDUSTRIES INC.

Call Toll Free For Credit Card Orders Only
24 Hours A Day 7 Days A Week

CP65

1-800-325-0800

For Toll Free Information, Call 6AM-5PM Monday-Friday PST

Technical Information. . . .1-800-272-3200
Any Other Inquiries. 1-800-423-2866
8200 Remmet Ave., Canoga Park, CA 91304

Bold Presentations

Unleash the dramatic presentation power of your IBM PC or Clone, with Quadram's 7-Color Ink Jet Printer. You'll produce attention grabbing professional graphs, charts, and drawings with the complete ADDED software DAK has put together. It connects instantly. And, you get EVERYTHING, a $1201.90 value, for just $499. WOW!

- **Quadram's $495 Color Ink Jet Printer**
- **DRI's New $299 Gem Draw Plus**
- **Softkey's $149 Keychart**
- **Savtek's $89.90 Desktop Publishing**
- **DAK's $49.90 DAKGraf**
- **Logitech's $119 LogiMouse Plus**

By Drew Kaplan

Let your ideas explode in front of your readers. Demonstrate what you want to say with dramatic charts, graphs, pictures and even technical illustrations.

If you're like me, columns of numbers are boring. Now you can effortlessly produce sophisticated, multicolor graphs and charts that will rivet your reader's attention to your presentation.

Turn your thoughts, calculations or sales results into a profusion of color that simply can't be ignored.

This Quadram 7-Color, virtually silent, ink jet printer, and all the software **I have added**, will let you make your most complex concepts easy to understand.

And, it will present your easy to understand numerical data and/or visual concepts in an incredibly impressive manner.

WHAT YOU NEED

You'll need an IBM PC, XT, AT or Clone with 256K of RAM, except for Gem Draw Plus which requires at least 384K. You need two floppy disk drives or a floppy and a hard disk.

And, all these programs require only standard IBM CGA Graphics (EGA is great of course). Plus, you don't even need a color monitor to use them.

WHAT IS INK JET PRINTING?

In a few minutes you'll learn how easy it is to instantly create multicolor Pie, Line and Bar Charts with DAKGraf. With Keychart you can quickly create sophisticated Hi/Low Stock market Charts, Scatter Charts, Exploded Pie Charts and a myriad of presentation charts from easy to follow menus.

If you want to Paint, you can create all types of dramatic pictures and text automatically with Savtek's Desktop Publishing Software.

And finally, Digital Research's new Gem Draw Plus lets you create the most sophisticated drawings ever. It is object oriented so you can even lay shapes on top of one another and then move them, expand them, reduce them or copy them.

Plus, I've included Logitech's C7 Serial Mouse with Plus Software. So, not only can you use this mouse for creating drawings, just wait till you use it for Lotus or any other mouse supporting programs.

Logitech says that the Plus Software with this mouse can increase the productivity of even the most experienced 1-2-3 user by up to 30%.

In virtually every computer magazine I see, Logitech sells this mouse direct for $119. Wait till you see all it can do for your productivity.

But, let's look at the printer. It simply plugs into the parallel output of your computer. And, don't worry about replacing your printer.

I've got an inexpensive option that lets you switch from your regular printer to this silent 7-color ink jet printer with the push of a button.

Anyway, speaking of 7 colors, let me tell you what they are and how easy they are to use. First, there are no messy ink trays or bottles. The ink is supplied in 2 sealed, plug-in plastic packs **(Included)**.

One pack supplies 4,000,000 characters of black. The second pack supplies 3,000,000 characters of yellow, blue (cyan), and red (magenta).

These are the industry standard basic colors. The other colors are obtained by the printer using two colors at once.

So, you'll print in black, yellow, red, blue, green (yellow and blue), orange (yellow and red) and purple (blue and red). You'll simply have to see the colors to see how really vibrant they are.

And speaking of vibrant, if they aren't sharp, they aren't vibrant. And, this printer can lay down 640 single dots per line at a speed of 2600 dots per second.

You'll create incredibly sharp detail, even in multicolors. Just look at the main picture above to get a minuscule idea of what this printer is capable of producing.

You can produce your graphs, pictures, drawings, or text (of course it has a full ASC11 character set, plus enlarged type) on just about any material you choose, from letterhead, to plain paper, to roll paper, to overhead transparencies.

Quadram's printer has all the expected features, including Line Feed, Form Feed, paper error/end and two self-test modes. But basically, it's control free.

Just plug it in, put in the paper and print! It's virtually maintenance free because the ink jets are self cleaning.

As you'll see from the following software, there's almost no limit to the presentations you can create with this printer.

And please note: I went to 5 separate companies to gather this software and mouse. This isn't cheap bundled software.

Just price Gem Draw Plus at $299 and the Logitech Mouse at $119 at any computer store, and you'll see just what an incredible value this system is.

We had to guarantee not to sell the

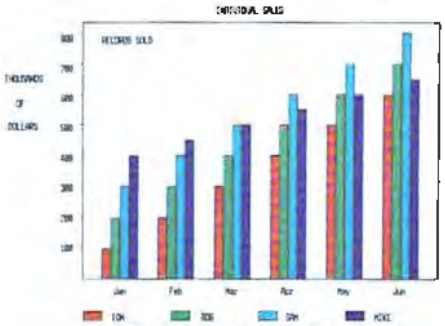
. . .Next Page Please

. . .Presentations Continued software separately and we had it shipped to us without the normal fancy retail decorative binder/boxes.

Note: I understand that Keychart was once sold with Quadram's printer, but we bought it direct from Softkey.

Anyway, other than the boxes, you get the same complete programs. There are drivers for dozens of other printers, every word in every manual is included, and every part of each program is included.

Now, let's take a look at all the things you can do. If you're already familiar with any of the software, you'll know that the following descriptions are simply the tip of the iceberg of the presentation graphics resources that you'll command.



$49.90 DAKGRAF

Now you can create powerful, vibrantly colored graphs that really get your meaning across. You'll be producing graphs in seconds. I find it easier to spot trends with a graph than with a spreadsheet.

Let's make a simple one element graph. Just enter up to 12 numerical values (called observations). Then touch G. Then select a Pie, Line or Bar graph.

Instantly you'll see a full preview of your graph. But, there's lots more.

If you touch ESC, you can re-select another type of graph and it will instantly appear, using your existing data. So, you can choose the type of graph you want.

Anyway, if you've selected Line or Bar charts, the tick marks and values (called scaling) that you see on the left side of most graphs, can be added automatically.

Across the bottom of your chart, you're going to want to assign names to the 12 observation points we've chosen.

The names may be people, things or months. Just touch ALT M, and the Months will appear automatically. Finally, you can add or move floating captions.

Pie charts are also very easy. When the pie chart prints out, it can either show the actual number beside each slice or the percentage of the whole.

So far, we've made a simple one element graph. But for Line and Bar charts, you can have 4 elements. So, at each observation, you'll have up to 4 different colored bars or lines. You can compare profits, costs, sales and salaries.

Below the graph, a box with the appropriate color is printed. Just type a title next to it to identify each line or bar.

Of course, you can save your graph. You can recall it and change it. You can even change it from a bar, to a line, to a pie, in any order you like. And, you'll be running this program faster than it took to read this description.

When you print out your color graphs, you can choose from 4 different sizes.

$149 KEYCHART

Here's the ultimate charting presentation program. It can run circles around the DAKGraf. Not only can you create regular charts, you can explode pie slices and print in any of 7 different hatch designs in addition to the 7 colors.

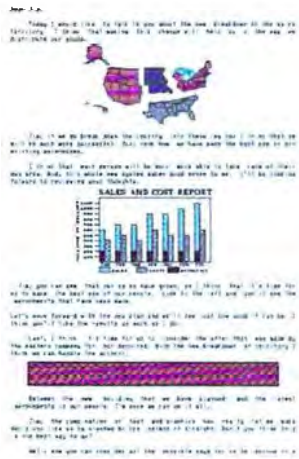
But simple Pie Line and Bar charts are too easy for this program. It's very easy to use and completely menu driven.

In addition to being able to preview your graphs on screen, you can even edit your chart's size, location or you name it. Look at all the charts you can produce.

You can make Bar Charts, Clustered Bar Charts, Stacked Bar Charts, Horizontal Bar Charts, Line Charts, Combination Bar and Line Charts, Pie Charts, Multiple Slice Exploded Pie Charts, Scatter Plots, Combination Line and Symbol Charts, X-Y Charts (Business & Scientific Formats), High-Low-Close-Open and Volume Stock Charts, Area Fill Charts, Regression Analysis Charts, Log-Log and Semi-Log charts, Text Plots, Combination Chart and Text Plots.

All these charts may look complicated, but they are really easy to use. There's a standard template menu and all you do is fill in the blanks.

And look at this. You can import electronic spreadsheet information from programs such as Lotus®, MultiPlan®, SuperCalc®, VisiCalc® and more. So, you can fill in the blanks or import the data.



$89.90 DESKTOP PUBLISHING

Savtek's Desktop Publishing program is incredibly easy to use. It's an integrated text and graphics program.

So, you can combine your text and pictures on the same page, but not on the same line.

Let me tell you about the paint program. It is absolutely a dream to use. It comes with over 100 predrawn pictures, but creating your own is easy.

And, if you're not an artist, don't worry, everything is automated. It uses the arrow keys to draw. It's very fast and you'll be amazed at how easy it is to use.

You can draw lines in any of 12 different width/styles. And, you can instantly erase them with the Eraser function.

If you don't want a line, use Haze. It's like using a spray can. It lightly mists the area you move over. The more you go back over an area, the more dense it becomes. It's great for shadows and fill.

Want a straight line? Just choose two points and 'Line' will do the rest.

What if you want a circle? Just touch the return key. Then use the diagonal arrow key to enlarge or reduce the circle. If you use the up/down or right/left arrows, you'll get an ellipse.

In the same way, you can create squares, rectangles or triangles. And you'll be amazed how many things, from houses to technical drawings, are made up of squares, rectangles, circles and triangles.

Now lets have some fun. This program is incredibly powerful. Let's say you've created a square. You can pick it up and move it anywhere on the screen that you wish by using 'Move'.

Let's say you really like your square. Use 'Copy', and you can make as many copies as you like. And you can place each copy wherever you like. This is really great for organization charts or anything where you need repetitive shapes.

But, we're not even close to being finished. What if you don't like the size of what you've drawn?

'Vary Size' lets you enlarge or reduce any object on the screen. You can even stretch it out or make it tall and skinny. I streched out a map of the United States.

And, there's still more. You can juggle a drawing on the screen. You can turn it over, around or sideways.

Finally, you can Zoom in on a section and 'operate' on it pixel by pixel for infinite control of each dot.

OK, now for some thoughts on color. There are 32 different fill patterns. You can see them in the main picture next to my map of the United States. Each of these patterns can be altered.

You can create your own patterns. So, you can make your logo, happy faces or you name it. Whatever you choose, you can automatically fill in any closed area.

This program will allow you to paint in 3 colors at a time. You can draw in black, blue and red, or yellow, blue and red.

You can also form patterns that are combinations of the colors for even more variation. There's also a provision for full size and half size picture printing.

My favorite part. OK, I'm not the world's greatest artist and I make mistakes (lots of them). There's an UNDO command. Whenever I mess up, I just touch UNDO, and my last action is instantly undone.

Finally, there are 12 font/sizes so that you can have headlines, titles or text within any of your drawings.

And, each of the 12 font/sizes can be shown on the screen and printed normally, in bold, in italic, in outline, or in shadow. Plus, you can write normally across the page, up the page, down the page or even upside down.

This is an incredible program. And, don't forget, you can integrate the pictures into its sophisticated word pro-

. . .Next Page Please

. . .Presentations Continued

cessing program instantly.

$299 GEM DRAW PLUS

Now, let's get really serious. While Savtek's paint program is superb, I think Gem Draw Plus is the most sophisticated drawing program in the industry.

Its power is virtually unbelievable. Making squares, rectangles, circles, arcs, and polygons, are mere child's play.

Expanding, shrinking or copying elements of your picture is accomplished with the click of a mouse (more later).

You can draw with up to 16 colors (this printer will print 8 including white). Each color is numbered for use on non-color monitors.

And, you can use 35 fill patterns. And each can be used with any color.

Gem Draw Plus is 'object oriented' rather than pixel or 'screen' oriented. It understands what you want to create, so it keeps the components (It calls them 'elements') separated in its memory.

So, for example, if you overlay a circle with a square, they mix on the screen, but not in the memory.

You can put one behind the other and then switch them. This program never forgets the objects you're working with.

If you design a house, you may make a toilet. Then you may make 10 copies. Later you may want to make them smaller.

Just gather them all together into a 'group', reduce them, and then 'ungroup' them and put them back wherever you want. It's great for architects, engineers or designers.

It's particularly strong for finance, manufacturing and higher education. Of course, you can draw sophisticated pictures just for fun. But, whatever you draw will be technically perfect.

And perfect is an understatement. Look at all the ways you can align the elements of your picture with just the click of the mouse. You can: Put In Front, Put In Back, Make Group, Break Group, Align Left, Align Center, Align Right, Align Top, Align Middle, Align Bottom, Page Center, and Even Spacing. Wow, all this is from just one pull down menu.

The Make Group and Break Group is incredible. If you've created a number of parts to your picture, a single command lets you combine them so you can do something to as many of them as you've chosen. Or, you can separate elements and act on each individually.

The list of drawing aids goes on and on, including auto-grid, and I certainly can't cover them all here. But look.

Let's say that last week you created a drawing. It could be an electrical schematic, an organization chart, or a forest full of trees. Now you want to create a new drawing. But, you want to 'pick up' some of the parts of your old drawing.

After all, the best part of computers is that you never have to do the same work twice. Well, Gem Draw Plus allows you to bring up two separate screens at one time. So, for example, you can have your new picture on the left and the old picture on the right side of the screen.

OK, now it gets exciting. You can 'drag' an element from your old picture across the screen into your new picture.

Wow, so if you do repetitive types of work, you can instantly pick up parts from old pictures to save yourself time.

Of course, you can alter the element you've moved just as if you'd just drawn it. And, you can move something you've just drawn back to the old picture.

Gem Draw Plus gives you incredible power. And its graphics are especially compatible with Ventura® Publisher.

There are multiple line sizes with choices such as arrows, straight or rounded endings. There are different size fonts, of course. There's a library of artwork and there's a Shadow Command that gives any object a 3-dimensional look.

Gem Draw Plus comes complete with GEM Desktop which is a utility program that provides the 'Gem Environment'.

$119 LOGIMOUSE PLUS SOFTWARE

It's keyboard freedom when you plug Logitech's C7 serial mouse into your RS232 serial port. I've included it because it makes Gem Draw Plus so easy to use. But, you'll use it all the time.

Just plug it into your serial port and get ready for super productivity. This is the advanced version with special software that really speeds up your work.

The Point-and-Click software make the popular Lotus 1-2-3 work like a mouse-based application. It fully integrates the mouse, making it easier to create and edit spreadsheets. With Logimouse and its Program, you can scroll to different sections of your spreadsheet and move quickly from cell to cell.

It has its own time-saving pop up menus, which you can customize to meet your needs. With its Point Text Editor, you can open many overlapping windows on the same or different files.

You'll find that a mouse added to your keyboard will make your work infinitely easier in lots of programs.

What is a mouse? Well, it's very simply a small device you move on your desk. As you move it, it causes the curser on the screen to move.

It replaces keyboard commands and is incredibly fast. It let's you be really productive. When the curser is where you want it, simply click (touch) a mouse button and your computer will react.

You will find more and more programs supporting mice because they are incredibly easy to use.

I think you'll have a hard time matching the quality of the Logimouse. And when you add the Plus Software, I can't match the productivity at any price.

WHY SO CHEAP

This system will come to you in just a few boxes. But, it took me over two months to assemble the software. I had to work with 5 separate companies, plus Quadram, to make it complete.

The problem was really very simple. Nobody at Quadram knew what software was availble for this printer.

So they were sitting with 4200 printers. I bought all 4200 for a song and put together this package.

The only reason that the price is so 'cheap' is because I got a ridiculous price from Quadram. They are a large company with lots of other products, and 4200 printers wasn't worth their effort.

From Boston to Toronto to Silicon Valley, I've covered this continent to put together this system. **Now, it's easy for you to use because you get everything.**

Of course, if you just wanted to print text, virtually every word processing program works great. But you'd be wasting the incredible presentation power of this remarkable 7-color ink jet printer.

It's backed, as is each separate software package and the mouse, by each of the 6 individual manufacturers' standard limited warranties.

BOLD PRESENTATIONS RISK FREE

Don't say it with words. Demonstrate your ideas with vivid colors. Show graphs to make your point. Design presentations to knock the socks off of prospects. Or, simply draw beautiful pictures.

If you aren't 100% awed, simply return it in its original boxes within 30 days for a courteous refund.

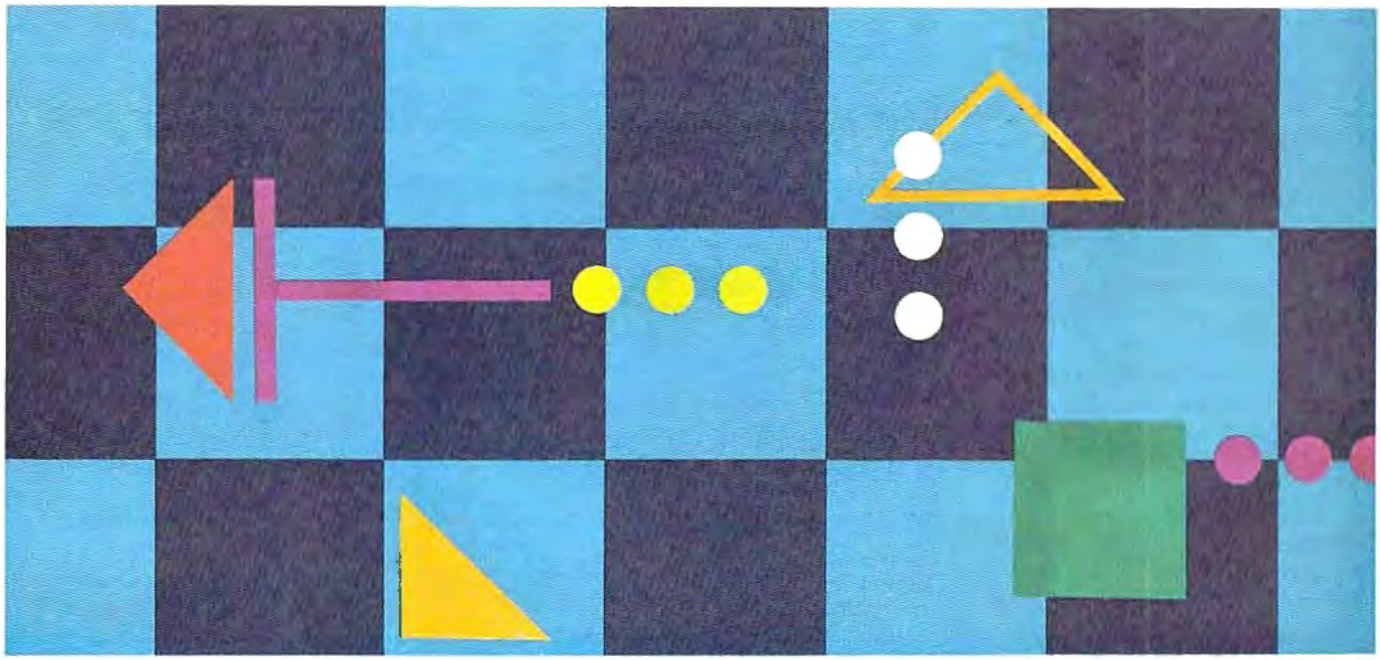
To order the Complete Presentation package including the $495 Quadram 7-Color Silent Ink Jet Printer, DAK's $49.90 DAKGraf, Savtek's $89.90 Desktop Publishing Program, SoftKey's $149 Keychart, Digital Research's newest $299 Gem Draw Plus, and Logitech's $119 Logimouse with Plus Software for a total $1201.80 retail value, risk free with your credit card, call toll free or send your check for just **$499** ($14 P&H) Order No. 4811. CA res add tax.

Special hookup bonus. Now you can switch between your current printer and this printer instantly. Just unplug the cable from your parallel printer and plug it into this box. Two identical cables are supplied that connect our switching box to both of your printers. It's just **$49.90** ($3 P&H). Order No. 4813.

A box of 4 125' rolls of 8½" wide paper is **$29.90** ($3 P&H) Order No. 4486.

Extra 4,000,000 character Black Ink Packs are just **$12.90** ($1 P&H) Order No. 4484. Extra 3,000,000 character Yellow/Red/Blue Ink Packs are just **$14.90** ($1 P&H). Order No. 4485.

You never get a second chance to make a good first impression. With this system, every impression will be dramatically bold and 100% professional.



Laser Chess™

Mike Duppong

Here's a game that's so good that we just had to share it with COMPUTE! readers. Laser Chess™ *won First Prize in our $10,000 programming contest for COMPUTE!'s Atari ST Disk & Magazine. Awarded $5,000 for its originality and skillful programming,* Laser Chess *is a two-player strategy game patterned after traditional chess—with some fascinating new twists. The original version was written in the Modula-2 language for the Atari ST. Here we have provided BASIC and machine language translations for the Amiga, Commodore 64 (and Commodore 128 in 64 mode), Apple II, and Atari XL and XE. The Amiga version requires at least 512K of memory. At least one joystick is required to play the Commodore 64 and Atari versions. The Apple II version runs on any Apple II–series computer, with either DOS 3.3 or ProDOS.*

Laser Chess™, as the name implies, is a chesslike strategy game for two players. The goal is to manipulate a laser-firing piece and various reflective objects to eliminate your opponent's king. As in traditional chess, there are an infinite number of ways to accomplish this.

Refer to the special notes for your computer; then type in and save a copy of the appropriate version. Be sure to read the general game rules before you play.

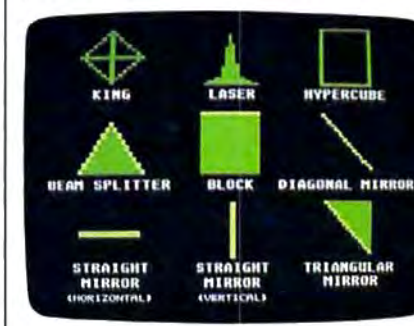
There are eight basic types of pieces in *Laser Chess*, and each has unique capabilities. Over time, you'll learn each piece's advantages and limitations. Obviously, the more you play *Laser Chess*, the more you'll understand the pieces in your arsenal, which in turn will make you a better player. So let's start with a description of the pieces.

A Geometric Army

Figure 1 shows each piece and its name. (The appearance of game pieces differs slightly in some versions; see the photos for your particular computer.) Notice that some sides of certain pieces are highlighted with a different color. This indicates a reflective surface. When a laser beam strikes a reflective surface, it bounces off without harming the piece. But if a piece is hit by a laser on a nonreflective surface, it is destroyed.

A piece can also be removed from the board if it is captured by an opposing piece. This is similar to traditional chess; to capture a piece, you simply move one of your own pieces onto its square.

Figure 1: These are the basic pieces in *Laser Chess*.



In addition to their ability to move from square to square, pieces with reflective surfaces can also be rotated in place in 90-degree increments. This lets you orient the piece to protect it against opposing laser shots, or to set up bounce shots with your own laser.

The *king* is the most important piece in *Laser Chess*. When the king is eliminated, the other player wins the game. Since it has no reflective surfaces, it can be destroyed by a laser from any angle. It can also be captured by an opposing piece. The king is not totally defenseless, however. It can capture any opposing piece by moving onto its square.

But you can use it for a capture only once per turn.

The second most important piece is the *laser*. This piece is your primary offensive weapon; it's the only piece which can fire a laser shot. To take aim, you can rotate it in place at 90-degree angles. Like the king, the laser is completely vulnerable to enemy laser strikes, because it has no reflective surfaces. If you lose your laser, the game is not over, but only the most skillful (or incredibly lucky) player can overcome its loss.

Tricky Pieces

The *hypercube* is an interesting piece. It can't harm an opposing piece directly, but may very well do so indirectly. When the hypercube is moved onto another piece (even your own), that piece disappears from its original position and reappears on a randomly selected empty square. This can happen only once per turn. The hypercube can be a two-edged sword; it may relocate a piece to a vulnerable position, or it may make it possible for the piece to capture an important opposing piece on the next move. The hypercube has no reflective surfaces and cannot be rotated. It is invulnerable to laser shots, however, because it's made of transparent material—a laser beam passes right through it. Remember that.

The *beam splitter* is another tricky piece. When a laser beam strikes a splitter's vertex (the point opposite its base), the beam splits in two. The two new beams travel in opposite directions, perpendicular to the original beam's path. (See Figure 2.) When a laser shot hits one of the beam splitter's reflective surfaces, it bounces off at a 90-degree angle *without* splitting. If the beam splitter's base is hit by a laser shot, it is destroyed. The beam splitter can be rotated.

The *blocks* are fairly simple pieces. However, they may impose some complex situations. A block can capture any opposing piece by moving onto that piece's square, much like a king. But unlike a king, a block has one reflective side and can be rotated as the situation demands. Therefore, blocks can be used either offensively or defensively. A laser beam that hits the reflective surface of a block is deflected 180 degrees—bouncing the beam back where it came from.

Figure 2: As seen in this magnified view, a beam splitter's vertex reflects a laser shot in two perpendicular directions.

A *diagonal mirror* cannot be destroyed by a laser, because both of its surfaces are reflective. Diagonal mirrors can be removed from the board only when captured by a block or a king. When a laser beam strikes a diagonal mirror, the beam is deflected 90 degrees. Diagonal mirrors can be flipped to their opposite diagonal, but cannot be rotated to face horizontally or vertically.

The *horizontal mirrors* and *vertical mirrors* (known collectively as *straight mirrors*) are also invulnerable to lasers due to their reflective surfaces. When a laser hits a straight mirror on its flat surface, the beam is deflected 180 degrees. If the laser hits a straight mirror edgewise, the beam passes straight through it. (Look closely at Figure 2; a laser beam is passing through a horizontal mirror just to the left of the red beam splitter.) Straight mirrors can be rotated to become either horizontal or vertical mirrors, but not diagonal mirrors.

The *triangular mirrors* deflect laser beams just as diagonal mirrors do, but they are vulnerable to hits on their two nonreflective sides. A triangular mirror can be rotated in 90-degree increments.

Making Moves

As in the conventional game of chess, a move in *Laser Chess* consists of moving or otherwise manipulating a game piece. The specific notes for your computer explain whether a move is made by means of a mouse pointer, a joystick, or keyboard controls. Every version uses a color change of some sort to indicate whose turn it is; for instance, in the Amiga version, the border of the game board changes color after each turn.

Figure 3: This full-screen view of Laser Chess shows its 9 × 9 board grid and game controls.

The same player always moves first in *Laser Chess*. There's no particular advantage or disadvantage to moving first.

A turn consists of two moves. The number of moves remaining in a turn is indicated visually on the screen. (See Figure 3.)

Before you move or rotate a piece, you must select it. The instructions for your version explain how this is done. When a piece is selected, the appearance of the piece (or the cursor, in some versions) changes.

If you accidentally select the wrong piece, you can deselect it by the same means used to select it, as long as you're still in the same square. Deselecting is usually done after rotating a piece—more on this in a moment.

After you've selected a piece, your next decision is whether to move or rotate it. Moving a distance of one square takes one move; moving two squares takes two moves (although you can move a piece two squares in one step). Since you have only two moves per turn, the maximum distance a piece can be moved in one turn is two squares. The computer does not allow illegal moves.

Pieces can be moved forward, backward, left, or right, but not diagonally. You can effectively move a piece diagonally by using two moves—forward and right, for instance.

You cannot move a piece onto a square occupied by another piece. The only exceptions are captures

with blocks and kings, and moves of the hypercube as described above.

Rotating A Piece

The instructions for your computer's version explain how to rotate a piece. The computer does not allow you to rotate a piece that's incapable of rotation. Otherwise, the piece rotates 90 degrees (one-quarter turn) clockwise. You may continue rotating the piece to any desired position before deselecting it. Rotating a piece to face any direction takes only one move, and the move is subtracted after the piece is deselected. If you deselect the piece in its original position, no move is subtracted.

You can combine a rotation and a move in a single action (except in the Atari XL/XE version). First, select the piece. Then rotate it to the direction you wish it to face. Finally, move to any adjacent square (except a diagonal) as you would normally do. The piece moves to that square and faces in the direction you've chosen. Since rotating a piece and moving a piece each take one move, this uses up your turn.

Special Features

At the center of the 9 × 9 board is a special square called a *hypersquare*. It absorbs laser beams and acts like a stationary hypercube. That is, if you try to move a piece onto it, the piece disappears from its original position and reappears on a randomly selected empty square. This can happen only once per turn, however.

Along the board on the left side of the screen are some geometric button shapes. The button labeled Q allows you to quit playing at any time. When selected, this option requires that you confirm your decision.

The restart button (R) lets you start a new game without finishing the current game. (For instance, a player may be so hopelessly behind that he or she wants to resign.) Again, the program asks that you confirm this choice.

Firing The Laser

The last button is the laser trigger. When it's your turn, you can select this button to fire your laser. If your laser piece has been captured or destroyed, the laser button won't appear on the screen during your turn.

Figure 4: The combination of reflective and transparent surfaces of the various pieces can result in complex bounce patterns. Here, the red laser takes advantage of the green beam splitter to destroy two blocks.

Figure 4 illustrates the general effect of firing the laser (some versions differ slightly in appearance). Firing your laser takes only one move, but can be done only once per turn. Therefore, you may want to use your first move in a turn to aim the laser, rotate a reflecting piece to set up a bounce shot, or move another piece into position.

Of course, you won't necessarily be firing the laser on every turn. Much of the strategy in *Laser Chess* involves moving and rotating your pieces to set up complex shots. It's important to realize that *any* laser hit on a piece's nonreflective or nontransparent surface will destroy that piece. You can destroy your own pieces just as easily as you can destroy your opponent's. You can even zap your own laser, particularly if you fire directly into the 180-degree reflective surface of a straight mirror or block, or if you fail to anticipate the effects of a beam splitter. Be forewarned.

Laser Chess Strategy

As in the conventional game of chess, much of the strategy in *Laser Chess* revolves around thoughtful placement of your pieces. However, the character of the game differs from that of chess in many ways. The laser, for example, can strike at long distances and in more than one direction at once. And the hypercube adds an extra element of uncertainty. The best strategy for any particular game depends to a great extent on the skill and personality of your opponent. However, here are some general tips you may find helpful.

Get your mirrors out early. Use them to gain the fullest potential of your laser. Try to position mirror networks on both sides of the beam splitter so you can inflict as much damage as possible.

Take advantage of the blocks. Since they "control" an area around them with their threat of capture, no other pieces can safely move within their range. Make your opponent work to displace them. Remember to rotate the reflective side of a block to the most probable direction of laser fire. If you can prevent a laser from destroying the block, your opponent will most likely have to gang up on it with two or more of his or her own blocks.

Use mirrors to protect your king. If you surround your king with straight and diagonal mirrors, there is no way it can be hit by a laser. Therefore, your opponent will have to break through your defense with blocks. (This is a pretty dirty trick, because when all of your opponent's blocks have been used, your king is almost invulnerable.) Defending your king with blocks is also a good strategy.

The hypercube should be used sparingly, since you have no idea where a relocated piece will reappear. Most players use the hypercube as a last resort—if another piece is going to be destroyed anyway, it doesn't hurt to take a chance and relocate it with the hypercube. Also, if your opponent's king is encircled with mirrors, you can march right in with your hypercube, followed by a block. This tactic may displace your opponent's defense, forcing him to evacuate the king from its mirrored fortress. Escorting the hypercube with an adjacent block prevents the opponent from attacking the hypercube with his or her king. Your opponent's only options will be to flee or be displaced.

Amiga Version

The Amiga version of *Laser Chess* (Program 1) requires 512K of mem-

When you want to talk computers..

ATARI COMPUTERS

Atari Computers

800 XL 64K Computer.....$63.99
65XE 64K Computer.....94.99
130XE 132K Computer.....129.00
520ST Monochrome System.....499.00
520ST Color System.....749.00

Atari 1040 Color System **$879**

Includes: 1040ST, 1 mb RAM with 3½" drive built-in, 192K ROM with TOS, Basic, Logo, ST language, power supply and color monitor.

ATARI SOFTWARE

Access
Leaderboard Golf.....26.99
Accolade
Fight Night.....19.99
Activision
Music Studio.....37.99
Antic
Cad 3-D.....34.99
Batteries Included
Paperclip w/Spellpack.....39.99
Degas Elite.....48.99
Epyx
World Karate Championship.....24.99
Infocom
Zork III.....29.99
Microprose
Top Gunner.....19.99
F-15 Strike Eagle.....24.99
Origin Systems
Ultima 4.....39.99
Paradox
Wanderer (3-D).....29.99
Psygnosis
Deep Space.....34.99
Timeworks
Wordwriter ST.....51.99
VIP
Professional (GEM).....144.00

COMMODORE COMPUTERS

Commodore 128 System

Includes: CB128 Computer, CB1571 Disk Drive, CB1902 Monitor **$759**

Commodore-64C 64K Computer189.00
Commodore-64C Sys. w/1802C.539.00
Commodore-128 128K Computer259.00
Amiga 1000 Computer.....899.00
Amiga 500.....Call for pricing
Amiga 2000.....and configuration

COMMODORE SOFTWARE

Activision
Hacker.....32.99
Broderbund
The Print Shop.....29.99
The Toy Shop.....39.99
Commodore
Textcraft w/Graphic Craft.....59.99
Assembler.....79.99
Enhancer DOS 1.2.....14.99
Discovery Software
Marauder Back-up.....32.99
Electronic Arts
Deluxe Paint.....69.99
Deluxe Print.....74.99
Instant Music.....34.99
Deluxe Video.....69.99
Infocom
Hitchhiker's Guide.....31.99
Micro Illusions
Dynamic-Cad.....349.00
Mindscape
Halley Project.....31.99
Deja Vu.....34.99
Micro Systems
Analyze Version 2.0.....119.00
Scribble.....64.99
On-Line/Comm.....49.99
Sublogic
Flight Simulator.....37.99
V.I.P.
V.I.P. Professional.....139.00

MS/DOS SYSTEMS

AT&T 6300.....from $1299.00
Compaq.....from 1699.0
IBM-XT.....from 1169.00
IBM-AT.....from 2599.00
Leading Edge.....from 999.00
NEC Multispeed.....from 1499.00
Panasonic Business Partnerfrom 799.00
Toshiba 1100 Plus.....from 1699.00

PC-TOO 20 Meg XT-Compatible **$999**

MULTIFUNCTION CARDS

AST
Six Pak Plus PC/XT.....$169.00
Hercules
Color Card.....159.00
Graphics Card Plus.....209.00
Fifth Generation
Logical Connection 256K.....329.00
Quadram
Expanded Quadboard.....119.00
Video 7
EGA Video Deluxe.....389.00
Zuckerboard
Color Card w/Parallel.....89.99

MS/DOS SOFTWARE

Ashton-Tate
d-Base III+.....409.00
5th Generation
Fastback Utility.....89.99
IMSI
Optimouse w/Dr. Halo.....99.99
Lotus
Lotus 1-2-3.....329.00
MicroPro
Professional 4.0 w/GL Demo.....239.00
Microstuf
Crosstalk XVI.....89.99
P.F.S.
First Choice.....119.00
Word Perfect Corp.
Word Perfect 4.2.....209.00

COMPUTER MAIL ORDER

.....When you want to talk price.

MODEMS

Anchor
6480 C64/128 1200 Baud.....$119.00
VM520 ST520/1040 1200 Baud.129.00
Atari
SX212 300/1200 (ST).....99.99
Commodore
Amiga 1680-1200 BPS.....169.00
CBM 1670 &C-128).....99.99
Everex
Evercom 1200 Baud Internal.....109.00
Hayes
Smartmodem 300 External.....139.00
Smartmodem 1200 External.....389.00
Practical Peripherals
1200 BPS External.....159.00
Quadram
Quadmodem II 1200 Baud.....299.00
Supra
MPP-1064 AD/AA C64.....69.99
1200AT 1200 Baud Atari.....139.00
U.S. Robotics
1200 Baud External.....129.00

MONITORS

Amdek 410 12" TTL Monitor $149

Amdek
Video 310A Amber TTL.....$139.00
Commodore
Commodore 1902.....299.00
Amiga 1080 Hi-Res Color.....269.00
Magnavox
8505 RGB/Composite.....199.00
515 RGB/Composite.....289.00
NEC
12" TTL Green or Amber.....109.00
JC-1401P3A Multi-Sync.....579.00
Princeton Graphics
MAX-12 12" Amber TTL.....169.00
Taxan
Model 124 12" Amber.....119.00
Zenith
ZVM 1220/1230.....(ea.) 99.99

DRIVES

Atari 1050 SS/DD (XL/XE) $139

Atari
AA354 SS/DD Disk (ST).....$129.00
SHD204 20 Meg Hard Drive (ST)599.00
Allied Technology
Apple Half-Heights.....109.00
Commodore
Amiga 1010 3½".....219.00
Amiga 1020 5¼".....189.00
1541C.....179.00
1571.....239.00
CSI
10 mb (64-128).....1099.00
Indus
GT Disk Drive.....179.00
Microbotics
20 mb Hard Drive (Amiga).....1299.00
Racore
PC Jr. Expansion Chassis.....299.00
Seagate
20 MB ST-225 Hard Drive Kit....379.00
Supra
20 Meg Hard Drive (XL/XE).....679.00
20 Meg Hard Drive (ST).....569.00
Xebec
20 mb (Amiga).....899.00

DISKETTES

Maxell
MD1-M SS/DD 5¼".....$9.99
MD2-DM DS/DD 5¼".....10.99
MF1-DDM SS/DD 3½".....12.99
MF2-DDM DS/DD 3½".....21.99
Sony
MD1D SS/DD 5¼".....8.99
MD2D DS/DD 5¼".....10.99
MFD-100 SS/DD 3½".....13.99
MFD-200 DS/DD 3½".....20.99
Hewlett-Packard Calculators
28C Scientific Pro.....199.99
18C Business Consultant.....139.95
12C Slim Financial.....74.99

PRINTERS

Atari
1020 XL/XE Plotter.....$29.99
1025 XL/XE Dot Matrix.....119.00
XDM121 Letter Quality.....209.00
XMM801 XL/XE Dot Matrix.....199.00
XMM804ST Dot Matrix.....189.00
Citizen
MSP-10 160 cps, 80-Column.....299.00
Premier 35 cps Daisywheel.....499.00
C.Itoh
8510-SP 180 cps, 80-Column.....Call
310-SEP Epson/IBM 80-Column.....Call

Epson LX-86 120 cps Dot Matrix $199

Epson
FX-86E 240 cps, 80-column.....Call
FX-286E 240 cps, 132-column.....Call
EX-800 300 cps, 80-column.....449.00
LQ-800 180 cps, 24-Wire PrintheadCall
Hewlett Packard
Thinkjet.....399.00
Juki
6300 40 cps Daisywheel.....659.00
6100 10 cps Daisywheel.....389.00
5510C Color Dot Matrix.....349.00
NEC
Pinwriter 660 24 Wire.....489.00
Pinwriter 760 24 Wire.....689.00
Okidata
Okimate 20 Color Printer.....129.00
ML-182 120 cps, 80-column.....239.00
ML-192+ 200 cps, 80-column...369.00
Panasonic
KX-1080i 120 cps, 80-column....219.00
KX-1091i 180 cps, 80-column....299.00
KX-1592 180 cps, 132-column...439.00
Star Micronics
NX-10 120 cps, 80-column.....209.00
NX-10C 120 cps, C64 Interface.219.00
NX-15 120 cps, 132-column.....369.00
Toshiba
P321 216cps, 24-Pin Printhead..479.00
P341 216cps, 24-Pin Printhead..589.00

In the U.S.A. and in Canada

Call toll-free: 1-800-233-8950.

Outside the U.S.A. call 717-327-9575 Telex 5106017898

Educational, Governmental and Corporate Organizations call toll-free 1-800-221-4283

CMO. 477 East Third Street, Dept. A206, Williamsport, PA 17701

ALL MAJOR CREDIT CARDS ACCEPTED.

POLICY: Add 3% (minimum $7.00) shipping and handling. Larger shipments may require additional charges. Personal and company checks require 3 weeks to clear. For faster delivery use your credit card or send cashier's check or bank money order. Pennsylvania residents add 6% sales tax. All prices are U.S.A. prices and are subject to change and all items are subject to availability. Defective software will be replaced with the same item only. Hardware will be replaced or repaired at our discretion within the terms and limits of the manufacturer's warranty. We cannot guarantee compatibility. All sales are final and returned shipments are subject to a restocking fee.

Written in Microsoft Amiga BASIC, the Amiga version of Laser Chess *duplicates almost exactly the original version of the game, which was written in the Modula-2 language for the Atari ST.*

The Commodore 64 version of Laser Chess *uses high-resolution graphics and sprites to good advantage.*

Laser Chess *for eight-bit Atari computers uses a graphics mode available only the Atari XE and XL models.*

The Apple II version of Laser Chess *employs keyboard controls and runs on any Apple II–series computer.*

ory and Microsoft Amiga BASIC. At the beginning of the game, you can choose between filled and unfilled playing pieces by pressing F or U, respectively. This option affects only the appearance of the pieces.

Amiga *Laser Chess* is played with the mouse, just like the original version for the Atari ST. To move a piece, position the mouse pointer over the desired piece and hold down the left mouse button. When the ghosted image of that piece appears, you can either drag the piece to a new location or rotate it by pressing a key. Release the mouse button to drop the piece in its new location.

The color of the playing-field border indicates the number of turns remaining and whose turn it is. There are three buttons to the left of the board. To select a button, move the mouse pointer over the button and press the left mouse button. The button labeled L fires the laser.

Commodore 64 Version

This version of *Laser Chess* (Program 2) requires at least one joystick. Since the program is written in machine language, it must be typed in with the "MLX" machine language entry program printed elsewhere in this issue. Here are the starting and ending addresses for MLX:

Starting address: 0801
Ending address: 1BB8

A cursor indicates your position on the board. Use the joystick to move the cursor over the piece you wish to move. (If you have only one joystick, plug it into port 2. You can simulate the second joystick by pressing the left-arrow, 1, 2, and CTRL keys to move the joystick left, right, up, and down, respectively, and pressing the space bar as a substitute for the button.)

To select a piece, hold down the fire button. To rotate a piece, move the joystick and press the button at the same time. To move a piece, move the pointer to the destination square after you have selected a piece; then press the button a second time.

Atari Version

The Atari version of *Laser Chess* (Program 3) works only on Atari XL and XE models, since it uses a graphics mode available only on those computers. The game begins by asking whether you are using one or two joysticks.

To move a piece, move the cursor onto the piece, press the fire button, and then move to the destination square and press the button a second time. The cursor turns a darker color when a piece has been selected. To rotate a piece, move the cursor over the piece and press the button until the piece has rotated to the desired position. If you decide not to rotate the piece, keep pressing until the piece is not selected any more. This version of *Laser Chess* does not allow you to combine a rotation and a move in one action. To fire the laser, move the cursor to the laser and press the button; then press L. To quit the game, press System Reset; to quit the game and start a new game, press System Reset and enter **RUN**.

Apple II Version

The Apple version of *Laser Chess* runs on any Apple II computer, under either DOS 3.3 or ProDOS. This program is written in two parts. The first part (Program 4) is written in BASIC. The second part (Program 5) is written in machine language and must be entered with the "MLX" program published elsewhere in this issue. Note that *you must save Program 5 with the filename LASER.ML because Program 4 attempts to load the file with that filename.* Enter the Program 5 addresses as indicated here:

STARTING ADDRESS? 6000
ENDING ADDRESS? 6577

Make sure that both Program 4 and Program 5 are present on the same disk before you start the game. Run Program 4 to begin.

This version of *Laser Chess* relies on keyboard controls. Use the arrow keys to move the cursor. If you are using an Apple II+, use CTRL-J to move the cursor down and CTRL-K to move it up. Press RETURN to select and place pieces. Once a piece has been selected, you can rotate it by pressing the < or > keys. The menu in the upper left portion of the screen is used to fire the laser or exit the game. Press the ESC key to enter the menu; then move to the desired menu selection with cursor controls. Press RE-

F L I G H T !

From the sophisticated realism, detail, and intellectual stimulation of Flight Simulator...

...to the brute-force fun, thrills and excitement of Jet...

...with new adventures in Scenery Disks...

...SubLOGIC. The State of the Art in Flight.

See Your Dealer. For additional product ordering information or the name of the dealer nearest you, call (800) 637- 4983.

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995
ORDER LINE: (800) 637-4983
(except in Illinois, Alaska and Hawaii)
Open 7 AM to 9 PM Central Time

Lyco Computer

Marketing & Consultants

Since 1981

Complete
COMMODORE 128 System

(reg. $1009.85)

$715

- Commodore 1571 Disk Drive
- Commodore 128 Computer
- Thompson 4120 Color Monitor w/Cable

(mouse optional)

Complete **COMMODORE 64 System**

- Commodore 64 C Computer
- 1541 C Drive
- 2 Joysticks

PLUS
- Seikosha SP-1000VC Printer

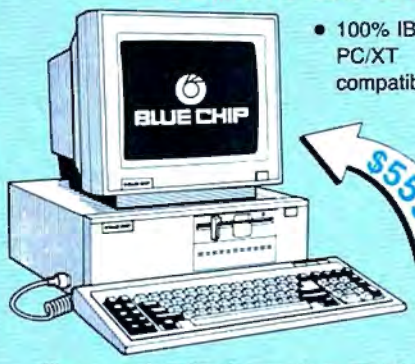
$509

(reg. $789)

BLUE CHIP PERSONAL COMPUTER

- 100% IBM PC/XT compatible!

$559



(Green, amber & color monitors available)

COMMODORE HARDWARE

128 Computer	$249	C-1700 128K RAM	$109.95
1571 Disk Drive	$229	1750 RAM	$169.95
64 C Computer	$175	Indus GT C-64 Drive	$185
1541 C Disk Drive	$185	GEOS	$CALL
1902 Monitor	$285	C-1351 Mouse	$39
1802C Monitor	$189	1670 Modem	$99

ATARI HARDWARE

SF 315 Disk Drive	$219.95
SF 354 Disk Drive	$175.95
1050 Drive (XE, XL)	$139.95
SHD 204 20 MEG Drive	$589
65XE	$99
520 ST Mono	$515
1040 Color	$879
Indus GT Atari Drive	$179
130XE Computer	$119
SX 212 Modem	$89.95

ATARI 520 ST Color System

- SC 1224 Color Monitor
- SF 354 Disk Drive
- 520 Keyboard

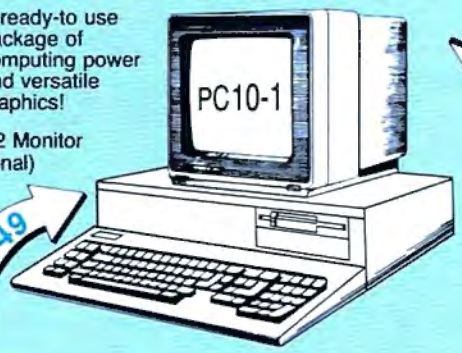
ATARI 1040 Monochrome System

COMMODORE PC10-1

- A ready-to use package of computing power and versatile graphics!

(1902 Monitor optional)

$749



$699

Lyco offers 90-day warranty on all Atari products at no additional charge.

$715

Star NP-10 Printer...**$164.95** when purchased with Atari 1040 System!

1-800-233-8760

Lyco Computer is one of the oldest and most established computer suppliers in America. Because we are dedicated to satisfying every customer, we have earned our reputation as the best in the business. And, our six years of experience in mail-order computer sales is your assurance of knowledgeable service and quality merchandise.

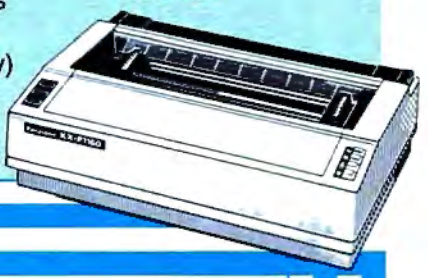
The Reliable, Affordable Choice

Panasonic 1080i

- 120 cps Draft Mode
- 24 cps NLQ
- Word Process
- Friction Feed

(2-year warranty)

$195



star micronics·inc

NP-10	$169
NX-10C	$209
NL-10	$209
NX-15	$329
NB-15	$889
SD-10	$259
ND-15	$425
SR-10	$469
NR-15	$529
NB24-15	$729

Panasonic

1091i	$269
1092i	$335
1592	$399

SEIKOSHA

SP-1200Ai	$195
SP-1000AP IIc	$179
SP-1200AS RS232	$195
SL-80Ai	$375
MP 1300Ai	$369
MP 5300Ai	$549
BP 5420Ai	$1075
1300 Color Kit	$119
BP5420Ribbon	$12.50
SP1000Ribbon	$8.50
1595	$499
3131	$249
3151	$379
1080 AP IIc	$239

JUKI

5510 w/color	$435
RS232 serial board	$55
Juki 6300	$739
Juki 6100	$369

LEGEND

808	$159
1080	$199
1380	$229
1385	$289

SILVER REED

EXP 420P	$209
EXP 600P	$539
EXP 800P	$649

OKIDATA

Okimate 20	$129
120 NLQ	$209
292 w/interface	$539
293 w/interface	$679
182	$245
192+	$355
193+	$539

DIABLO

D25	$499
635	$779
D-80 IF	$1029

Toshiba

321 P/S	$479
P341E	$699
P351 Model II	$1099

CITIZEN

120 D	$179
Premiere 35	$469
MSP-10	$285
MSP-15	$385
MSP-20	$325
MSP-25	$485
Tribute 224	$649

EPSON

LX 86	$209
EX 800	$355
EX 1000	$579
LQ 800	$449
LQ 1000	$659
FX 86E	$369
FX 286E	$519

Monitors

Thompson:	
4120 RGB/COM	$249
4160	$279
Teknika:	
MJ-305	$309
MJ-503	$529
Zenith:	
ZVM 1220	$89
ZVM 1230	$89
Panasonic:	
TR-122 MYP 12" Am TTL	$139
TR-122 M9P 12" Gr TTL	$139
Commodore:	
1902 Color	$285
1802 C	$189
NEC:	
Multisync	$CALL

Modems

Avatex:	
1200	$89
1200hc	$119
1200i	$99
2400	$289
Atari:	
XM-301	$42.95
SX 212	$89.95
Hayes:	
Smartmodem 300	$125
Smartmodem 1200	$369
Smartmodem 1200B	$339
Smartmodem 2400	$559
Micromodem IIe	$125
Smart 300 Apple IIc	$149
Commodore:	
1670	$99
US Robotics:	
Password 1200	$189
Microlink 1200	$139
Microlink 2400	$219
Courier HST 9600	$879
Compuserve	$17.95

Connect multiple printers with QVS Switch Boxes... starting at $39.95!

Purchase orders accepted from educational institutions. Also, ask about volume discounts!

Versatile and IBM Compatible

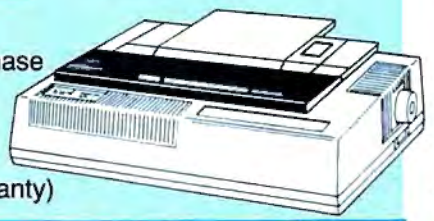
star micronics·inc

NX-10

$175

* with purchase of printer stand limited offer

(1-year warranty)

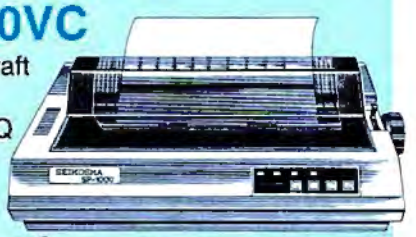


C-64/C-128 Compatible

SEIKOSHA (2-year warranty)

SP-1000VC

- 100 cps Draft Mode
- 20 cps NLQ
- Direct Connect C-64/128



(retail $299 — save $145!)

$154

We stock interfacing for Atari, Commodore, Apple and IBM.

What Makes Lyco Different?

Vickie Blaker, Customer Service Dept. Manager

Working in customer service gives me and my staff the opportunity to talk with people from all over the world who do their computer shopping by mail-order. Our loyal customers tell us that they keep coming back to Lyco because we are dedicated to customer satisfaction.

Call Lyco

Here Are Some Examples.

Giant inventory and low prices:

We hate to disappoint our customers. So, we keep a multimillion dollar inventory of all the factory-fresh merchandise you want. This means we can give you the lowest prices and the fastest delivery. And, it's also why we fill over 95% of all our customers' orders every month!

Prompt, courteous service:

When you call Lyco to place an order, you'll be in touch with some of the friendliest computer professionals in the industry. Everyone on our sales staff is very knowledgeable about the products they sell. They know that you want courteous and fast service, and that's exactly what you'll get when you call Lyco. (And for your convenience, we even have Saturday hours!)

Order Now

Many companies seem to forget about customers once a sale is made. Not Lyco. Our Customer Service Department is always here to help you — from questions about the status of an order, to warranties, to product availability and price. You'll **always** find friendly service at Lyco — before, during, **and after** your purchase.

Fast and easy delivery:

We know that when you place an order, you'd like to receive your shipment as soon as possible. At Lyco, we don't just say it — we do it. For instance, orders are normally shipped within 24 hours. Shipping on prepaid cash orders is free in the U.S., and there is **no deposit** required on C.O.D. orders. Air freight or UPS Blue/Red Label shipping is available, too.

Here's How

TO ORDER, CALL TOLL-FREE: 1-800-233-8760
In PA: 1-717-494-1030

Hours: 9AM to 8PM, Mon.-Thurs.
9AM to 6PM, Friday – 10AM to 6PM, Saturday

Or, send orders to: Lyco Computer, Inc., P.O. Box 5088
Jersey Shore, PA 17740
For Customer Service, call 1-717-494-1670,
9AM-5PM, Mon.-Fri.

Risk-Free Policy: • full manufacturers' warranties • no sales tax outside PA • prices show 4% cash discount: add 4% for credit cards • APO, FPO, international: add $5 plus 3% for priority • 4-week clearance on personal checks • we check for credit card theft • compatability not guaranteed • return authorization required • price/availability subject to change: call to confirm

MasterCard VISA

ATARI

Access:
Leader Board $24.95
Tournament #1 ..... $14.95
Triple Pack $14.95
10th Frame $24.95

Activision:
Hacker $15.95
Hitch Hikers $22.95
Leather Goddesses $22.95
Moonmist $22.95
Music Studio $22.95

Microprose:
Conflict in Vietnam $24.95
Decision in Desert $24.95
F-15 Strike Eagle $22.95
Kennedy Approach $19.95
Silent Service $22.95
Top Gunner $19.95

Microleague:
Microleag. Baseball $24.95
General Manager $24.95
Stat Disk $17.95
'86 Team Disk $14.95

Broderbund:
Print Shop $25.75
Print Shop Compan. $22.75
Graphic Lib. I, II, III $15.75
Karateka $18.75
Bank St. Writer $29.75

Optimized Systems:
Action $46.95
Action Tool Kit $18.95
Basic XE $46.95
Basic XL $36.95
Basic XL Tool Kit $18.95
Mac 65 ..... $46.95
Mac 65 Tool Kit $18.95

Strategic Simulations:
Battle of Antetiem $32.95
Battlecruiser $35.95
Nam $24.95
Phantasie $24.95
Wargame Construc. $18.95
Warship $39.95
Wizards Crown $24.95

Sublogic:
Flight Simulator II $31.95
Night Mission Pinball ... $21.95
Scenery Japan $15.95
Scenery #1 – #6 ..... $69.95
Scenery Disks, each ... $14.95

Epyx:
Eidolon $22.95
Karate Champ $18.95
Koronis Rift $22.95
Summer Games $24.95

Firebird:
Golden Path $22.95
The Pawn $27.95

Datasoft:
Alt. Real.: The Dungeon..$24.95
Gunslinger $19.95
221 B Baker St. $19.95

ATARI ST

Access:
Leader Board $24.95
Tournament #1 ..... $13.95
10th Frame $24.95

Activision:
Ballyhoo $23.95
Borrowed Time $32.95
Champion. Baseball ... $29.95
Champion. Basketball .. $28.95
Championship Golf ..... $New

GFL Football ...
Game Maker ...
Hacker ...
Hacker 2 ...
Leather Goddess
Little People ...
Moonmist ...
Music Studio ...
Paint Works ...
Portal ...
Shanghai ...
Tass Times ...
Bureaucracy ...

Data Soft:
Mercenary ...
Mercenary – 2nd
Alt. Reality – The
Alt. Rlty.-T. Dung

Microprose:
Silent Service ...
F-15 Strike Eagl

Optimized Syst
Personal Pascal
Personal Prolog

Strategic Simul
Phantasie ...
Phantasie II ...
Road War 2000
Colonial Conque

Epyx:
Apshai Trilogy ...
Sub Battle Simul
Super Cycle ...
World Games ...
Wrestling ...
Karate Champ ...

Firebird:
Pawn ...
Starglider ...
Golden Path ...
Guild of Thieves
Tracker ...

VIP:
Accounts ...
VIP Professional
VIP Professional

Unison World:
Art Gallery 1 or 2
Print Master ...

Zoom:
Zoomracks ...
Zoomracks II ...

Microleague:
Microleague Bas

COMM

Broderbund:
Bank St. Writer
Carmen San Die
Champ. Lode Ru
Graphic Lib. I, II
Karateka ...
Print Shop ...
Print Shop Comp
Print Shop Pape
Science Kit ...

Access:
Leader Board ...
Mach 5 ...
Mach – 128 ...
10th Frame ...
Exec. Tourname
Tournament #1
Triple Pack ...

SOFTWARE

ARI ST

..... $24.95
..... $New
..... $28.95
..... $32.95
s $23.95
..... $32.95
..... $23.95
..... $37.95
..... $43.95
..... $34.95
..... $28.95
..... $32.95
..... $24.95

..... $26.75
City .. $15.95
City . $34.95
eon ... $26.75

..... $24.95
..... $24.95

ms:
..... $48.95
..... $51.75

tions:
..... $24.95
..... $24.95
..... $26.95
t $24.95

..... $24.95
tor ... $24.95
..... $24.95
..... $24.95
..... $24.95
..... $24.95

..... $26.95
..... $26.95
..... $28.95
..... $27.95
..... $28.95

..... $174.95
..... $124.95
Lite ... $74.95

..... $18.95
..... $24.95

..... $54.95
..... $79.95

ball .. $35.95

DORE

28K .. $32.75
o $22.75
ner .. $24.75
ill $15.75
..... $18.75
..... $25.75
an. $22.75
..... $12.75
..... $35.75

..... $24.95
..... $23.95
..... $29.95
..... $24.95
t $13.95
..... $13.95
..... $14.95

COMMODORE

Wld. Cl. Leader Brd. ... $24.95

Activision:
Aliens $24.75
Champion. Basketball . $22.75
Hacker $18.75
Hacker 2 ..... $20.75
Labyrinth $22.75
Music Studio $22.75
Portal $26.95
Tass Times $22.75
Titanic $19.95

Microprose:
F-15 Strike Eagle $22.95
Gunship $22.95
Kennedy Approach $19.95
Silent Service $22.95
Solo Flight $19.95
Top Gunner $19.95

Microleague:
Microleag. Baseball $24.95
General Manager $24.95
Stat Disk $17.95
'86 Team Disk $14.95
Microleag. Wrestling ..... $New

Strategic Simulations:
Gemstone Healer $18.95
Gettysburg $36.95
Kampfgruppe $36.95
Phantasie II $24.95
Phantasie III $26.95
Ring of Zelfin $24.95
Road War 2000 ..... $24.95
Shard of Spring $24.95
Wizards Crown $24.95
War in the S. Pacific ... $39.95
Wargame Constr. $21.95
Battlecruiser $36.95
Gemstone Warrior $14.95

Sublogic:
Baseball Stadium $15.95
Flight Simulator II $31.95
Jet Simulator $31.95
Night Mission Pinball ... $21.95
Scenery Disk $15.95
Sublogic Baseball $31.95
Sublogic Football $26.95

Epyx:
Create A Calendar $17.95
Destroyer $24.75
Fastload $24.75
Football $24.75
Karate Champ $19.75
Movie Monster $24.75
Multiplan $24.75
Sub Battle $23.95
Winter Games $23.95
Super Cycle $23.95
Graphics Scrapbook $14.95
Str. Sports Basketball . $24.95
Wld.'s Great Baseball . $19.95
Summer Games II $24.75
Vorpol Utility Kit $15.75
World Games $24.75

Firebird:
Colossus IV $21.95
Elite $19.95
Frankie Goes to Italy ... $19.95
Golden Path $34.95
Guild of Thieves $24.95
Pawn $28.95
Sentinal $24.95
Talking Teacher $28.95
Tracker $31.95
Starglider $24.95
Sentry $24.95

HI Tech Expressions:
Heart Ware $6.75
Holiday Paper $8.95
Card Ware $6.75
Party Ware $9.75

Unison World:
Art Gallery 1 or 2 ..... $15.95
Print Master $22.95

Batteries Included:
Paper Clip $26.95
Paper Clip II $44.95
Consultant 64 ..... $24.95
Consultant 128 ..... $34.95
CalKit $14.95

Datasoft:
Alt. Real.: The Dungeon..$24.95
Black Magic $15.95
Saracen $15.95

Action Soft:
Up Periscope $19.95

apple

Access:
Triple Pack $14.95

Activision:
Enchanter Trilogy $49.75
Game Maker $31.75
Leather Goddesses $25.75
Term Paper $35.75
Champ. Baseball $24.95
Champ. Basketball $26.95
Championship Golf ..... $New
GFL Football $24.95
Portal $27.95

Microprose:
Crusade in Europe $24.95
Decision in Desert $24.95
F-15 Strike Eagle $22.95
Silent Service $22.95

Microleague:
Microleag. Baseball $25.95
General Manager $25.95
Stat Disk $17.95
'86 Team Disk $14.95

Broderbund:
Airheart $22.75
Ancient Art of War $25.75
Print Shop $29.75
Print Shop Comp. $25.75
Carmen S. Diego (USA). $28.75
On Balance $62.75
Bank St. Writer + ... $44.75

Strategic Simulations:
Battlecruiser $35.95
Battlegruppe $38.95
Colonial Conquest $25.95
Gettysburg $35.95
Phantasie II $25.95
Phantasie III $24.95
Realms of Darkness $24.95
Shard of Spring $25.95
Wizards Crown $25.95

Sublogic:
Flight Simulator II $32.95
Jet Simulator $26.95
Night Mission Pinball ... $22.95
Scenery #1-#6 ..... $69.95
Sublogic Baseball $31.95

Epyx:
Create A Calendar $17.95
Destroyer $24.95
Karate Champ $19.95
Movie Monster $24.95
St. Sports Basketball ... $24.95
Sub Battle Simulator ... $24.95
Winter Games $24.95
World Games $24.95
Wrestling $24.95

Firebird:
Colossus IV Chess $25.95
Elite $22.95
The Pawn $26.95
Starglider $28.95

Unison World:
Art Gallery 2 ..... $19.95
Print Master $25.95

Datasoft:
Saracen $15.95

AMIGA

Access:
Leader Board $24.95
10th Frame $24.95
Tournament #1 ..... $14.95

Activision:
Borrowed Time $26.95
Champ. Basketball $26.95
Championship Golf $32.95
GFL Football $27.95
Hacker $26.95
Hacker 2 ..... $29.95
Little People $29.95
Mindshadow $26.95
Music Studio $35.95
Tass Times $26.95

Strategic Simulations:
Computer Baseball $24.95
Kampfgruppe $34.95

Sublogic:
Flight Simulator II $32.95
Jet Simulator ..... $CALL

Epyx:
Apshai Trilogy $22.95
Rogue $22.95
Winter Games $22.95

Firebird:
Guild of Thieves $27.95
Pawn $26.95
Starglider $26.95

Unison World:
Print Master $24.95
Art Gallery 1 or 2 ..... $18.95

VIP:
VIP Professional $115

Microprose:
Silent Service $24.95

IBM

Activision:
Pebble Beach $32.95
Portal $28.95
Champ. Baseball $24.95
Champ. Basketball $26.95
Ballyhoo $25.75
Enchanter Trilogy $49.75
GFL Football $24.95
Leather Goddessses ... $25.75
Moonmist $25.75
Music Studio $32.75

Microprose:
Conflict in Vietnam $24.95
Crusade in Europe $24.95
Decision in Desert $24.95
F-15 Strike Eagle $22.95
Silent Service $22.95

Microleague:
Microleag. Baseball $25.95
General Manager $25.95
Stat Disk $17.95
'86 Team Disk $14.95

Epyx:
Apshai Trilogy $24.75
Create A Calendar $17.95
Destroyer $24.75
Karate Champ $19.75
Movie Monster $24.75
Rogue $19.75
St. Sports Basketball ... $24.95
Sub Battle Simulator ... $24.95
Winter Games $24.75
World Games $24.75

Strategic Simulations:
Road War 2000 ..... $24.95
Kampfgruppe $36.95
Shard of Spring $24.95
Battle of Antetiem $38.75
Computer Baseball $14.95
Gettysburg $38.75
Knights in Desert $25.75

Sublogic:
Jet Simulator $32.95
Scenery Japan $15.95
Scenery San Fran. $15.95
Scenery #1-#6 ..... $69.95
Sublogic Baseball $31.95
Sublogic Football $31.95

Broderbund:
Ancient Art of War $28.95
Print Shop $37.75
Print Shop Compan. $31.75
Graphic Lib. I or II $21.75
Karateka $21.75
Toy Shop $39.95
Bank St. Writer + ... $54.95

Unison World:
Art Gallery 2 ..... $18.95
News Master $55.75
Print Master $36.75

Firebird:
Starglider $27.95
Guild of Thieves $27.95

Diskettes

5-1/4

Maxell:
SSDD $9.99
DSDD $12.99

Bonus:
SSDD $6.99
DSDD $7.50

SKC:
SSDD $8.50
DSDD $9.50
DSHD $16.95

Verbatim:
SSDD $9.99
DSDD $12.99

3.5

3M:
SSDD $16.99
DSDD $23.99

Maxell:
SSDD $15.99
DSDD $21.99

Verbatim:
SSDD $16.99
DSDD $24.99

SKC:
SSDD $14.99
DSDD $16.99

Lyco stocks joysticks by Wico, Suncom, Tac, Epyx, and more, starting as low as $5.95!

TURN to select an item from this menu.

For instructions on entering these programs, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

Program 1: Amiga *Laser Chess*

Version by Tim Midkiff, Editorial Programmer

```
'Laser Chess
'Copyright 1987 Compute! Publica
tions, Inc.
'All Rights Reserved
CLEAR,25000:CLEAR ,50000&
DEFINT a-z:DEFSNG r,g,b,mx:RANDO
MIZE TIMER:SCREEN 1,320,200,4,1
WINDOW 3,,(0,0)-(311,186),16,1:W
INDOW OUTPUT 3:COLOR ,0
DIM sn(8,3,1,1),es(155,1),shape(
155,87),piece(9,9),orient(9,9),c
Lr(9,9)
DIM os(155),beamck(3,9,9),dirck(
8,3,3),bmd0(158),bmd1(22),shpt(8
)
DIM ddrcx(1,20),ddrcy(1,20),pt(1
4)
DIM s(255),n(255),sq(255),freq(2
0,4),shptx(8,19),shpty(8,19)
LOCATE 1,4:PRINT CHR$(169)"1987
Compute! Publications, Inc."
LOCATE 3,11:PRINT"All Rights Res
erved"
LOCATE 12,9:PRINT"(F)illed or (U
)nfilled?"
WHILE NOT(k$="F" OR k$="U"):k$=U
CASE$(INKEY$):WEND:fL=k$="F"
PALETTE 0,.15,.05,.5:PALETTE 1,.
15,.25,.95
FOR i=2 TO 14:PALETTE i,.15,.05,
.5:NEXT:PALETTE 15,.15,.25,.95
ON TIMER(1) GOSUB CLock:ti=36-fL
*10
COLOR 1,0:CLS:LOCATE 10,14:PRINT
"PLEASE  WAIT"
LOCATE 12,18:PRINT "seconds":TIM
ER ON
FOR i=0 TO 255:s(i)=127-i:NEXT:F
OR i=0 TO 255:n(i)=127-RND*255:N
EXT:WAVE 0,s
FOR i=0 TO 127:sq(i)=127-RND*50:
NEXT:FOR i=128 TO 255:sq(i)=-128
+RND*50:NEXT
cop(1)=4:cop(2)=6:GOSUB InitShap
es:GOSUB InitObjects:TIMER OFF:C
LS
RESTORE PaLetteData:FOR i=2 TO 1
4:READ r,g,b:PALETTE i,r,g,b:NEX
T
PaLetteData:
DATA 0,0,0,.3,.3,.3
DATA .6,0,0,1,0,0,0,.55,0,0,.9,0
DATA 1,1,0,1,1,0,.6,.6,.6,1,1,1
DATA 1,1,0,1,1,0,1,1,0
Start:
L(1)=1:L(2)=1:Lpx(1)=4:Lpy(1)=1:
Lpx(2)=6:Lpy(2)=9
COLOR ,0:GOSUB DrawBoard:k=0:pL=
1
Main:
pL=pL XOR 3:px=5:py=5:move=2:hyc
ube=0:hysq=0:taken=0:fired=1
LINE(40,10)-(288,186),cop(pL),b:
LINE(42,12)-(286,184),cop(pL),b
MovePiece:
WHILE MOUSE(0)>-1:WEND:x=MOUSE(3
):y=MOUSE(4)
px=INT((x-17)/27):py=INT((y+6)/1
9):moves=0
IF NOT((px>0 AND px<10) AND (py>
0 AND py<10)) THEN Options
IF cLr(px,py)<>pL THEN MovePiece
piece=piece(px,py):rot=orient(px
,py)
obindex=oi(piece,rot):spx=px:spy
=py
IF NOT(obindex>0) THEN MovePiece
OBJECT.X obindex,x-14:OBJECT.Y o
bindex,y-10:OBJECT.ON obindex
WHILE MOUSE(0)<0
OBJECT.X obindex,MOUSE(1)-14:OBJ
ECT.Y obindex,MOUSE(2)-10
IF INKEY$<>"" THEN
rot=(rot+1) AND turns(piece):j=o
bindex:obindex=oi(piece,rot)
OBJECT.X obindex,MOUSE(1)-14:OBJ
ECT.Y obindex,MOUSE(2)-10
OBJECT.OFF j:WAVE 0,s:SOUND 4000
,.1,255,0:OBJECT.ON obindex
END IF
WEND
OBJECT.OFF obindex
GOSUB EraseSquare
px=INT((MOUSE(5)-17)/27):py=INT(
(MOUSE(6)+6)/19)
GOSUB CheckMove
GOSUB PutShape
IF piece(px,py)=2 THEN Lpx(pL)=p
x:Lpy(pL)=py
EndMove:IF k THEN EndGame
move=move-moves:IF move=1 THEN L
INE(40,10)-(288,186),0,b
IF move>0 THEN MovePiece
GOTO Main

CLock:ti=ti-1:LOCATE 12,14:PRINT
STR$(ti)" ":RETURN

InitShapes:
LINE(0,0)-(0,0),10,bf:GET(0,0)-(
0,0),pt:PUT(0,0),pt
LINE(0,0)-(0,18),10:GET(0,0)-(0,
18),bmd0:PUT(0,0),bmd0
LINE(0,0)-(26,0),10:GET(0,0)-(26
,0),bmd1:PUT(0,0),bmd1
LINE(0,0)-(26,18),2,bf:GET(0,0)-
(26,18),es(0,0)
LINE(0,0)-(26,18),3,bf:GET(0,0)-
(26,18),es(0,1)
RESTORE LaserDir:FOR i=0 TO 3:RE
AD dirx(i),diry(i):NEXT
LaserDir:DATA 0,-1,1,0,0,1,-1,0
RESTORE ShapePts
k=0:x=0:y=0:FOR i=1 TO 8:READ tu
rns(i),shpt(i)
FOR j=0 TO shpt(i)+1:READ shptx(
i,j),shpty(i,j):NEXT
GOSUB GetShapes:NEXT
RESTORE ShapeRefLect
FOR i=1 TO 8:FOR j=0 TO turns(i)
:FOR k=0 TO 3:READ dirck(i,j,k):
NEXT k,j,i
RETURN
ShapePts:
DATA 1,1,-1,17,17,1,0,0
DATA 3,6,7,17,9,1,11,17,7,17,1,1
5,17,15,11,17,9,9
DATA 1,1,-1,9,17,9,0,0
DATA 0,7,5,9,9,5,13,9,9,13,5,9,1
3,9,9,13,9,5,0,0
DATA 3,6,1,2,17,2,17,17,1,17,1,2
,-1,1,17,1,9,9
DATA 0,4,1,1,17,1,17,17,1,17,1,1
,0,0
DATA 3,6,2,1,16,1,9,8,2,1,-1,1,-
9,9,17,1,9,4
DATA 3,5,2,17,17,17,17,2,2,17,-1
,17,17,1,13,13
ShapeRefLect:
DATA 1,0,3,2,3,2,1,0,-1,-1,-1,-1
,-1,-1,-1,-1
DATA -1,-1,-1,-1,-1,-1,-1,-1,2,1
,0,3,0,3,2,1
DATA -1,-1,-1,-1,-1,-1,0,-1,-1,-
1,-1,1,2,-1,-1,-1
DATA -1,3,-1,-1,0,1,2,3,-2,2,-1,
2,3,-2,3,-1
DATA -1,0,-2,0,1,-1,1,-2,-1,0,3,
-1,-1,-1,1,0
DATA 1,-1,-1,2,3,2,-1,-1
GetShapes:
FOR angLe=0 TO turns(i):FOR bkgd
=0 TO 1:FOR pL=1 TO 2
co=cop(pL):PUT(0,0),es(0,bkgd),P
SET
ON angLe+1 GOSUB rotate0,rotate9
0,rotate180,rotate270
sn(i,angLe,pL-1,bkgd)=k:GET(0,0)
-(26,18),shape(0,k):k=k+1
NEXT pL,bkgd,angLe:RETURN

rotate0:
FOR j=1 TO shpt(i):IF shptx(i,j-
1)<0 THEN hue=co+1 ELSE hue=co
LINE(ABS(shptx(i,j-1))+4+x,shpty
(i,j-1)+y)-(ABS(shptx(i,j))+4+x,
shpty(i,j)+y),hue:NEXT
IF shptx(i,shpt(i)+1)>0 AND fL T
HEN PAINT(shptx(i,shpt(i)+1)+4+x
,shpty(i,shpt(i)+1)+y),co,co
RETURN
rotate90:
FOR j=1 TO shpt(i):IF shptx(i,j-
1)<0 THEN hue=co+1 ELSE hue=co
LINE(18-shpty(i,j-1)+4+x,ABS(shp
tx(i,j-1))+y)-(18-shpty(i,j)+4+x
,ABS(shptx(i,j))+y),hue:NEXT
IF shptx(i,shpt(i)+1)>0 AND fL T
HEN PAINT(18-shpty(i,shpt(i)+1)+
4+x,shptx(i,shpt(i)+1)+y),co,co
RETURN
rotate180:
FOR j=1 TO shpt(i):IF shptx(i,j-
1)<0 THEN hue=co+1 ELSE hue=co
LINE(18-ABS(shptx(i,j-1))+4+x,18
-shpty(i,j-1)+y)-(18-ABS(shptx(i
,j))+4+x,18-shpty(i,j)+y),hue:NE
XT
IF shptx(i,shpt(i)+1)>0 AND fL T
HEN PAINT(18-shptx(i,shpt(i)+1)+
4+x,18-shpty(i,shpt(i)+1)+y),co,
co
RETURN
rotate270:
FOR j=1 TO shpt(i):IF shptx(i,j-
1)<0 THEN hue=co+1 ELSE hue=co
LINE(shpty(i,j-1)+4+x,18-ABS(shp
tx(i,j-1))+y)-(shpty(i,j)+4+x,18
-ABS(shptx(i,j))+y),hue:NEXT
IF shptx(i,shpt(i)+1)>0 AND fL T
HEN PAINT(shpty(i,shpt(i)+1)+4+x
,18-shptx(i,shpt(i)+1)+y),co,co
RETURN

InitObjects:
k=1:si$=STRING$(26,0):POKE SADD(
si$)+11,4:POKE SADD(si$)+15,27
POKE SADD(si$)+19,19:POKE SADD(s
i$)+21,24:POKE SADD(si$)+23,15
FOR piece=1 TO 8:FOR angLe=0 TO
turns(piece):seLect=sn(piece,ang
Le,0,0)
PUT(0,0),es(0,0),PSET:PUT(0,0),s
hape(0,seLect)
oi(piece,angLe)=k:GET(0,0)-(26,1
8),os
sd$="":FOR i=3 TO 154:sd$=sd$+MK
I$(os(i)):NEXT
OBJECT.SHAPE k,si$+sd$:OBJECT.PL
ANES k,3,8
k=k+1:NEXT angLe,piece:RETURN

DrawBoard:
COLOR 3,2:LINE(11,54)-STEP(16,11
),,b:PAINT(12,55),2,3:LOCATE 8,3
```

```
:PRINT"Q"
LINE(11,94)-STEP(16,10),,b:PAINT
(12,95),2,3:LOCATE 13,3:PRINT"R"
LINE(11,134)-STEP(16,10),,b:PAIN
T(12,135),2,3:LOCATE 18,3:PRINT"
L"
FOR py=1 TO 9:FOR px=1 TO 9:GOSU
B EraseSquare:NEXT px,py
LINE(151,89)-(177,107),0,bf
RESTORE ShapePos:FOR py=1 TO 2:F
OR px=1 TO 9:cLr(px,py)=1:cLr(px
,py+7)=2
READ piece(px,py),orient(px,py),
orient(10-px,10-py)
piece(10-px,10-py)=piece(px,py):
NEXT px,py
FOR px=1 TO 9:FOR py=1 TO 9:IF p
iece(px,py)>0 THEN GOSUB PutShap
e
NEXT py,px
ShapePos:
DATA 8,2,0,8,2,0,1,1,1,2,2,0,4,0
,0,6,0,0
DATA 1,0,0,8,3,1,8,3,1,8,3,1,5,2
,0,5,2,0
DATA 7,2,0,3,0,0,3,1,1,5,2,0,5,2
,0,8,2,0
RETURN

PutShape:
x=px*27+16:y=py*19-6:bkgd=(px+py
+1) AND 1
PUT(x,y),shape(0,sn(piece(px,py)
,orient(px,py),cLr(px,py)-1,bkgd
)),PSET
RETURN

EraseSquare:
x=px*27+16:y=py*19-6:bkgd=(px+py
+1) AND 1:PUT(x,y),es(0,bkgd),PS
ET
RETURN

Fire:
px=Lpx(pL):py=Lpy(pL):Lx(1)=px:L
y(1)=py:dir(1)=orient(px,py)
FOR i=1 TO 3:aLive(i)=0:term(i)=
0:NEXT:aLive(1)=1
WHILE (aLive(1)=1) OR (aLive(2)=
1) OR (aLive(3)=1)
FOR i=1 TO 3:IF aLive(i)<1 THEN
AdvBeam
nLx(i)=Lx(i)+dirx(dir(i)):nLy(i)
=Ly(i)+diry(dir(i))
IF beamck(dir(i),Lx(i),Ly(i))=1
THEN EndBeam
beamck(dir(i),Lx(i),Ly(i))=1:GOT
O DrawBeam
Hit:term(i)=1:drk(i)=tdir:IF d T
HEN EndBeam
tx=px:ty=py:px=Lx(i):py=Ly(i):IF
piece(px,py)=4 THEN k=k+cLr(px,p
y)
IF piece(px,py)=2 THEN L(cLr(px,
py))=0
x=px*27+16:y=py*19-6
m=piece(px,py):shpt(0)=shpt(m):F
OR j=0 TO shpt(0)+1:shptx(0,j)=s
hptx(m,j)
shpty(0,j)=shpty(m,j)::NEXT:t=i:
i=0:co=8
ON orient(px,py)+1 GOSUB rotate0
,rotate90,rotate180,rotate270
i=t:px=tx:py=ty
EndBeam:aLive(i)=-1
AdvBeam:NEXT:WEND
RETURN

DrawBeam:
x=Lx(i)*27+29:y=Ly(i)*19+3
ON dir(i) GOTO BRt,BDn,BLt
BUp:PUT(x,y-19),bmd0:GOTO CkBeam

BRt:PUT(x,y+1),bmd1:GOTO CkBeam
BDn:PUT(x+1,y),bmd0:GOTO CkBeam
BLt:PUT(x-27,y),bmd1:GOTO CkBeam


CkBeam:
IF (nLx(i)>9) OR (nLy(i)>9) OR (
nLx(i)<1) OR (nLy(i)<1) THEN End
Beam
IF nLx(i)=5 AND nLy(i)=5 THEN En
dBeam
Lx(i)=nLx(i):Ly(i)=nLy(i):IF pie
ce(nLx(i),nLy(i))=0 THEN AdvBeam

tdir=dir(i):dir(i)=dirck(piece(L
x(i),Ly(i)),orient(Lx(i),Ly(i)),
dir(i))
IF dir(i)=-1 THEN Hit
IF dir(i)>-2 THEN AdvBeam
IF aLive(2)=0 THEN j=2 ELSE j=3
aLive(j)=1:Lx(j)=Lx(i):Ly(j)=Ly(
i)
dir(i)=tdir+1 AND 3:dir(j)=tdir-
1 AND 3
GOTO AdvBeam

Laser:
k=0:d=0:GOSUB Fire
FOR i=0 TO 3:FOR x=1 TO 9:FOR y=
1 TO 9:beamck(i,x,y)=0:NEXT y,x,
i
FOR i=1 TO 3
IF term(i)=1 THEN
IF piece(Lx(i),Ly(i))>0 THEN
tx=px:ty=py:px=Lx(i):py=Ly(i):GO
SUB ExpLode:px=tx:py=ty
END IF
END IF
NEXT
TIMER OFF:d=1:GOSUB Fire
FOR i=1 TO 3
IF term(i)=1 THEN
IF piece(Lx(i),Ly(i))>0 THEN
tx=px:ty=py:px=Lx(i):py=Ly(i):pi
ece(px,py)=0:cLr(px,py)=0
GOSUB EraseSquare:px=tx:py=ty
END IF
END IF
NEXT
FOR i=0 TO 3:FOR x=1 TO 9:FOR y=
1 TO 9:beamck(i,x,y)=0:NEXT y,x,
i
RETURN

ExpLode:
FOR j=0 TO 4:vol(4-j)=(j+1)*40:N
EXT:ch=0
FOR j=0 TO 20:t=900-INT(RND*8)*1
00:FOR m=0 TO 4:freq(j,m)=t:NEXT
m,j
Lv=120:cx=px*27+29:cy=py*19+3:WA
VE 0,n:WAVE 1,n
IF dirx(drk(i))=0 THEN
FOR j=0 TO 20:ddrcy(0,j)=INT(RND
*10)*diry(drk(i))+cy
ddrcx(0,j)=cx+10-INT(RND*20)
ddrcy(1,j)=INT(RND*20)*diry(drk(
i))+cy
ddrcx(1,j)=cx+20-INT(RND*40):NEX
T
ELSE
FOR j=0 TO 20:ddrcx(0,j)=INT(RND
*10)*dirx(drk(i))+cx
ddrcy(0,j)=cy+10-INT(RND*20)
ddrcx(1,j)=INT(RND*20)*dirx(drk(
i))+cx
ddrcy(1,j)=cy+20-INT(RND*40):NEX
T
END IF
GOSUB EraseSquare
FOR j=0 TO 20:PUT(ddrcx(0,j),ddr
cy(0,j)),pt:IF (j AND 4)=4 THEN
GOSUB ExpSnd
NEXT
FOR j=0 TO 20:PUT(ddrcx(0,j),ddr
cy(0,j)),pt:PUT(ddrcx(1,j),ddrcy
(1,j)),pt
IF (j AND 4)=4 THEN GOSUB ExpSnd

NEXT:FOR j=0 TO 20:PUT(ddrcx(1,j
),ddrcy(1,j)),pt:NEXT
RETURN

ExpSnd:
ch=1-ch:FOR m=0 TO 4:SOUND freq(
j,m),.05,vol(m),ch:NEXT:RETURN


CheckMove:
dx=ABS(px-spx):dy=ABS(py-spy)
moves=dx+dy+ABS(rot<>orient(spx,
spy))
IF dx=0 AND dy=0 THEN VaLidMove
IF NOT(px>0 AND px<10 AND py>0 A
ND py<10) THEN InVaLidMove
IF moves>move THEN InVaLidMove
IF moves=2 THEN
midx=(px+spx)/2:midy=(py+spy)/2
IF midx=5 AND midy=5 THEN InVaLi
dMove
IF dx=2 THEN IF piece(midx,py)<>
0 THEN InVaLidMove
IF dy=2 THEN IF piece(px,midy)<>
0 THEN InVaLidMove
IF dx=1 AND dy=1 THEN
IF ((piece(px,spy)<>0) OR (px=5
AND spy=5)) AND ((piece(spx,py)<
>0) OR (spx=5 AND py=5)) THEN In
VaLidMove
END IF
END IF
IF piece(px,py)<>0 THEN
IF piece=4 OR piece=5 THEN
IF taken THEN InVaLidMove
IF piece(px,py)=4 THEN k=cLr(px,
py)
IF piece(px,py)=2 THEN L(cLr(px,
py))=0
WAVE 0,n:WAVE 1,n
FOR i=255 TO 10 STEP -20:SOUND 4
00,.1,i,0:SOUND 400,.1,i,1:NEXT
taken=1:GOTO VaLidMove
ELSEIF piece=6 THEN
IF hycube THEN InVaLidMove
hycube=1:GOTO HyperCube
ELSE
GOTO InVaLidMove
END IF
END IF
IF NOT(px=5 AND py=5) THEN VaLid
Move
IF hysq THEN InVaLidMove
WHILE (px=5 AND py=5) OR piece(p
x,py)<>0
px=INT(RND*9+1):py=INT(RND*9+1)
WEND
WAVE 0,n:FOR i=250 TO 0 STEP -2:
SOUND 100+i*2,.03,i,0:NEXT:WAVE
1,n
GOSUB VaLidMove:FOR i=0 TO 250 S
TEP 2:SOUND 100+500-i*2,.03,i,1:
NEXT
hysq=1:GOSUB PutShape
RETURN
HyperCube:
nx=INT(RND*9+1):ny=INT(RND*9+1)
IF (nx=5 AND ny=5) OR piece(nx,n
y)<>0 THEN HyperCube
WAVE 0,n:FOR i=250 TO 0 STEP -2:
SOUND 100+i*2,.03,i,0:NEXT:WAVE
1,n
piece(nx,ny)=piece(px,py):orient
(nx,ny)=orient(px,py):cLr(nx,ny)
=cLr(px,py)
GOSUB VaLidMove:FOR i=0 TO 250 S
TEP 2:SOUND 100+500-i*2,.03,i,1:
NEXT
GOSUB PutShape:piece(spx,spy)=0:
cLr(spx,spy)=0:px=nx:py=ny
RETURN
```

```
VaLidMove:
piece(px,py)=piece:orient(px,py)
=rot:cLr(px,py)=cLr(spx,spy)
IF dx>0 OR dy>0 THEN piece(spx,s
py)=0:cLr(spx,spy)=0
RETURN
InVaLidMove:
px=spx:py=spy:moves=0:RETURN
Confirm:
WINDOW 2,,(114,82)-(216,105),0,1
:WINDOW OUTPUT 2:PRINT"Are you s
ure?"
COLOR 3,2:LINE(27,14)-STEP(16,10
),,b:PAINT(28,15),2,3:LOCATE 3,5
:PRINT"Y"
LINE(59,14)-STEP(16,10),,b:PAINT
(68,15),2,3:LOCATE 3,9:PRINT"N"
CkCon:WHILE MOUSE(0)>-1:WEND:x=M
OUSE(3):y=MOUSE(4):co=POINT(x,y)
IF NOT(co=2 OR co=3) THEN CkCon
IF x>27 AND x<43 THEN
c=1
ELSEIF x>59 AND x<75 THEN
c=0
ELSE
GOTO CkCon
END IF
WINDOW CLOSE 2:WHILE MOUSE(0)<>0
:WEND:RETURN
Options:
co=POINT(x,y):moves=0
IF NOT(co=2 OR co=3) THEN MovePi
ece
IF y>133 AND y<145 AND fired AND
L(pL) THEN
fired=0:PALETTE 10,1,1,0:PALETTE
15,1,1,0
WAVE 2,sq:WAVE 3,sq:ON TIMER(1)
GOSUB LSnd:Lv=200:GOSUB LSnd:TIM
ER ON
GOSUB Laser:moves=1
PALETTE 10,.6,.6,.6:PALETTE 15,.
15,.25,.95
ELSEIF y>93 AND y<105 THEN
GOSUB Confirm:IF c THEN Restart
ELSEIF y>53 AND y<66 THEN
GOSUB Confirm:IF c THEN SCREEN C
LOSE 1:WINDOW CLOSE 3:CLEAR,2500
0:END
END IF
GOTO EndMove

LSnd:SOUND 120,18.2,Lv,2:SOUND 1
21,18.2,Lv,3:RETURN

Border:LINE(40,10)-(288,186),,b:
LINE(42,12)-(286,184),,b:RETURN

Restart:
COLOR ,0:CLS:FOR i=1 TO 9:FOR j=
1 TO 9:piece(i,j)=0:cLr(i,j)=0:N
EXT j,i
GOTO Start

EGOpt:
co=POINT(x,y)
IF co=2 OR co=3 THEN
IF y>93 AND y<105 THEN Restart
IF y>53 AND y<66 THEN SCREEN CLO
SE 1:WINDOW CLOSE 3:CLEAR,25000:
END
END IF
GOTO EndGWait

EndGame:
IF k=3 THEN
COLOR 10,0:GOSUB Border:COLOR 11
:LOCATE 1,19:PRINT"Draw"
ELSE
IF k=2 THEN COLOR 4,0:GOSUB Bord
er:COLOR 5:LOCATE 1,16:PRINT"Red
";
IF k=1 THEN COLOR 6,0:GOSUB Bord
er:COLOR 7:LOCATE 1,15:PRINT"Gre
en";
PRINT" victory"
END IF
EndGWait:
WHILE MOUSE(0)>-1:WEND:x=MOUSE(3
):y=MOUSE(4)
px=INT((x-8)/27):py=INT((y+6)/19
):moves=0
IF NOT((px>0 AND px<10) AND (py>
0 AND py<10)) THEN EGOpt
GOTO EndGWait
```

Program 2: Commodore 64
Laser Chess

Version by Bill Chin, Editorial Programmer

```
0801:0B 08 0A 00 9E 32 30 36 2E
0809:31 00 00 00 A9 00 8D 21 3B
0811:D0 8D 20 D0 A9 0F 8D 86 29
0819:02 20 C0 0A A9 78 A0 18 73
0821:20 1E AB A5 C6 F0 FC A9 36
0829:FF 8D 21 C8 20 7B 0B A9 FB
0831:09 8D 04 C8 A9 07 8D 21 DB
0839:D0 A9 00 8D 16 C8 A9 00 1C
0841:85 FD A9 C0 85 FE A9 CE 1F
0849:85 FB A9 18 85 FC A0 00 33
0851:B1 FB F0 04 C9 19 90 0C 77
0859:91 FD 20 FD 08 20 EC 08 38
0861:D0 EE F0 17 AA A9 FF 91 B2
0869:FD CA F0 07 20 EC 08 D0 4F
0871:F6 F0 08 20 FD 08 20 EC 78
0879:08 D0 D5 20 04 09 20 4C 4F
0881:0B 20 AA 0E AD 16 C8 49 F5
0889:01 8D 16 C8 20 06 0B F0 EC
0891:F0 A9 07 20 91 0A A9 03 72
0899:8D 24 C8 A9 09 8D 23 C8 BA
08A1:A9 0A 8D 31 C9 8D 32 C9 80
08A9:A2 01 A9 05 20 2A 10 20 BA
08B1:15 17 20 38 0F 20 8B 0F B8
08B9:AD 0A C8 F0 08 DE 31 C9 33
08C1:F0 10 4C CB 08 A9 0A 9D 2D
08C9:31 C9 CA 10 DD A2 01 4C 07
08D1:AB 08 AD 23 C8 C9 09 D0 F1
08D9:B8 AD 24 C8 C9 03 F0 05 04
08E1:C9 02 D0 AD 00 20 04 09 DD
08E9:4C 82 08 E6 FD D0 02 E6 4E
08F1:FE A5 FD C9 E0 D0 04 A5 3F
08F9:FE C9 C4 60 E6 FB D0 02 65
0901:E6 FC 60 AD C6 17 A2 00 84
0909:9D E0 C8 E8 E0 12 D0 F8 B3
0911:A9 FF A2 12 9D E0 C8 E8 58
0919:E0 3F D0 F8 AD C7 17 A2 72
0921:3F 9D E0 C8 E8 E0 51 D0 21
0929:F8 A2 00 8E 25 C8 A0 08 DE
0931:BD C8 17 9D 3E C8 99 86 DF
0939:C8 BD D1 17 9D 47 C8 99 FF
0941:7D C8 BD F0 17 9D 8F C8 22
0949:BD 0B 18 9D D7 C8 BD F9 31
0951:17 9D 98 C8 BD 02 18 9D B9
0959:CE C8 E8 88 10 D2 A9 03 CC
0961:8D E5 CA A9 4D 8D E6 CA E0
0969:A9 FF 8D 21 C8 20 B3 0B 4D
0971:A9 08 8D 23 C8 8D 24 C8 CB
0979:20 40 0D AD 0E C8 C9 FF 4F
0981:F0 06 20 A3 0C 4C 8C 09 7F
0989:20 FC 0B CE 23 C8 10 E8 7E
0991:A9 08 8D 23 C8 CE 24 C8 F0
0999:10 DE A9 04 8D 23 C8 8D F8
09A1:24 C8 20 40 0D A9 00 8D 9C
09A9:21 C8 20 5F 0C A9 01 8D 0F
09B1:66 C8 A9 03 8D 20 C8 8D 9A
09B9:24 C8 A9 16 8D 31 C9 A9 15
09C1:01 8D 23 C8 8D 1F C8 20 43
09C9:DE 09 EE 31 C9 EE 24 C8 99
09D1:AD 24 C8 C9 06 D0 E8 EE AD
09D9:24 C8 EE 24 C8 AD 31 C9 79
09E1:8D 02 C8 8D 00 C8 AD 24 CF
09E9:C8 8D 20 C8 20 58 0D 20 F0
09F1:A3 0C A9 00 8D 21 C8 20 B0
09F9:5F 0C A9 E1 8D 22 C8 20 B8
0A01:81 0C 60 8A 48 A9 08 8D 14
0A09:23 C8 8D 24 C8 20 40 0D 29
0A11:AD 0E C8 CD 1A C8 D0 14 1F
0A19:20 E6 0B AD 21 C8 8D 22 9C
0A21:C8 20 81 0C A9 FF AC 12 4B
0A29:C8 99 E0 C8 CE 23 C8 10 55
0A31:DC A9 08 8D 23 C8 CE 24 F5
0A39:C8 10 D2 68 AA 60 A9 4E 0F
0A41:8D F9 57 A9 DC 8D 03 D0 14
0A49:A9 A8 8D 02 D0 A9 00 8D E8
0A51:10 D0 A9 06 8D 28 D0 A9 8F
0A59:02 8D 1D D0 A9 03 8D 15 0C
0A61:D0 20 E4 FF C9 00 D0 D6 49
0A69:A9 07 20 91 0A EE 28 D0 5E
0A71:CE 27 D0 20 E4 FF C9 00 8D
0A79:F0 F9 C9 4E F0 0A C9 59 3F
0A81:D0 E6 A9 01 8D 15 D0 60 BF
0A89:A9 01 8D 15 D0 A9 00 60 43
0A91:8D FA 0A 8A 48 98 48 AE F8
0A99:FA 0A BD DA 0A A8 BD E2 64
0AA1:0A 99 05 D4 A9 00 99 06 95
0AA9:D4 BD EA 0A 99 01 D4 BD CD
0AB1:F2 0A 99 04 D4 49 01 99 9C
0AB9:04 D4 68 A8 68 AA 60 A2 ED
0AC1:18 A9 00 9D 00 D4 CA 10 1F
0AC9:FA A9 0F 8D 18 D4 A9 FF E7
0AD1:8D 0F D4 A9 80 8D 12 D4 D8
0AD9:60 00 07 00 00 07 00 00 1B
0AE1:00 01 19 04 19 8A 92 84 36
0AE9:19 17 10 50 0A 01 64 03 77
0AF1:1E 10 10 80 20 80 80 80 A7
0AF9:20 00 A0 A5 A2 8D EA 0A B7
0B01:A9 00 4C 91 0A AD 25 C8 A8
0B09:F0 3A C9 02 F0 22 C9 03 26
0B11:F0 33 A0 51 B9 3E C8 D0 BE
0B19:14 B9 E0 C8 CD C6 17 F0 F8
0B21:18 AD C6 17 8D 1A C8 20 7F
0B29:04 0A A9 01 60 88 10 E4 33
0B31:AD C6 17 8D 1A C8 20 04 C3
0B39:0A AD C7 17 8D 1A C8 20 B0
0B41:04 0A A9 01 60 20 04 09 B5
0B49:A9 00 60 AD 11 D0 09 20 19
0B51:8D 11 D0 AD 18 D0 09 08 85
0B59:29 0F 09 50 8D 18 D0 AD 0A
0B61:16 D0 09 10 8D 16 D0 AD EC
0B69:02 DD 09 03 8D 02 DD AD 27
0B71:00 DD 29 FC 09 02 8D 00 5F
0B79:DD 60 A9 00 A2 3F 9D C0 D9
0B81:53 CA 10 FA A9 FF A2 00 38
0B89:9D C0 53 9D F6 53 E8 E0 9A
0B91:06 D0 F5 A2 06 A9 80 9D 3D
0B99:C0 53 A9 01 9D C2 53 E8 B1
0BA1:E8 E8 E0 36 90 EF A2 3F AE
0BA9:BD 79 1B 9D 80 53 CA 10 31
0BB1:F7 60 A9 00 8D C3 0B A9 4C
0BB9:60 8D C4 0B AD 21 C8 A0 D0
0BC1:00 99 41 C9 C8 D0 FA EE 71
0BC9:C4 0B AE C4 0B E0 80 D0 D4
0BD1:F0 A0 00 A9 00 99 00 D8 62
0BD9:99 00 D9 99 00 DA 99 E8 19
0BE1:DA C8 D0 F1 60 A9 0B 8D 1E
0BE9:21 C8 AD 23 C8 18 6D 24 50
0BF1:C8 29 01 D0 05 A9 0F 8D 5E
0BF9:21 C8 60 20 0B 0D A9 FF C0
0C01:8D 22 C8 AC 12 C8 99 3E 71
0C09:C8 99 8F C8 99 E0 C8 A9 F5
0C11:00 8D 02 C8 20 46 0D 20 AD
0C19:25 0C 20 B3 0C 20 2F 0C 51
0C21:20 5C 0C 60 AD 2A C8 85 15
0C29:FD AD 2B C8 85 FE A0 00 07
0C31:AD 22 C8 91 FD C8 C0 18 87
0C39:D0 F9 60 AD 2A C8 85 FD 9C
0C41:AD 2B C8 85 FE AD 2C C8 3C
0C49:85 FB AD 2D C8 85 FC A0 A2
0C51:00 B1 FB 91 FD C8 C0 18 1B
0C59:D0 F7 60 20 E6 0B A0 00 8A
0C61:AD 2E C8 85 FD AD 2F C8 1B
0C69:85 FE AD 21 C8 91 FD C8 1D
0C71:91 FD C8 91 FD A0 28 91 58
0C79:FD C8 91 FD C8 91 FD 60 BD
0C81:A0 00 AD 30 C8 85 FD AD A8
```

```
0C89:31 C8 85 FE AD 22 C8 91 26
0C91:FD C8 91 FD C8 91 FD A0 16
0C99:28 91 FD C8 91 FD C8 91 1E
0CA1:FD 60 20 3C 0C 20 B3 0C EC
0CA9:20 50 0C 20 81 0C 20 5C 42
0CB1:0C 60 A5 FD 18 69 40 85 E8
0CB9:FD A5 FE 69 01 85 FE A5 72
0CC1:FB 18 69 18 85 FB A5 FC F0
0CC9:69 00 85 FC 60 AD 1B D4 DB
0CD1:29 0F C9 09 B0 F7 F0 F5 49
0CD9:60 AD 1B D4 6D 41 C9 29 6B
0CE1:7F C9 51 B0 F4 A8 60 49 B5
0CE9:FF 85 02 E6 02 A5 02 60 1D
0CF1:8C 23 C8 A9 00 8D 24 C8 14
0CF9:AD 23 C8 38 E9 09 30 09 2B
0D01:8D 23 C8 EE 24 C8 4C F9 89
0D09:0C 60 AD 24 C8 8D 35 C9 E9
0D11:0A 0A 0A 18 6D 35 C9 6D B6
0D19:23 C8 8D 35 C9 8D 12 C8 6D
0D21:A8 60 B9 3E C8 8D 00 C8 08
0D29:B9 8F C8 8D 10 C8 18 6D 37
0D31:00 C8 8D 02 C8 B9 E0 C8 07
0D39:8D 0E C8 8D 22 C8 60 20 A4
0D41:0B 0D 20 23 0D AD 24 C8 8A
0D49:18 69 01 8D 20 C8 AD 23 65
0D51:C8 18 69 02 8D 1F C8 AD 4B
0D59:20 C8 0A 8D 2B C8 A9 00 9F
0D61:8D 2A C8 AD 20 C8 4A 6E E7
0D69:2A C8 18 6D 2B C8 8D 2B 67
0D71:C8 AD 1F C8 0A 0A 0A 8D E5
0D79:35 C9 0A 18 6D 35 C9 18 4F
0D81:6D 2A C8 8D 2A C8 AD 2B C9
0D89:C8 69 00 8D 2B C8 AD 2A 3D
0D91:C8 18 69 48 8D 2A C8 AD 1C
0D99:2B C8 69 61 8D 2B C8 A9 13
0DA1:00 8D 36 C9 AD 02 C8 0A 93
0DA9:2E 36 C9 0A 2E 36 C9 0A 2A
0DB1:2E 36 C9 0A 2E 36 C9 8D B5
0DB9:35 C9 8D 37 C9 AD 36 C9 41
0DC1:8D 38 C9 0E 35 C9 2E 36 2E
0DC9:C9 AD 35 C9 18 6D 37 C9 26
0DD1:8D 35 C9 AD 36 C9 6D 38 FF
0DD9:C9 8D 36 C9 AD 35 C9 69 DE
0DE1:00 8D 2C C8 AD 36 C9 69 B4
0DE9:C0 8D 2D C8 A9 00 8D 35 97
0DF1:C9 8D 36 C9 AD 20 C8 0A 41
0DF9:2E 36 C9 0A 2E 36 C9 0A 7A
0E01:2E 36 C9 0A 2E 36 C9 8D 07
0E09:35 C9 8D 37 C9 AD 36 C9 92
0E11:8D 38 C9 0E 37 C9 2E 38 91
0E19:C9 0E 37 C9 2E 38 C9 AD B4
0E21:37 C9 18 6D 35 C9 8D 2E 3F
0E29:C8 AD 38 C9 6D 36 C9 8D 1E
0E31:2F C8 AD 1F C8 8D 35 C9 6F
0E39:0A 18 6D 35 C9 8D 35 C9 1A
0E41:18 6D 2E C8 8D 35 C9 AD 99
0E49:2F C8 69 00 8D 36 C9 AD E2
0E51:35 C9 18 69 29 8D 2E C8 B8
0E59:AD 36 C9 69 D8 8D 2F C8 CD
0E61:AD 35 C9 18 69 29 8D 30 97
0E69:C8 AD 36 C9 69 54 8D 31 A1
0E71:C8 AD 02 C8 0A 0A 8D 32 EF
0E79:C8 A9 20 18 6D 32 C8 8D 3D
0E81:32 C8 A9 18 69 00 8D 33 39
0E89:C8 AD 1F C8 0A 0A 8D 38 B1
0E91:C8 0A 18 6D 38 C8 69 04 2A
0E99:8D 38 C8 AD 20 C8 0A 0A C0
0EA1:0A 0A 18 69 0C 8D 39 C8 B0
0EA9:60 A9 00 8D 03 C8 8D 05 94
0EB1:C8 8D 14 C8 8D 26 C8 8D C8
0EB9:E3 CA 8D E4 CA A9 02 8D 09
0EC1:0C C8 A9 04 8D 23 C8 8D A3
0EC9:24 C8 AE 16 C8 BD C6 17 43
0ED1:8D 0F C8 20 B8 12 AD 25 22
0ED9:C8 D0 0B 20 E8 0E 20 15 C6
0EE1:17 AD 0C C8 D0 F0 60 A9 B7
0EE9:05 20 2A 10 20 E2 10 AE 32
0EF1:16 C8 20 38 0F AD 05 C8 D4
0EF9:F0 03 4C 01 0F 4C 2A 0F F5
0F01:20 D0 0F A9 14 8D 04 C8 87
0F09:20 0B 0D CC 13 C8 F0 03 09
0F11:4E 04 C8 AD 0A C8 F0 0B AB
0F19:AD 0B C8 F0 03 4C E2 0F 17
0F21:4C 39 10 8D 03 C8 4C 8B ED
0F29:0F AD 0A C8 F0 03 4C 39 6D
0F31:10 8D 03 C8 4C 8B 0F A9 FF
0F39:00 8D 0A C8 8D 0B C8 8D 40
0F41:17 C8 8D 1B C8 BD 00 DC 9A
0F49:8D 41 C9 4E 41 C9 B0 08 37
0F51:A9 FF 8D 1B C8 8D 0B C8 03
0F59:4E 41 C9 B0 08 A9 01 8D A9
0F61:1B C8 8D 0B C8 4E 41 C9 6D
0F69:B0 08 A9 FF 8D 17 C8 8D FE
0F71:0B C8 4E 41 C9 B0 08 A9 EF
0F79:01 8D 17 C8 8D 0B C8 4E 63
0F81:41 C9 B0 05 A9 01 8D 0A 8F
0F89:C8 60 AD 23 C8 18 6D 17 A4
0F91:C8 8D 23 C8 AD 24 C8 18 10
0F99:6D 1B C8 8D 24 C8 AD 24 EA
0FA1:C8 30 0E C9 09 F0 02 90 2F
0FA9:0D A9 00 8D 24 C8 4C B7 26
0FB1:0F A9 08 8D 24 C8 AD 23 5E
0FB9:C8 30 0E C9 0A F0 02 90 4F
0FC1:0D A9 00 8D 23 C8 4C CF 4E
0FC9:0F A9 09 8D 23 C8 60 AD 7E
0FD1:36 C8 85 FD AD 37 C8 85 2F
0FD9:FE A9 04 8D 21 C8 4C 5F 5F
0FE1:0C 20 0B 0D 20 23 0D AD 95
0FE9:12 C8 CD 13 C8 D0 35 AD CF
0FF1:10 C8 18 6D 0B C8 29 03 F4
0FF9:8D 34 C9 AD 00 C8 AA BD 36
1001:DA 17 2D 34 C9 8D 34 C9 F3
1009:AC 12 C8 AD 34 C9 99 8F 83
1011:C8 20 23 0D 20 46 0D 20 27
1019:A3 0C A9 1E 20 2A 10 A9 98
1021:02 4C 91 0A A9 01 4C 91 A3
1029:0A 18 65 A2 8D 38 10 A5 3E
1031:A2 CD 38 10 D0 F9 60 00 4D
1039:AE 03 C8 EC 04 C8 F0 05 83
1041:E8 8E 03 C8 60 A9 00 8D 9D
1049:03 C8 AD 23 C8 C9 09 D0 55
1051:03 4C C0 12 AD 05 C8 49 9B
1059:01 8D 05 C8 F0 4E 20 40 CB
1061:0D AD 0E C8 CD 0F C8 D0 CE
1069:39 AD 12 C8 8D 13 C8 A9 54
1071:02 20 91 0A AD 2E C8 8D B2
1079:36 C8 AD 2F C8 8D 37 C8 43
1081:AD 0C C8 8D 0D C8 AD 10 64
1089:C8 8D 11 C8 8D 34 C9 AD 9E
1091:23 C8 8D 06 C8 AD 24 C8 95
1099:8D 07 C8 AD 00 C8 8D 01 75
10A1:C8 60 A9 00 8D 05 C8 A9 2F
10A9:01 4C 91 0A A9 0A 8D 04 C4
10B1:C8 AD 23 C8 CD 06 C8 D0 7B
10B9:16 AD 24 C8 CD 07 C8 D0 4E
10C1:0E A9 00 8D 15 C8 20 89 C1
10C9:12 20 86 17 4C DF 10 20 5F
10D1:25 11 F0 42 20 89 12 F0 47
10D9:3D 20 CF 11 F0 38 4C B8 65
10E1:12 AE 16 C8 BD E5 CA 10 31
10E9:05 A9 01 8D 26 C8 A0 02 87
10F1:AD 0F C8 99 BC 55 AE 0C 04
10F9:C8 E0 02 F0 02 A9 00 99 56
1101:E4 55 AD 0F C8 AE 26 C8 A7
1109:F0 02 A9 00 99 AC 56 99 1F
1111:D4 56 88 10 DB 60 AD 0D 0E
1119:C8 8D 0C C8 A9 01 8D 05 82
1121:C8 4C 91 0A AD 23 C8 38 51
1129:ED 06 C8 8D 18 C8 10 03 BC
1131:20 E8 0C 8D 35 C9 AD 24 48
1139:C8 38 ED 07 C8 8D 1C C8 79
1141:10 03 20 E8 0C 18 6D 35 8F
1149:C9 8D 15 C8 F0 08 C9 01 1F
1151:F0 04 C9 02 F0 03 A9 01 2E
1159:60 20 0B 0D AD 18 C8 C9 0F
1161:FE D0 06 B9 3F C8 4C C6 10
1169:11 AD 18 C8 C9 02 D0 06 0D
1171:B9 3D C8 4C C6 11 AD 1C 8F
1179:C8 C9 02 D0 06 B9 35 C8 0A
1181:4C C6 11 AD 1C C8 C9 FE 0F
1189:D0 06 B9 47 C8 4C C6 11 57
1191:AD 18 C8 C9 01 D0 0A B9 5F
1199:3D C8 C9 FF F0 2D 4C A9 44
11A1:11 B9 3F C8 C9 FF F0 23 82
11A9:AD 1C C8 C9 01 D0 0C A0 63
11B1:00 B9 35 C8 C9 FF F0 13 B8
11B9:4C C3 11 B9 47 C8 C9 FF A1
11C1:F0 09 A9 00 60 C9 FF F0 EE
11C9:02 D0 F7 A9 01 60 20 0B 8F
11D1:0D C0 28 D0 2D AD E4 CA 71
11D9:D0 3E A9 05 20 91 0A A9 7E
11E1:00 8D 41 C9 EE E4 CA 20 EC
11E9:DA 0C C0 28 F0 0D B9 3E 84
11F1:C8 C9 FF D0 06 20 F1 0C 98
11F9:4C 2F 12 EE 41 C9 D0 E7 F9
1201:F0 16 B9 3E C8 C9 FF F0 9C
1209:25 AD 01 C8 C9 00 F0 0B 13
1211:C9 01 F0 20 C9 06 F0 03 C5
1219:A9 00 60 AD 14 C8 D0 F8 57
1221:EE 14 C8 B9 3E C8 20 EE BA
1229:15 A9 06 20 91 0A 20 95 8F
1231:17 A9 01 60 AD E3 CA F0 F4
1239:03 4C 19 12 20 DA 0C B9 74
1241:3E C8 C9 FF D0 07 C0 28 3C
1249:F0 03 4C 56 12 EE 41 C9 2E
1251:D0 EA 4C 19 12 8C 42 C9 C4
1259:AE 12 C8 20 73 17 EE E3 2E
1261:CA AD 23 C8 48 AD 24 C8 51
1269:48 AC 42 C9 20 F1 0C 20 C2
1271:40 0D A9 05 20 91 0A 20 F9
1279:A3 0C 68 8D 24 C8 68 8D FA
1281:23 C8 20 95 17 A9 01 60 88
1289:AD 0C C8 8D 35 C9 AD 34 D9
1291:C9 CD 11 C8 F0 05 CE 35 2B
1299:C9 30 10 AD 15 C8 F0 0E 47
12A1:CE 15 C8 CE 35 C9 30 03 AC
12A9:4C 9C 12 A9 00 60 AD 35 0A
12B1:C9 8D 0C C8 A9 01 60 A9 E7
12B9:01 8D 15 D0 4C 91 0A AD DB
12C1:24 C8 C9 02 D0 06 20 3F A1
12C9:0A F0 EC 00 C9 03 D0 10 D8
12D1:20 3F 0A F0 E2 A9 00 8D 71
12D9:0C C8 A9 03 8D 25 C8 60 8E
12E1:AD 05 C8 D0 D2 AD 24 C8 A2
12E9:C9 04 D0 06 A9 00 8D 0C E2
12F1:C8 60 C9 07 F0 01 60 AD 36
12F9:26 C8 D0 BB AE 16 C8 A9 42
1301:02 8D 28 C8 8D 29 C8 A2 62
1309:02 A9 04 9D 51 C9 A9 FF F9
1311:9D 45 C9 9D 55 C9 9D 57 CE
1319:C9 E8 E0 06 D0 EB A9 00 64
1321:AA 9D 63 C9 E8 E0 60 D0 69
1329:F8 EE 26 C8 CE 0C C8 A2 B3
1331:51 BD 3E C8 C9 12 F0 06 42
1339:CA 10 F6 4C B8 12 BD E0 D6
1341:C8 CD 0F C8 D0 F2 8A AE C3
1349:16 C8 9D E5 CA A8 20 F1 E9
1351:0C AD 23 C8 8D 47 C9 AD A4
1359:24 C8 8D 4D C9 8D 24 C8 DF
1361:20 0B 0D B9 8F C8 8D 53 A5
1369:C9 A9 00 8D 27 C8 A0 02 57
1371:8C 28 C8 B9 51 C9 C9 04 E5
1379:F0 1D B9 45 C9 8D 23 C8 7E
1381:B9 4B C9 8D 24 C8 20 40 2E
1389:0D 20 69 15 AC 28 C8 20 74
1391:74 16 F0 03 20 0E 15 EE 18
1399:28 C8 AC 28 C8 CC 29 C8 B2
13A1:F0 D1 90 CF AD 27 C8 D0 30
13A9:C0 A9 0F 8D 33 C9 A9 0A 73
13B1:8D 1E C8 AD 1E C8 20 2A 98
13B9:10 CE 1E C8 10 03 EE 1E 74
13C1:C8 A2 00 8E 27 C8 20 38 B2
13C9:14 BD 63 C9 F0 08 20 64 BE
13D1:14 A9 01 8D 27 C8 AD 33 50
13D9:C9 D0 35 BD 63 C9 48 BD 2C
13E1:C3 C9 48 BD 34 03 9D 63 8D
13E9:C9 BD 94 03 9D C3 C9 20 D6
13F1:38 14 BD C3 C9 9D 94 03 1E
13F9:BD 63 C9 9D 34 03 F0 08 82
1401:A9 01 8D 27 C8 20 64 14 06
1409:68 9D C3 C9 68 9D 63 C9 2C
1411:E8 E0 60 90 B1 AD 33 C9 6F
1419:F0 12 CE 33 C9 D0 0D A9 A0
1421:77 8D 1A C8 20 04 0A A9 07
1429:08 8D 1E C8 AD 27 C8 F0 95
1431:03 4C B4 13 4C 69 09 BD 8D
1439:63 C9 F0 26 18 7D 23 CA CD
1441:C9 9F 90 08 A9 00 9D 63 B4
1449:C9 4C 63 14 9D 63 C9 BD E2
```

```
1451:C3 C9 18 7D 83 CA 9D C3 EE
1459:C9 C9 C7 90 05 A9 00 9D 47
1461:63 C9 60 8A 48 98 48 BD 55
1469:63 C9 4A 4A 8D 0D 15 BD 2C
1471:C3 C9 4A 4A 4A 8D 0C 15 91
1479:A9 00 8D 36 C9 AD 0C 15 BD
1481:0A 2E 36 C9 0A 2E 36 C9 DC
1489:0A 2E 36 C9 0A 2E 36 C9 E4
1491:0A 2E 36 C9 0A 2E 36 C9 EC
1499:8D 35 C9 AD 36 C9 18 6D 60
14A1:0C 15 8D 36 C9 A9 00 8D AC
14A9:38 C9 AD 0D 15 0A 2E 38 4C
14B1:C9 0A 2E 38 C9 0A 2E 38 95
14B9:C9 8D 37 C9 18 6D 35 C9 58
14C1:8D 35 C9 AD 36 C9 6D 38 FD
14C9:C9 8D 38 C9 8D 36 C9 BD 74
14D1:C3 C9 29 07 18 6D 35 C9 8E
14D9:8D 35 C9 AD 36 C9 69 60 36
14E1:8D 36 C9 8E 41 C9 BD 63 90
14E9:C9 29 03 AA AD 35 C9 85 A7
14F1:FD AD 36 C9 85 FE A0 00 51
14F9:B1 FD 5D 08 15 91 FD AE 40
1501:41 C9 68 A8 68 AA 60 C0 45
1509:30 0C 03 00 00 20 40 0D BC
1511:AC 28 C8 AD 12 C8 C9 28 FE
1519:D0 06 A9 04 99 51 C9 60 A8
1521:AD 0E C8 C9 FF D0 04 EE 95
1529:27 C8 60 AD 32 C8 85 FD BD
1531:AD 33 C8 85 FE AC 28 C8 34
1539:B9 51 C9 8D 32 C9 A8 B1 62
1541:FD 8D 31 C9 10 0C EE 27 46
1549:C8 AD 32 C9 8D 31 C9 4C 37
1551:D0 15 C9 04 D0 09 AC 28 CE
1559:C8 99 51 C9 4C 0D 16 EE C6
1561:27 C8 AC 28 C8 4C BD 16 72
1569:AE 28 C8 BD 63 C9 48 BD 7A
1571:C3 C9 48 BD 23 CA 48 BD 67
1579:83 CA 48 BD 51 C9 A8 B9 B9
1581:14 18 9D 23 CA 6D 39 C8 E8
1589:69 03 9D C3 C9 B9 18 18 96
1591:9D 83 CA 6D 38 C8 69 05 58
1599:9D 63 C9 B9 1C 18 8D 19 B5
15A1:C8 20 38 14 BD 63 C9 F0 80
15A9:10 20 64 14 20 FC 0A AD 68
15B1:1B D4 30 05 A9 01 20 2A B0
15B9:10 CE 19 C8 D0 E3 68 9D D3
15C1:83 CA 68 9D 23 CA 68 9D F9
15C9:C3 C9 68 9D 63 C9 60 AC DE
15D1:28 C8 EE 31 C9 20 BD 16 93
15D9:AC 29 C8 C8 C0 06 90 01 8A
15E1:60 8C 29 C8 CE 31 C9 CE AE
15E9:31 C9 4C BD 16 D0 04 EE 6F
15F1:25 C8 60 C9 12 D0 14 8A 10
15F9:48 A2 00 AD C6 17 D9 E0 F2
1601:C8 F0 01 E8 A9 FF 9D E5 EA
1609:CA 68 AA 60 20 40 0D AD D9
1611:00 C8 20 EE 15 A9 77 8D 2E
1619:22 C8 20 81 0C A9 77 AC 47
1621:12 C8 99 E0 C8 A2 00 BD 58
1629:63 C9 F0 06 E8 E0 60 D0 54
1631:F6 60 8E 41 C9 A9 04 20 F3
1639:91 0A A0 00 AE 41 C9 20 F2
1641:CE 0C 6D 38 C8 9D 63 C9 56
1649:9D 34 03 20 CE 0C 6D 39 6E
1651:C8 9D C3 C9 9D 94 03 20 C3
1659:AD 16 9D 23 CA 20 AD 16 10
1661:9D 83 CA 1D 23 CA F0 EF 7E
1669:E8 C8 E0 60 F0 04 C0 20 97
1671:D0 CD 60 B9 51 C9 AA BD E5
1679:14 18 18 79 45 C9 30 24 26
1681:C9 09 F0 20 B0 1E 99 45 6B
1689:C9 8D 23 C8 BD 18 18 18 85
1691:79 4B C9 30 0F C9 09 F0 2C
1699:0B B0 09 99 4B C9 8D 24 F2
16A1:C8 A9 01 60 A9 04 99 51 A4
16A9:C9 A9 00 60 AD 1B D4 30 DE
16B1:04 20 CE 0C 60 20 CE 0C AF
16B9:20 E8 0C 60 AD 31 C9 29 A6
16C1:03 8D 31 C9 AD 00 C8 C9 5E
16C9:0E D0 36 A2 00 BD 57 C9 91
16D1:CD 12 C8 D0 0E BD 5D C9 7B
16D9:CD 31 C9 D0 06 A9 04 99 F7
16E1:51 C9 60 E8 E0 06 D0 E5 6A
16E9:A2 00 BD 57 C9 30 06 E8 98
16F1:E0 06 D0 F6 60 AD 12 C8 40
16F9:9D 57 C9 AD 31 C9 9D 5D 28
1701:C9 AD 31 C9 99 51 C9 AD 95
1709:23 C8 99 45 C9 AD 24 C8 98
1711:99 4B C9 60 A9 07 8D 27 C9
1719:D0 AD 24 C8 0A 0A 0A 0A C2
1721:69 48 8D 42 C9 AD 23 C8 FF
1729:C9 09 F0 35 0A 0A 0A 8D 0A
1731:41 C9 0A 6D 41 C9 69 50 DE
1739:8D 41 C9 90 08 A9 01 8D 37
1741:10 D0 4C 4B 17 A9 00 8D D6
1749:10 D0 AD 41 C9 8D 00 D0 D2
1751:AD 42 C9 8D 01 D0 A9 01 98
1759:8D 15 D0 A9 4F 8D F8 57 42
1761:60 A9 00 8D 10 D0 A9 38 52
1769:8D 00 D0 AD 42 C9 8D 01 A8
1771:D0 60 BD 3E C8 99 3E C8 AD
1779:BD 8F C8 99 8F C8 BD E0 19
1781:C8 99 E0 C8 60 AD 36 C8 12
1789:8D 2E C8 AD 37 C8 8D 2F 25
1791:C8 4C 5C 0C 20 86 17 20 EC
1799:0B 0D AE 13 C8 20 73 17 5C
17A1:AD 23 C8 48 AD 24 C8 48 E4
17A9:AD 06 C8 8D 23 C8 AD 07 C0
17B1:C8 8D 24 C8 20 FC 0B 68 2C
17B9:8D 24 C8 68 8D 23 C8 20 02
17C1:40 0D 4C A3 0C D5 A2 0A 1E
17C9:0A 04 12 00 01 04 0A 0A 76
17D1:0A 06 06 0E 02 02 06 06 52
17D9:0A 00 00 01 01 01 01 03 2E
17E1:03 03 03 03 03 03 03 03 10
17E9:03 03 03 03 03 03 03 03 18
17F1:03 00 02 00 00 01 02 02 EB
17F9:02 02 02 00 01 00 02 02 F7
1801:03 00 00 00 00 01 02 00 BA
1809:00 01 01 01 01 00 00 00 B1
1811:00 00 00 00 01 00 FF FF 49
1819:00 01 00 10 0C 10 0C 04 47
1821:04 04 04 00 01 02 03 00 EA
1829:03 02 01 02 01 00 03 03 AC
1831:02 01 00 01 00 03 02 04 C6
1839:04 00 04 04 04 04 01 02 60
1841:04 04 04 04 03 04 04 04 69
1849:00 03 04 04 04 01 00 03 22
1851:02 04 04 01 04 04 02 04 4C
1859:00 FF 00 01 04 01 FF FF BD
1861:02 04 02 03 FF 03 04 04 1C
1869:04 04 04 04 04 04 04 04 99
1871:04 04 04 04 04 04 04 93 31
1879:11 43 4F 50 59 52 49 47 DF
1881:48 54 20 31 39 38 37 20 3B
1889:43 4F 4D 50 55 54 45 21 85
1891:20 50 55 42 4C 49 43 41 04
1899:54 49 4F 4E 53 0D 41 4C B2
18A1:4C 20 52 49 47 48 54 53 36
18A9:20 52 45 53 45 52 56 45 C1
18B1:44 0D 11 11 12 41 4E 59 06
18B9:20 4B 45 59 20 54 4F 20 1C
18C1:43 4F 4E 54 49 4E 55 45 E9
18C9:00 00 81 4C 0F 03 FE FE 70
18D1:FE FE FE 03 FB FB FB EA 04
18D9:A6 03 EF EF EF EF AF FE B8
18E1:FE FE 01 FE 03 95 A6 AA C7
18E9:01 AA 03 AF AF AF 01 AF 8E
18F1:05 FA FB FB FB FB FB 02 6C
18F9:AA 02 55 7D 7D 02 AF EF C5
1901:EF EF EF EF FB FB FB FB E7
1909:FB FA 02 7D 7D 55 02 AA FF
1911:02 EF EF EF EF EF AF 0C E7
1919:E7 E7 E7 E7 E7 E7 10 E7 9B
1921:E7 E7 E7 E7 E7 11 FA 07 3D
1929:AA 07 AF F5 07 55 07 5F C2
1931:0A FA FD 07 7F AF D7 FA BC
1939:10 FD 07 7F AF DF 13 FD EC
1941:FA 04 DF AF 7F 03 FD FA E7
1949:04 D7 AF 7F 10 F5 01 FA B6
1951:FA FA 03 55 EB AA AA AA 7F
1959:03 5F 01 AF AF AF FA FA 2D
1961:FA FA FA 03 AA AA AA AA 5F
1969:AA 03 AF AF AF AF AF 06 44
1971:FA FA FA FA FA 03 AA AA D2
1979:AA AA AA 03 DF DF DF 9F 0F
1981:9F FA FA FA FA FA 03 AA C5
1989:AA AA AA AA 03 9F 9F DF 71
1991:DF DF 06 FA FA FA FA FA D0
1999:03 AA AA AA AA AA 03 AF AD
19A1:AF AF AF AF FA FA FA 01 43
19A9:F5 03 AA AA AA EB 55 03 4A
19B1:AF AF AF 01 5F 06 F7 F7 A8
19B9:F7 F6 F6 03 AA AA AA AA B4
19C1:AA 03 AF AF AF AF AF F6 8D
19C9:F6 F7 F7 F7 03 AA AA AA B6
19D1:AA AA 03 AF AF AF AF AF AA
19D9:10 FD F6 DA 03 DF 6F AF 46
19E1:AF AF 01 FD F6 DA 04 6A 6D
19E9:AA AA AA 04 AF AF AF AF FC
19F1:07 F7 F9 FA FA FA 05 7F E1
19F9:9F A7 08 FA FA FA FA 04 54
1A01:A9 AA AA AA 05 7F 9F A7 C1
1A09:07 DA F6 FD 05 AA AA AA 09
1A11:6A DA 03 AF AF AF AF AF D7
1A19:08 F6 FD 06 AF AF 6F DF 2A
1A21:07 FA FA FA FA FA 03 AA 1B
1A29:AA AA A9 A7 03 A7 9F 7F 82
1A31:02 FA FA F9 F7 04 9F 7F B2
1A39:14 FD F5 03 F7 D5 55 55 FC
1A41:55 05 7F 5F 57 FA FA 06 E9
1A49:AA AA 06 AB AB 0A FE FE 7B
1A51:FE FE 04 7F 5F 57 55 08 48
1A59:FE FE FE FE FE 03 55 55 A0
1A61:57 5F 7F 03 7F 0D FA FA 5A
1A69:06 AA AA 06 AB AB F5 FD F6
1A71:06 55 55 55 D5 F7 03 57 E9
1A79:5F 7F 11 FD F5 D5 55 04 F4
1A81:BF BF BF BF FD 07 55 55 85
1A89:D5 F5 FD 03 BF BF BF BF 52
1A91:BF 0E FB F7 FB F7 FB 10 EF
1A99:EA E6 A6 66 6A AA 04 BF FD
1AA1:7F 7F BF 07 F9 FA FA 05 94
1AA9:BF BF AB 08 F9 FA FA F9 4F
1AB1:04 66 AB BF BF 04 6F 12 F1
1AB9:AA 6A 66 A6 E6 03 BF 7F 57
1AC1:7F BF 09 EA FB F7 FB F7 25
1AC9:FB 17 FE FE EA 05 6F AF 8B
1AD1:AF F9 07 99 EA FE FE 04 2C
1AD9:6F AF AF 6F 05 D5 DF DF BD
1AE1:DF DE DE DE 01 55 01 EB D2
1AE9:BE 04 57 F7 F7 F7 B7 B7 AF
1AF1:B7 DE DE DF DF DF DF D5 A7
1AF9:03 BE EB 02 55 01 B7 B7 D2
1B01:F7 B7 E7 F7 57 02 D5 DF EB
1B09:DE DE DE DE DE 01 55 01 D6
1B11:AA 02 FE AB 01 57 F7 F7 05
1B19:B7 B7 F7 F7 DE DE DE DE A6
1B21:DE DF D5 01 FB FE FE 03 66
1B29:55 01 F7 F7 F7 B7 B7 F7 CE
1B31:57 03 FE FE FE FE FE 03 98
1B39:AA AA AA AA AA 03 BF BF 10
1B41:BF BF BF 02 FE FE FE FE 50
1B49:FE 03 AA AA AA AA AA 03 18
1B51:BF BF BF BF BF 05 FE FA 56
1B59:EA EA EA 02 AA AA 00 2A 67
1B61:2A 00 03 BF AF AB AB AB 38
1B69:EA EA FA FE 04 2A 2A 2A 66
1B71:AA AA 03 AB AB AF BF 04 62
1B79:00 00 00 00 00 00 0C E7 AF
1B81:80 12 94 00 1E E7 00 12 B1
1B89:94 00 12 97 80 00 00 00 C9
1B91:12 64 80 12 94 80 0C 94 6E
1B99:80 04 94 80 04 63 00 00 59
1BA1:00 06 74 B9 E9 84 A5 02 32
1BA9:64 B9 C4 14 A5 04 E3 25 84
1BB1:E0 00 00 04 00 00 00 00 98
```

Program 3: *Laser Chess* For Atari XL And XE

Version by Rhett Anderson, Assistant Editor

```
BO 0 ? "{CLEAR}":POKE 710,4:
     CLR :DIM S$(7680),ML$(4
     0),SI(26):AS=ADR(S$):S$
     (1)=CHR$(0):S$(7680)=CH
```

```
   R$(0):S$(2)=S$:GOSUB 95
   00
BK 1 GRAPHICS 0:POKE 710,0:?
   "LASER CHESS (tm)":? "
   1 JOYSTICK OR 2";:INPUT
   NUMJOY:IF NUMJOY<>1 AN
   D NUMJOY<>2 THEN 1
LB 2 REM COPYRIGHT 1987 COMP
   UTE! PUBLICATIONS, INC.
   (3 SPACES)ALL RIGHTS RE
   SERVED
CP 3 PLAYER=1:PMBASE=54279:R
   AMTOP=106:SDMCTL=559:GR
   ACTL=53277:HPOS0=53248:
   PCOLR0=704:MOVE=2
CK 4 CN0=0:CN1=1:CN2=2:CN3=3
   :CN4=4:CN5=5:CN9=9:CN20
   =20:CN99=99:S$(192*40)=
   CHR$(0):? "(CLEAR)Lase
   r Chess(tm)";:GOSUB 250
   00
CL 5 DIM CLR(9,9),PIECE(9,9)
   ,ORIENT(9,9),A$(100),A(
   11),TURNS(8),DIRCK(8,15
   ),DIRX(3),DIRY(3),DIR(3
   ),BEAMCK$(399)
LL 6 DIM LX(3),LY(3),ALIVE(3
   ),NLX(3),NLY(3),TERM(3)
   ,DRK(3):GRAPHICS 2+16
NJ 7 A$="n  laser chess (TM)
   nn(3 SPACES)COPYRIGHT 1
   987n(6 SPACES)COMPUTE!n
   PUBLICATIONS, INC.nn(a)
   ll rights reserved"
NL 8 BEAMCK$(399)=CHR$(0):BE
   AMCK$(1,1)=CHR$(0):BEAM
   CK$(2)=BEAMCK$
DI 10 FOR X=CN0 TO CN9:FOR Y
   =CN0 TO CN9:PIECE(X,Y)
   =CN0:LET CLR(X,Y)=CN0
DP 19 D=10-ABS(X-Y):E=X*Y*3+
   11:IF E=11 THEN 22
JE 20 SOUND CN1,E,10,D:POKE
   709,D
AK 21 SOUND CN2,E+CN1,10,D
LI 22 T=T+CN1:IF T>LEN(A$) T
   HEN 30
FI 23 IF A$(T,T)="n" THEN PR
   INT #6:GOTO 30
EM 24 PRINT #6;A$(T,T);
MM 30 NEXT Y:NEXT X
HC 50 RESTORE 62:FOR Y=CN1 T
   O CN2:FOR X=CN1 TO CN9
   :LET CLR(X,Y)=CN1:LET
   CLR(X,Y+7)=CN2
MG 60 READ DUM1:READ DUM2:RE
   AD DUM3:PIECE(X,Y)=DUM
   1:PIECE(10-X,10-Y)=DUM
   1
OD 61 ORIENT(X,Y)=DUM2+CN1:O
   RIENT(10-X,10-Y)=DUM3+
   CN1:NEXT X:NEXT Y
NJ 62 DATA 8,2,0,8,2,0,1,1,1
   ,2,2,0,4,0,0
OH 64 DATA 6,0,0,1,0,0,8,3,1
   ,8,3,1,8,3,1
MM 66 DATA 5,2,0,5,2,0,7,2,0
   ,3,0,0,3,1,1
LA 68 DATA 5,2,0,5,2,0,8,2,0
BI 70 FOR I=CN1 TO 8:READ DU
   M:TURNS(I)=DUM:NEXT I:
   DATA 1,3,1,0,3,0,3,3
EJ 80 FOR I=CN1 TO 8:FOR J=C
   N0 TO TURNS(I):FOR K=C
   N0 TO CN3:READ DUM:DIR
   CK(I,J+K*CN4)=DUM:NEXT
   K:NEXT J:NEXT I
JE 82 DATA 1,0,3,2,3,2,1,0,-
   1,-1,-1,-1,-1,-1,-1,-1
JG 84 DATA -1,-1,-1,-1,-1,-1
   ,-1,-1,2,1,0,3,0,3,2,1
HF 86 DATA -1,-1,-1,-1,-1,-1
   ,0,-1,-1,-1,-1,1,2,-1,
   -1,-1
HF 88 DATA -1,3,-1,-1,0,1,2,
   3,-2,2,-1,2,3,-2,3,-1
IP 90 DATA -1,0,-2,0,1,-1,1,
   -2,-1,0,3,-1,-1,-1,1,0
PJ 92 DATA 1,-1,-1,2,3,2,-1,
   -1
IJ 95 GRAPHICS CN9
LP 100 FOR I=CN0 TO CN3:READ
   DUM1,DUM2:DIRX(I)=DU
   M1:DIRY(I)=DUM2:NEXT
   I:DATA 0,-1,1,0,0,1,-
   1,0
KD 101 VTAB=PEEK(134)+PEEK(1
   35)*256:ATAB=PEEK(140
   )+PEEK(141)*256
BG 102 OFFS=PEEK(88)+256*PEE
   K(89)-ATAB:HI=INT(OFF
   S/256):LO=OFFS-HI*256
   :OFFS2=(INT(AS/1024)+
   CN1)*1024-ATAB
OH 103 POKE VTAB+CN2,LO:POKE
   VTAB+CN3,HI
FL 112 COLOR 6:PLOT CN0,CN0:
   DRAWTO 79,CN0:COLOR 9
   :PLOT CN0,CN1:DRAWTO
   79,CN1
NA 113 S$(81,192*40)=S$
OO 114 COLOR CN0:PLOT 6,CN5:
   DRAWTO 60,CN5:PLOT 6,
   6:DRAWTO 60,6:S$(281,
   7440)=S$(201)
LL 117 COLOR 6:FOR I=86 TO 1
   05 STEP CN2:PLOT 31,I
   :DRAWTO 36,I:NEXT I:C
   OLOR CN9:FOR I=87 TO
   105 STEP CN2:PLOT 31,
   I
AL 118 DRAWTO 36,I:NEXT I
MB 119 GRAPHICS 63:POKE 710,
   9:POKE 708,132:POKE 7
   09,84:COLOR 1:PLOT 12
   ,5:DRAWTO 12,186:DRAW
   TO 122,186:DRAWTO 122
   ,5:DRAWTO 12,5
NE 120 PLOT CN0,CN0:DRAWTO 1
   59,CN0:DRAWTO 159,191
   :DRAWTO CN0,191:DRAWT
   O CN0,CN0
HG 121 GOSUB 21080:HI=INT(OF
   FS2/256):LO=OFFS2-HI*
   256
LA 122 A=INT((AS/1024)+CN1)*
   CN4:POKE PMBASE,A:MYB
   ASE=256*A:POKE SDMCTL
   ,46:POKE GRACTL,CN3
IL 123 FOR I=CN1 TO CN3:POKE
   53256+I,CN3:POKE VTA
   B+CN2,LO:POKE VTAB+CN
   3,HI
CP 124 POKE PCOLR0+I,8:POKE
   HPOS0+I,(I-CN1)*32+62
   :NEXT I:POKE 623,4:PO
   KE 53254,166:POKE 532
   55,158:POKE 53260,255
LA 125 FOR I=CN0 TO 89:HH=CN
   0:IF INT(INT(I/10)/CN
   2)=INT(I/10)/CN2 THEN
   HH=CN1
AP 126 POKE MYBASE+787+I,113
   :IF HH THEN POKE MYBA
   SE+787+I,142
OE 127 POKE MYBASE+915+I,199
   :IF HH THEN POKE MYBA
   SE+915+I,56
IP 128 POKE MYBASE+403+I,0:I
   F HH THEN POKE MYBASE
   +403+I,224
OH 129 POKE MYBASE+659+I,28:
   IF HH THEN POKE MYBAS
   E+659+I,227
AF 130 NEXT I:FOR PX=CN1 TO
   CN9:FOR PY=CN1 TO CN9
   :GOSUB 8000:NEXT PY:N
   EXT PX:PX=CN1:PY=PX:G
   OSUB 8300:GOSUB 8900
DE 131 S=STICK((PLAYER-CN1)*
   NUMJOY=CN2):IF MOVE=C
   N2 THEN HYCUBE=CN0:FI
   RED=CN0:HYSQ=CN0:TAKE
   N=1
EJ 135 IF STRIG((PLAYER-CN1)
   *NUMJOY=CN2)=0 THEN 2
   00
MG 136 IF SEL>CN0 AND PEEK(7
   64)=CN0 THEN POKE 764
   ,255:GOTO 800
OF 138 IF S=15 THEN 131
MA 139 IF STICK((PLAYER-CN1)
   *NUMJOY=CN2)<>15 THEN
   139
OB 140 GOSUB 8400
MG 141 PX=PX+(S=7)-(S=11)
AI 150 PY=PY+(S=13)-(S=14)
PC 160 PX=PX+(PX=CN0)*CN9-(P
   X=10)*CN9
PH 170 PY=PY+(PY=CN0)*CN9-(P
   Y=10)*CN9
JG 175 POKE 77,0
DE 180 GOSUB 8300
PD 185 IF SEL=CN1 THEN TX=PX
   :TY=PY:PX=SPX:PY=SPY:
   GOSUB 8600:PX=TX:PY=T
   Y
GI 190 GOTO 131
MH 200 IF SEL>CN0 THEN 400
JC 210 IF CLR(PX,PY)=PLAYER
   THEN SEL=1:SPX=PX:SPY
   =PY
NP 215 IF STRIG((PLAYER-CN1)
   *NUMJOY=CN2)=CN0 THEN
   215
GK 220 GOTO 139
OG 280 COLOR PLAYER
JK 400 FPX=PX:FPY=PY:IF PX=S
   PX AND PY=SPY THEN 70
   0
IA 410 IF ABS(PX-SPX)+ABS(PY
   -SPY)>MOVE THEN 139
HO 420 IF ABS(PX-SPX)=CN2 TH
   EN IF PIECE((PX+SPX)/
   CN2,PY)<>CN0 THEN 139
IC 430 IF ABS(PY-SPY)=CN2 TH
   EN IF PIECE(PX,(PY+SP
   Y)/CN2)<>CN0 THEN 139
PI 440 IF ABS(PY-SPY)<>CN1 O
   R ABS(PX-SPX)<>CN1 TH
   EN 450
CA 445 IF ((PIECE(PX,SPY)<>C
   N0 OR (PX=CN5 AND SPY
   =CN5)) AND (PIECE(SPX
   ,PY)<>CN0 OR (SPX=CN5
   AND PY=CN5))) THEN 1
   39
IE 450 IF PIECE(PX,PY)<>CN0
   THEN IF PIECE(SPX,SPY
   )<>CN4 AND PIECE(SPX,
   SPY)<>CN5 AND PIECE(S
   PX,SPY)<>6 THEN 139
CB 455 IF PX=CN5 AND PY=CN5
   THEN 650
FM 460 IF PIECE(PX,PY)<>CN0
   THEN 500
EL 470 TX=PX:TY=PY:PX=SPX:PY
   =SPY:GOSUB 8200:PX=TX
   :PY=TY
GE 475 FOR T=0 TO 10:SOUND 0
   ,100,10,10-T:NEXT T
CE 480 PIECE(PX,PY)=PIECE(SP
   X,SPY):LET CLR(PX,PY)
   =CLR(SPX,SPY):ORIENT(
   PX,PY)=ORIENT(SPX,SPY
   ):GOSUB 8000
OH 490 PIECE(SPX,SPY)=CN0:LE
   T CLR(SPX,SPY)=CN0
FF 491 MOVE=MOVE-ABS(FPX-SPX
```

```
    )-ABS(FPY-SPY):IF MOV
    E=CNØ THEN MOVE=CN2:P
    LAYER=(PLAYER=CN1)+CN
    1:GOSUB 2800
BC 492 PX=FPX:PY=FPY:GOSUB 8
    700:GOTO 131
MH 500 IF PIECE(SPX,SPY)=CN4
     OR PIECE(SPX,SPY)=CN
    5 THEN IF TAKEN THEN
    TAKEN=Ø:GOTO 600
EL 501 IF PIECE(SPX,SPY)=CN4
     OR PIECE(SPX,SPY)=CN
    5 THEN 139
MA 505 IF HYCUBE THEN IF PIE
    CE(SPX,SPY)=6 THEN 13
    9
FF 510 HYCUBE=CN1
MB 514 NX=INT(RND(CNØ)*CN9+C
    N1):NY=INT(RND(CNØ)*C
    N9+CN1)
BG 516 IF NX=CN5 AND NY=CN5
    THEN 510
FG 520 IF PIECE(NX,NY)<>CNØ
    THEN 510
FJ 525 FOR T=Ø TO 20:SOUND Ø
    ,30+T,T,(20-T)/2:NEXT
     T
KA 530 PIECE(NX,NY)=PIECE(PX
    ,PY):ORIENT(NX,NY)=OR
    IENT(PX,PY):LET CLR(N
    X,NY)=CLR(PX,PY)
JP 540 PIECE(PX,PY)=PIECE(SP
    X,SPY):ORIENT(PX,PY)=
    ORIENT(SPX,SPY):LET C
    LR(PX,PY)=CLR(SPX,SPY
    )
OE 550 PIECE(SPX,SPY)=CNØ:LE
    T CLR(SPX,SPY)=CNØ
MP 560 GOSUB 8200:TX=PX:TY=P
    Y:PX=SPX:PY=SPY:GOSUB
     8200:PX=TX:PY=TY
EC 565 FOR T=Ø TO 20:SOUND Ø
    ,45+(20-T),T,(20-T)/2
    :NEXT T
DD 570 GOSUB 8000:TX=PX:TY=P
    Y:PX=NX:PY=NY:GOSUB 8
    000:PX=TX:PY=TY:GOTO
    491
LD 600 K=Ø:IF PIECE(PX,PY)=C
    N4 THEN K=CLR(PX,PY)
GP 605 FOR T=10 TO 20:SOUND
    Ø,180,6,20-T:NEXT T
JN 610 PIECE(PX,PY)=PIECE(SP
    X,SPY):ORIENT(PX,PY)=
    ORIENT(SPX,SPY):LET C
    LR(PX,PY)=CLR(SPX,SPY
    )
DC 620 PIECE(SPX,SPY)=CNØ:LE
    T CLR(SPX,SPY)=CNØ
MN 630 GOSUB 8200:TX=PX:TY=P
    Y:PX=SPX:PY=SPY:GOSUB
     8200:PX=TX:PY=TY
OM 640 GOSUB 8000:IF K THEN
    7000
HC 641 GOTO 491
DL 650 IF HYSQ THEN 139
MH 651 PX=INT(RND(CNØ)*CN9+C
    N1):PY=INT(RND(CNØ)*C
    N9+CN1)
GJ 655 IF PIECE(PX,PY)<>CNØ
    THEN 651
PB 660 HYSQ=CN1:GOTO 455
IA 700 RESTORE PIECE(PX,PY)*
    10+9990
LI 702 FOR T=Ø TO 5:SOUND Ø,
    100+T*5,6,5-T:NEXT T
DH 705 IF STRIG((PLAYER-CN1)
    *NUMJOY=CN2)=CNØ THEN
     705
JL 710 HOLD=ORIENT(PX,PY)
EH 720 READ DUMMY2:TEMP=HOLD
JA 740 TEMP=TEMP+CN1:IF TEMP
    >DUMMY2 THEN TEMP=CN1
LF 760 GOSUB 8200:ORIENT(PX,
    PY)=TEMP:GOSUB 8000
EO 762 IF TEMP=HOLD THEN GOS
    UB 8700:GOTO 131
MI 765 IF STICK((PLAYER-CN1)
    *NUMJOY=CN2)<>15 THEN
     790
PF 766 IF STRIG((PLAYER-CN1)
    *NUMJOY=CN2)=CN1 THEN
     765
HD 780 GOTO 740
BC 790 MOVE=MOVE-CN1:IF MOVE
    =CNØ THEN MOVE=CN2:PL
    AYER=(PLAYER=CN1)+CN1
    :FIRED=CNØ:GOSUB 2800
PM 795 GOSUB 8700:GOTO 131
FF 800 IF FIRED THEN 131
FN 801 IF PIECE(PX,PY)<>CN2
    THEN 131
CM 802 SOUND CNØ,255,10,10:S
    OUND CN1,254,10,10
DH 805 FIRED=CN1:K=CNØ:DEAD=
    Ø:COLOR CN3:GOSUB 200
    Ø
BI 808 BEAMCK$(399)=CHR$(CNØ
    ):BEAMCK$(CN1,CN1)=CH
    R$(CNØ):BEAMCK$(CN2)=
    BEAMCK$
NA 810 FOR I=CN1 TO CN3:IF T
    ERM(I)<>CN1 THEN 830
OM 820 IF PIECE(LX(I),LY(I))
    >CNØ THEN TX=PX:TY=PY
    :PX=LX(I):PY=LY(I):GO
    SUB 9000:PX=TX:PY=TY
CD 830 NEXT I
AK 850 DEAD=1:GOSUB 2000:FOR
     I=CN1 TO CN3:IF TERM
    (I)<>CN1 THEN 870
FJ 860 IF PIECE(LX(I),LY(I))
    =CNØ THEN 870
LJ 861 TX=PX:TY=PY:PX=LX(I):
    PY=LY(I)
NO 862 IF PIECE(PX,PY)=CN4 T
    HEN K=CN1
LN 865 PIECE(PX,PY)=CNØ:LET
    CLR(PX,PY)=CNØ:GOSUB
    8200:PX=TX:PY=TY
CH 870 NEXT I
EH 875 BEAMCK$(399)=CHR$(Ø):
    BEAMCK$(1,1)=CHR$(Ø):
    BEAMCK$(2)=BEAMCK$
AC 876 SOUND CNØ,CNØ,CNØ,CNØ
    :SOUND CN1,CNØ,CNØ,CN
    Ø
HB 890 IF K THEN 7000
GJ 900 MOVE=MOVE-CN1:IF MOVE
    =CNØ THEN MOVE=CN2:PL
    AYER=(PLAYER=CN1)+CN1
    :GOSUB 2800
PB 910 GOSUB 8700:GOTO 131
CN 2000 LX(CN1)=PX:LY(CN1)=P
    Y:DIR(CN1)=ORIENT(PX
    ,PY)-CN1
CG 2002 FOR I=CN1 TO CN3:ALI
    VE(I)=CNØ:TERM(I)=CN
    Ø:NEXT I:ALIVE(CN1)=
    CN1
FI 2005 IF  NOT (ALIVE(CN1)=
    CN1 OR ALIVE(CN2)=CN
    1 OR ALIVE(CN3)=CN1)
     THEN 2100
CF 2010 FOR I=CN1 TO CN3:IF
    ALIVE(I)<CN1 THEN 20
    90
GH 2020 NLX(I)=LX(I)+DIRX(DI
    R(I)):NLY(I)=LY(I)+D
    IRY(DIR(I))
AL 2030 IF BEAMCK$(LX(I)+LY(
    I)*10+DIR(I)*100,LX(
    I)+LY(I)*10+DIR(I)*1
    00)=CHR$(CN1) THEN 2
    085
LJ 2035 BEAMCK$(LX(I)+LY(I)*
    10+DIR(I)*100,LX(I)+
    LY(I)*10+DIR(I)*100)
    =CHR$(CN1):GOTO 3000
JF 2080 TERM(I)=CN1:DRK(I)=T
    DIR:IF DEAD THEN 208
    5
CP 2081 TX=PX:TY=PY:PX=LX(I)
    :PY=LY(I):IF PIECE(P
    X,PY)=4 THEN K=K+CLR
    (PX,PY)
GL 2083 LET CLR(PX,PY)=3:GOS
    UB 8000:PX=TX:PY=TY
HF 2085 ALIVE(I)=-1
IN 2090 NEXT I:GOTO 2005
KD 2100 RETURN
HK 2800 COLOR PLAYER:PLOT 12
    ,CN5:DRAWTO 12,186:D
    RAWTO 122,186:DRAWTO
     122,CN5:DRAWTO 12,C
    N5
CG 2810 PLOT CNØ,CNØ:DRAWTO
    159,CNØ:DRAWTO 159,1
    91:DRAWTO CNØ,191:DR
    AWTO CNØ,CNØ:RETURN
BD 3000 X=LX(I)*12+7:Y=LY(I)
    *20-CN5
MA 3010 ON DIR(I) GOTO 3030,
    3040,3050
HC 3020 IF Y<20 THEN Y=20
KM 3021 POKE 1632,X:POKE 163
    3,Y-19:POKE 1634,Y:D
    UMMY=USR(1648):GOTO
    5000
KF 3030 POKE 1632,X:POKE 163
    3,X+11:POKE 1634,Y:D
    UMMY=USR(1679):GOTO
    5000
OE 3040 IF Y>171 THEN Y=171
AI 3041 POKE 1632,X+1:POKE 1
    633,Y:POKE 1634,Y+19
    :DUMMY=USR(1648):GOT
    O 5000
AF 3050 POKE 1632,X-11:POKE
    1633,X:POKE 1634,Y+1
    :DUMMY=USR(1679):GOT
    O 5000
MH 3060 GOTO 5000
HD 5000 IF NLX(I)>CN9 OR NLY
    (I)>CN9 OR NLX(I)<CN
    1 OR NLY(I)<CN1 THEN
     2085
EJ 5005 IF NLX(I)=CN5 AND NL
    Y(I)=CN5 THEN 2085
MN 5010 LX(I)=NLX(I):LY(I)=N
    LY(I):IF PIECE(NLX(I
    ),NLY(I))=CNØ THEN 2
    090
BF 5020 TDIR=DIR(I):DIR(I)=D
    IRCK(PIECE(LX(I),LY(
    I)),ORIENT(LX(I),LY(
    I))-1+DIR(I)*4)
GE 5030 IF DIR(I)=-1 THEN 20
    80
GI 5040 IF DIR(I)>-2 THEN 20
    90
NK 5050 J=CN3:IF ALIVE(CN2)=
    CNØ THEN J=CN2
MA 5060 ALIVE(J)=CN1:LX(J)=L
    X(I):LY(J)=LY(I)
PL 5070 DIR(I)=TDIR+CN1:IF D
    IR(I)=4 THEN DIR(I)=
    CNØ
CO 5071 DIR(J)=TDIR-CN1:IF D
    IR(J)=-1 THEN DIR(J)
    =CN3
MB 5080 GOTO 2090
JG 7000 FOR T=53248 TO 53255
    :POKE T,Ø:NEXT T:FOR
     T=10 TO 70 STEP 5:F
    OR TT=Ø TO 10-(T/7):
    SOUND Ø,T+20-TT,10,1
    Ø-TT
```

10 MHz IBM® XT Compatible Computer System Sale

Complete System for Only

$599.00

COMPUTER DIRECT

22292 N. Pepper Road, Barrington, Ill. 60010

"The Computer Experts"

Call (312) 382-5050 or 382-5244 for

Free Catalogs of Over 1000 Programs & Accesories

Best Service • Best Price • 1000 Programs • 500 Accessories • 15 Day Free Trial

BEST GUARANTEE

FREE CATALOGS

10 MHz Super Turbo IBM® XT Compatible Computer System

Twice the speed* at just a fraction of the cost!

* Computes over two times faster than the IBM® XT

Run 1000's of IBM® software programs available.

Look at all you get for only $599.00

15 Day Free Trial

90 Day Replacement Policy

Double 90 Day Policy On Computer

This computer system is sold as a complete unit. NO SUBSTITUTIONS are allowed.

The complete system	List Price	Sale Price
10 MHz Super Turbo XT Computer	$1295.00	$499.00
* 512K Memory	$99.95	No extra cost
* Single floppy disk drive	$129.95	No extra cost
* Parallel printer port	$59.95	No extra cost
* Serial printer port	$59.95	No extra cost
* Mouse/joystick port	$59.95	No extra cost
* RGB color graphics port	$99.95	No extra cost
* Hercules compatible monochrome port	$79.95	No extra cost
MS DOS 3.2 & GW Basic	$199.00	$99.00
12" Hi-Res 35 MHz Green Screen Monitor (TTL & EGA compatible)	$249.00	$99.00
Monitor interface cable	$24.95	$19.95
Big Blue Printer	$199.00	$39.95
RS 232 IBM to Big Blue cable	$19.95	$9.95
2 rolls of paper	$19.95	$5.95
Word First • Word Processor	$99.00	$39.95
Data First • Data Base	$99.00	$39.95
Calc First • Spreadsheet	$99.00	$39.95
Total price when bought separately	$2893.85	$892.85

Home & Business

This IBM® XT compatible is perfect for your home and/or business uses. It makes life easier in more ways than you can imagine. Use the system for personal letters, form letters, address storage, listing valuables, figuring finances, school reports, business reports, calculations, business projections...the list can go on and on. With the addition of some of the thousands of software programs available for IBM® you can increase the capabilities of your system even further. A terrific home improvement, business enhancer, entertainment center & educational aid!

Save over $275.00 off sale prices!

Complete System only $599

* Built-in the Super Turbo XT

IBM® is the trademark of International Business Machines Inc.

VISA · MASTERCARD · C.O.D. **Shipping, Handling & Insurance Charges and Information** VISA · MASTERCARD · C.O.D.

Add $35.00 for shipping, handling and insurance. Illinois residents please add 6¼% sales tax. Add $70.00 for CANADA, PUERTO RICO, HAWAII, ALASKA and APO-FPO. All orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA & PUERTO RICO. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice. Please call for C.O.D. charges.

CALL COMPUTER DIRECT AT

(312) 382-5050 or 382-5244

8am - 8pm CST Weekdays / 9am - 12 noon Saturdays

MasterCard VISA

Complete Apple® Compatible
Word Processing System

• 15 Day Free Trial • 90 Day Immediate Replacement Policy •

Home • Business • School • Word Processing

Look At All You Get for only $499.00

This computer system is sold as a complete unit, NO SUBSTITUTIONS are allowed.

		List Price	Sale Price
1	Laser 128 Apple Compatible Computer	$499.00	$395.00
2	Big Blue Printer	199.00	39.95
3	12'' 35 MHz Hi-Resolution Monitor	229.95	99.00
4	Magic Window IIe wordprocessor	150.00	49.95
5	Magicalc Spreadsheet	150.00	49.95
6	Magic Memory Data Base	60.00	39.95
7	Laser 128 Big Blue Printer Interface	19.95	12.95
8	2 Rolls of Heat Transfer Paper	19.95	5.95
	Comparable Apple System ($2495.00)	$1327.85	**$692.70**

Save Over $700.00 Off List Prices

All 8 Pieces Only $499

Magic Window IIe: Word processing is easy and more efficient because of a simple-to-read menu system and editing commands. Many powerful features of professional quality programs and more. Supports virtually all major printer functions. Operates with most 80 column video cards or 40/70 column no-hardware modes.

Magicalc: Rated as the best electronic spreadsheet on the market for Apple. The speed with which Magicalc solves number problems allows you to accurately analyze decisions beforehand. Supports multiple RAM cards of most manufacturers in any combinations up to a full 512 K.

Magic Memory: File anything, any way you want to: names, addresses, important numbers, dates and notes. Your information is organized simply, easy to maintain, and available instantly. Supports all popular printers and video cards.

Apple® is the registered trademark of Apple Computers Inc.

Shipping, Handling & Insurance Charges and Information

Add $35.00 for shipping, handling and insurance. Illinois residents please add 6¼% sales tax. Add $70.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, and APO-FPO. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA & PUERTO RICO. All orders must be in U.S. dollars. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice.

VISA - MASTERCARD - C.O.D. Please Call for C.O.D. Charges

COMPUTER DIRECT

22292 N. Pepper Road, Barrington Il. 60010

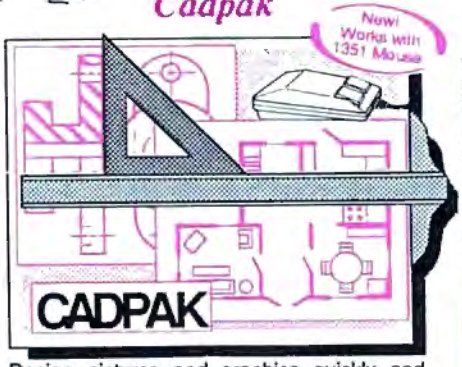
CALL (312) 382-5050 TO ORDER

We Love Our Customers!

```
GN 7001 SOUND 1,T+21-TT,10,1
        0:NEXT TT:NEXT T
LO 7002 FOR V=10 TO CN0 STEP
         -0.5:SOUND CN1,254,
        10,V:SOUND CN0,255,1
        0,V:POKE 710,V*CN4+1
        25:NEXT V
KM 7003 FOR T=1 TO 200:NEXT
        T:PRINT "PLAY AGAIN"
        ;:INPUT A$:IF A$="Y"
         THEN RUN
OO 7040 GRAPHICS CN0:END
KB 8000 REM [[DRAW A PIECE O
        N THE BOARD]]7030
IM 8010 PIECE=PIECE(PX,PY):L
        ET CLR=CLR(PX,PY)
MM 8020 IF PIECE=CN0 THEN RE
        TURN
HI 8030 ORIENT=ORIENT(PX,PY)
BI 8040 RESTORE PIECE*10+199
        90
MM 8050 BRDX=12*PX+CN2:BRDY=
        20*PY-14
BA 8060 READ DUMMY:COLOR CLR
MI 8070 FOR II=CN1 TO DUMMY:
        READ X1,Y1,X2,Y2
AO 8090 IF X1<CN0 THEN X1=-X
        1:COLOR CN3
IO 8100 IF ORIENT=CN1 THEN P
        LOT BRDX+X1,BRDY+Y1*
        CN2:DRAWTO BRDX+X2,B
        RDY+Y2*CN2:NEXT II:R
        ETURN
CA 8110 IF ORIENT=CN2 THEN P
        LOT BRDX+(CN9-Y1),BR
        DY+X1*CN2:DRAWTO BRD
        X+(CN9-Y2),BRDY+X2*C
        N2:NEXT II:RETURN
LC 8120 IF ORIENT=CN3 THEN P
        LOT BRDX+(CN9-X1),BR
        DY+(CN9-Y1)*CN2:DRAW
        TO BRDX+(CN9-X2),BRD
        Y+(CN9-Y2)*CN2:NEXT
        II:RETURN
GO 8130 PLOT BRDX+Y1,BRDY+(C
        N9-X1)*CN2:DRAWTO BR
        DX+Y2,BRDY+(CN9-X2)*
        CN2
JO 8140 NEXT II
KO 8150 RETURN
PJ 8200 REM [[ERASE BLOCK]]
ED 8210 BRDX=12*PX+CN1:BRDY=
        20*PY-14:COLOR CN0
BO 8220 FOR VX=CN1 TO 6:PLOT
         BRDX+VX,BRDY:DRAWTO
         BRDX+VX,BRDY+19
MJ 8221 PLOT BRDX+12-VX,BRDY
        :DRAWTO BRDX+12-VX,B
        RDY+19:NEXT VX:RETUR
        N
CC 8300 REM [[HIGHLIGHT BLOC
        K]]
JL 8310 BRDX=12*PX+CN1:BRDY=
        10*PY-7
MO 8320 S$(BRDY+528,BRDY+528
        )=CHR$(255):S$(BRDY+
        529,BRDY+539)=S$(BRD
        Y+528):POKE HPOS0,BR
        DX+47
PE 8325 POKE 53256,CN1
MH 8330 IF SEL=CN0 THEN POKE
         PCOLR0,88+48*(PLAYE
        R=CN1)
KP 8340 RETURN
MG 8400 REM [[UNHIGHLIGHT BL
        OCK]]
NO 8410 BRDY=10*PY-7
CP 8420 S$(BRDY+528,BRDY+528
        )=CHR$(CN0):S$(BRDY+
        529,BRDY+539)=S$(BRD
        Y+528)
KP 8430 RETURN
EN 8600 REM [[SELECT BLOCK]]
MK 8610 SEL=SEL+CN1:BRDX=12*
        PX+CN1:BRDY=10*PY-7
BC 8620 POKE 53252,BRDX+51:F
        OR T=BRDY+15 TO BRDY
        +26:POKE 384+MYBASE+
        T,PEEK(384+MYBASE+T)
        +3:NEXT T
CG 8625 POKE PCOLR0,80+48*(P
        LAYER=CN1)
LB 8630 RETURN
MH 8700 REM [[DESELECT BLOCK
        ]]
PN 8710 SEL=CN0:BRDY=10*SPY-
        7:FOR T=BRDY+15 TO B
        RDY+26:POKE 384+MYBA
        SE+T,PEEK(384+MYBASE
        +T)-CN3:NEXT T
EE 8720 POKE PCOLR0,88+48*(P
        LAYER=CN1):RETURN
MC 8900 REM [[TURN OFF SOUND
        ]]
MP 8910 FOR V=10 TO CN0 STEP
         -0.1:SOUND CN1,254,
        10,V:SOUND CN2,255,1
        0,V:POKE 710,V*CN4+1
        25:NEXT V:RETURN
BA 9000 CX=12*PX+17:CY=20*PY
        +4
EM 9020 FOR JJ=CN1 TO 13
JN 9030 N1=SI(JJ)+CX:N2=SI(J
        J+13)+CY
LI 9040 POKE 1632,N1:POKE 16
        33,N2:POKE 1634,N2+5
        :DUMMY=USR(1638)
AJ 9045 SOUND CN2,N1*10,8,13
        -JJ
KB 9050 NEXT JJ
KP 9060 RETURN
LO 9500 FOR JJ=1 TO 13
AH 9510 SI(JJ)=SIN(JJ)*JJ/2-
        10:SI(JJ+13)=COS(JJ)
        *JJ/2-10
LN 9520 NEXT JJ:RETURN
DB 10000 DATA 2,2,1,4,3,4,3,
         2,1
AA 10010 DATA 4,0,0,0,0,0,0,0,
         0,0,0,0,0,0,0,0,0,0
DD 10020 DATA 2,3,2,1,4,1,4,
         3,2
KP 10030 DATA 1,0,0,0,0
AN 10040 DATA 4,0,0,1,0,0,0,
         0,2,3,0,0,0,0,4,0,0
LL 10050 DATA 1,1,2,3,4
MB 10060 DATA 4,-1,3,0,3,4,-
         1,4,0,0,1,-1,1,2,0,
         2,-1
BK 10070 DATA 4,0,0,2,1,2,0,
         0,3,4,3,0,0,0,1,4,0
LP 20000 DATA 1,1,8,8,1
MB 20010 DATA 6,5,1,5,8,2,8,
         8,8,3,8,5,1
MH 20011 DATA 7,8,5,1,2,8,3,
         6,8,8,7,6
IG 20020 DATA 1,2,4.5,7,4.5
NI 20030 DATA 6,5,2,9,5,9,5,
         5,8,5,8,1,5
GJ 20031 DATA 1,5,5,2,5,2,5,
         8,1,5,9,5
FE 20040 DATA 14,1.5,2,1.5,8
         ,2,2,2,8,2.5,2
NA 20041 DATA 2.5,8,3,2,3,8,
         3.5,2,3.5,8
NA 20042 DATA 4,2,4,8,4.5,2,
         4.5,8,5,2,5
OA 20043 DATA 8,5.5,2,5.5,8,
         6,2,6,8,6.5
OB 20044 DATA 2,6.5,8,7,2,7,
         8,7.5,2,7.5
LM 20045 DATA 8,-1.5,2,7.5,2
NA 20050 DATA 4,2,2,7,2,7,2,
         7,7,7,7,2,7
GD 20051 DATA 2,7,2,2
MH 20060 DATA 8,2,2,7,2
KJ 20061 DATA 2,2.5,7,2.5,3,
         3,6,3
DC 20062 DATA 3,3.5,6,3.5,4,
         4,5,4,4,4.5,5
MO 20063 DATA 4.5,-2,2,5,5.5
         ,-8,2,5,5.5
CA 20070 DATA 15,2,7,2,8,2.5
         ,7
ML 20071 DATA 2.5,8,3,6,3,8,
         3.5,6,3.5,8
ML 20072 DATA 4,5,4,8,4.5,5,
         4.5,8,5,4,5
OG 20073 DATA 8,5.5,4,5.5,8,
         6,3,6,8,6.5
OF 20074 DATA 3,6.5,8,7,2,7,
         8,7.5,2,7.5
CI 20075 DATA 8,8,1,8,8,8.5,
         1,8.5,8
JM 20076 DATA -8,1,1,8
PL 21000 DATA 3,0,0,0,10,10,
         10
MA 21010 DATA 4,0,10,5,0,10,
         10,2,4
LP 21020 DATA 4,10,0,0,3,10,
         6,0,10
OA 21030 DATA 5,10,0,0,6,10,
         10,-10,6,0,6
KM 21040 DATA 5,0,10,0,0,10,
         3,2,5,10,10
AF 21050 DATA 3,10,0,0,5,10,
         10
PD 21060 DATA 6,0,0,0,0,10,-10
         ,0,10,10,-10,6,0,5
AC 21070 DATA 1,2,3,4,5,0,6,
         7,4,3,3
JH 21080 RESTORE 21070:FOR I
         =CN1 TO 11:READ A:A
         (I)=A:NEXT I:XX=126
         :YY=CN0:COLOR CN0
PF 21090 FOR I=CN1 TO 11:YY=
         YY+15:IF A(I)=CN0 T
         HEN NEXT I
KN 21092 FOR KK=CN0 TO CN1 S
         TEP 0.5
LK 21095 RESTORE A(I)*10+209
         90
DH 21100 READ DUMMY:FOR J=CN
         1 TO DUMMY:READ X,Y
         :FOR K=CN0 TO CN1
AH 21110 IF J=CN1 OR X<CN0 T
         HEN PLOT 0.6*ABS(X)
         +XX+KK+K,1.2*Y+YY:G
         OTO 21130
LP 21120 DRAWTO 0.6*X+XX+K+K
         K,1.2*Y+YY+KK/CN2
BF 21130 NEXT K:NEXT J:NEXT
         KK:NEXT I
MP 21140 FOR K=1 TO 1:FOR KK
         =51 TO 51
BB 21150 PLOT 135+K,130+KK:D
         RAWTO 138+K,130+KK:
         PLOT 136+K,130+KK:D
         RAWTO 136+K,135+KK
DJ 21160 PLOT 140+K,135+KK:D
         RAWTO 140+K,130+KK:
         DRAWTO 142+K,134+KK
         :DRAWTO 144+K,130+K
         K:DRAWTO 144+K,135+
         KK
JE 21170 NEXT KK:NEXT K
MM 21180 RETURN
LO 25000 DATA 68CE6106AD6006
         8D0006EE6106
EH 25010 DATA AD61068D010620
         AE06AD6106
MF 25020 DATA CD6206D0E66068
         CE6006EE6006
DM 25030 DATA AD60068D0006AD
         62068D0106
FN 25040 DATA 20AE06AD6006CD
         6106D0E660
LE 25045 DATA A9008D0406AD00
         064A4A8D0306
```

Hop to it! check out this Great Software

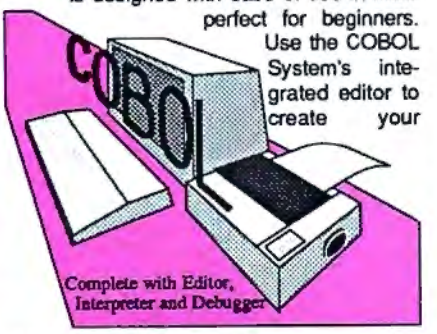
Cadpak



Design pictures and graphics quickly and precisely. Unlike other drawing programs, you can produce exact scaled output on your printer. Design in the units of your drawing problem (feet, miles, meters, etc.) and send hardcopy to most printers. Uses either the keyboard, lightpen or 1351 mouse. Two separate work screens—transfer artwork from one screen to the other. Place text in four sizes anywhere in the picture—three extra fonts included: Old English, 3-D and Tech. "Try Again" allows you to undo mistakes. Draw solid or dashed lines, circles, ellipses at any angle, rays and boxes. Design fill patterns, fonts and objects. ***Cadpak*** is the full-featured design and graphics package for your computer.
for C-64 **$39.95** for C-128 **$59.95**

COBOL

COBOL is the most widely used commercial programming language today. COBOL is a language that is common to many computers. Most computers equipped with a COBOL system can process any COBOL program with only minor revisions. Now you can learn the COBOL language using your Commodore. COBOL is easy to learn because of its English-like syntax. COBOL is designed with ease of use in mind perfect for beginners. Use the COBOL System's integrated editor to create your



COBOL source. Then the compiler checks your program's syntax and immediately converts it into an executable form. Includes sample programs and exercises to make learning COBOL even easier for the novice or experienced programmer.
for C-64 **$39.95** for C-128 **$39.95**

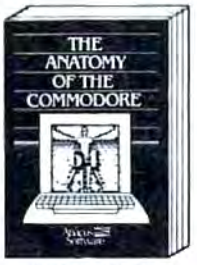
Super Pascal

Your complete system for developing applications in Pascal. A complete implementation of standard Pascal (Jensen and Wirth). C-64 version has a high-speed DOS (3X) for quick and efficient use. The extensive editor (source included) contains added features: append, search and replace. Includes assembler for any of your machine code requirements. Used in hundreds of schools to teach programming and also used for serious development projects. But it can be used for more than just learning Pascal, use it for serious programming. With complete graphic library (source included) in machine language for super-fast execution. Want to learn Pascal or develop software using the best tool? ***Super Pascal*** is your first choice.
for C-64 **$59.95** for C-128 **$59.95**

Super Pascal

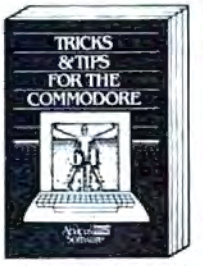
Learn the world's second most widely used language!

...and SUPER BOOKS!

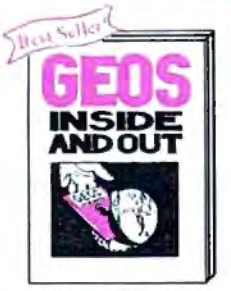


Anatomy of the C-64
Insider's guide to 64 internals. Graphics, sound, I/O, kernal, memory maps, and much more. Complete commented ROM listings. **300pp $19.95**

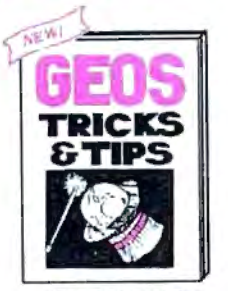
Anatomy of the 1541 Drive
Best handbook on this drive, explains all. Filled with many examples programs, utilities. Fully commented 1541 ROM listings. **500pp $19.95**



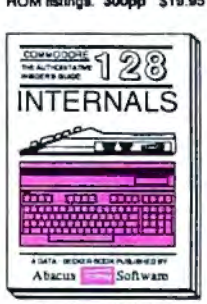
Tricks & Tips for the C-64
Collection of easy-to-use techniques: advanced graphics, improved data input, CP/M, enhanced BASIC, data handling and more. **275pp $19.95**



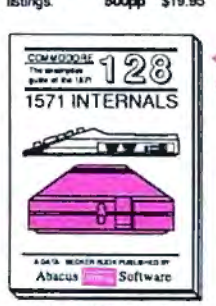
GEOS Inside and Out
Detailed info on GEOS. Add your own applications to GEOS. Edit icons. Constant display clock. Single-step through memory. **$19.95**



GEOS Tricks and Tips
Collection of helpful techniques for all GEOS users Includes font editor, machine language monitor, quick backup, more. **$19.95**

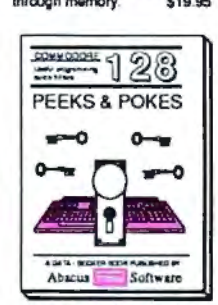


C-128 INTERNALS
Important C-128 information. Covers graphic chips, MMU, I/O, 80 column graphics and fully commented ROM listings, more. 500pp $19.95

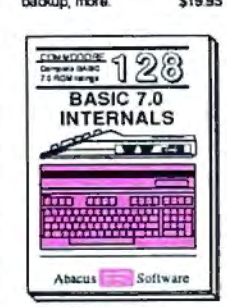


1571 INTERNALS
Essential reference. Internal drive functions. Explains various disk and file formats. Fully-commented ROM listings. **450pp $19.95**

C-128 TRICKS & TIPS
Fascinating and practical info on the C-128. 80-col hires graphics, bank switching. 300 pages of useful information for everyone. **$19.95**



C-128 PEEKS & POKES
Dozens of programming quick-hitters. techniques on the operating system, stacks, zero page, pointers, and BASIC. **240pp $16.95**



C-128 BASIC 7.0 Internals
Get all the inside info on BASIC 7.0. This exhaustive handbook is complete with fully commented BASIC 7.0 ROM listings. **$24.95**

Call now for the name of your nearest dealer. Or order direct with your credit card by calling 616/241-5510. Add $4.00 per order for S&H. Foreign add $12.00 per item.

Other books and software also available. Call or write for your **free catalog.** Dealers inquires welcome–2000 nationwide.

Abacus
P.O. Box 7219
Dept. C6
Grand Rapids, MI 49510
Telex 709-101 • Fax 616/241-5021
Phone 616/241-5510

```
CN 25050 DATA A02718AD01066D
         03068D0306
ND 25060 DATA AD04066900 8D04
         068B10EB

IG 20020 DATA 1,2,4.5,7,4.5
NI 20030 DATA 6,5,2,9,5,9,5,
         5,8,5,8,1,5
GJ 20031 DATA 1,5,5,2,5,2,5,
         8,1,5,9,5
FE 20040 DATA 14,1.5,2,1.5,8
         ,2,2,2,8,2.5,2
NA 20041 DATA 2.5,8,3,2,3,8,
         3.5,2,3.5,8
NA 20042 DATA 4,2,4,8,4.5,2,
         4.5,8,5,2,5
OA 20043 DATA 8,5.5,2,5.5,8,
         6,2,6,8,6.5
OB 20044 DATA 2,6.5,8,7,2,7,
         8,7.5,2,7.5
LM 20045 DATA 8,-1.5,2,7.5,2
NA 20050 DATA 4,2,2,7,2,7,2,
         7,7,7,7,2,7
GD 20051 DATA 2,7,2,2
MH 20060 DATA 8,2,2,7,2
KJ 20061 DATA 2,2.5,7,2.5,3,
         3,6,3
DC 20062 DATA 3,3.5,6,3.5,4,
         4,5,4,4,4.5,5
MD 20063 DATA 4.5,-2,2,5,5.5
         ,-8,2,5,5.5
CA 20070 DATA 15,2,7,2,8,2.5
         ,7
NL 20071 DATA 2.5,8,3,6,3,8,
         3.5,6,3.5,8
NL 20072 DATA 4,5,4,8,4.5,5,
         4.5,8,5,4,5
DG 20073 DATA 8,5.5,4,5.5,8,
         6,3,6,8,6.5
OF 20074 DATA 3,6.5,8,7,2,7,
         8,7.5,2,7.5
CI 20075 DATA 8,8,1,8,8,8.5,
         1,8.5,8
JM 20076 DATA -8,1,1,8
PL 21000 DATA 3,0,0,0,10,10,
         10
MA 21010 DATA 4,0,10,5,0,10,
         10,2,4
LP 21020 DATA 4,10,0,0,3,10,
         6,0,10
DA 21030 DATA 5,10,0,0,6,10,
         10,-10,6,0,6
KM 21040 DATA 5,0,10,0,0,10,
         3,2,5,10,10
AF 21050 DATA 3,10,0,0,5,10,
         10
PD 21060 DATA 6,0,0,0,10,-10
         ,0,10,10,-10,6,0,5
GM 25070 DATA A558186D030685
         D0A559
CJ 25080 DATA 6D0406 85D1A000
         AD00062903
MJ 25090 DATA AABDF10651D091
         D060C0300C03
JI 25100 DATA ENDOFDATA
DL 26000 RESTORE 25000:MEM=1
         647
GD 26010 READ ML$:IF ML$="EN
         DOFDATA" THEN RETUR
         N
LI 26020 FOR I=CN1 TO LEN(ML
         $) STEP CN2
DD 26030 MEM=MEM+CN1
PG 26040 H=ASC(ML$(I,I))-48:
         IF H>9 THEN H=H-7
DB 26050 L=ASC(ML$(I+CN1,I+C
         N1))-48:IF L>9 THEN
          L=L-7
LG 26060 POKE MEM,H*16+L:NEX
         T I:GOTO 26010
```

Program 4: Apple II *Laser Chess*

Version by Tim Victor, Editorial Programmer

```
BD 100 GOTO 510
99 110 GOSUB 130: IF I = SP THEN
        330
15 120 RETURN
3E 130 Y = Y(I):X = X(I): IF BD(
       I) = - 1 THEN 200
9E 140 GOSUB 180: IF X < > 2 * I
       NT (X / 2) THEN C = 1 - C
17 150 C = C + 2 * (I - 2 * INT
       (I / 2))
59 160 CALL 25306,SN(P) + D,C,X,
       Y
1F 170 RETURN
AJ 180 C = C(BD(I)):P = P(BD(I))
       :D = D(BD(I)): RETURN
23 190 RETURN
E2 200 CALL 25309,X,Y
77 210 IF X = 4 AND Y = 4 THEN H
       COLOR= 6: HPLOT 146,74 TO
        146,91
16 220 RETURN
#D 230 HCOLOR= 4: IF I < > 2 * I
       NT (I / 2) THEN HCOLOR= 7
46 240 LX = 42 + 21 * X(I):LY =
       2 + 18 * Y(I): ON D + 1 G
       OTO 250,260,270,280
E# 250 HPLOT LX + 9,LY + 8 TO LX
        + 9,LY + 17 TO LX + 10,L
       Y + 17 TO LX + 10,LY + 8:
        RETURN
5F 260 HPLOT LX,LY + 8 TO LX + 1
       0,LY + 8 TO LX + 10,LY +
       9 TO LX,LY + 9: RETURN
A8 270 HPLOT LX + 9,LY TO LX + 9
       ,LY + 9 TO LX + 10,LY + 9
       TO LX + 10,LY: RETURN
11 280 HPLOT LX + 9,LY + 8 TO LX
        + 20,LY + 8 TO LX + 20,L
       Y + 9 TO LX + 9,LY + 9: R
       ETURN
A# 290 HCOLOR= 7: IF CP = 2 * IN
       T (CP / 2) THEN HCOLOR= 4
7F 300 PX = 43 + X(CP) * 21:PY =
        3 + Y(CP) * 18: HPLOT PX
       ,PY TO PX + 18,PY TO PX +
        18,PY + 15 TO PX,PY + 15
        TO PX,PY
B# 310 IF CP = SP THEN GOSUB 330
17 320 RETURN
AD 330 HC = 7:X = X(SP) * 21:Y =
        Y(SP) * 18: IF SP = 2 *
       INT (SP / 2) THEN HC = 4
CB 340 IF SP = CP THEN HC = 11 -
        HC
59 350 HCOLOR= HC: FOR J = # TO
       1: FOR K = # TO 1:PX = 43
        + X + J * 18:PY = 2 + Y
       + K * 16: HPLOT PX,PY TO
       PX,PY + 1: NEXT : NEXT :
       RETURN
D# 360 GOSUB 130: CALL 25312,SN(
       P) + D,C,X,Y
21 370 RETURN
42 380 R = INT ( RND (1) * 81):
       IF BD(R) < > - 1 OR R = 4
       # THEN 380
#5 390 IF I = L1 THEN L1 = R
35 400 IF I = L2 THEN L2 = R
6E 410 BD(R) = BD(I):BD(I) = - 1
       :I = R: GOSUB 110
18 420 RETURN
E5 430 VTAB 7: HTAB 2: NORMAL :
       IF MS = 1 THEN INVERSE
A# 440 PRINT "FIRE": NORMAL : IF
        MS = 2 THEN INVERSE
D5 450 VTAB 9: HTAB 2: PRINT "PA
       SS": NORMAL : IF MS = 3 T
       HEN INVERSE
#7 460 VTAB 11: HTAB 2: PRINT "Q
       UIT": NORMAL : RETURN
A9 470 PX = 44 + 21 * X(I):PY =
       4 + 18 * Y(I): HCOLOR= 7:
        IF I < > 2 * INT (I / 2)
        THEN HCOLOR= 4
F1 480 R = 6: FOR K = 1 TO 18: H
       PLOT PX,PY + R TO PX + 15
       ,PY + R:R = R + 7 - 16 *
       (R > 8): NEXT : RETURN
49 490 VTAB 24: HTAB 14: PRINT "
       PRESS A KEY";
6# 500 GET A$: HTAB 14: PRINT SP
       C( 11);: RETURN
#A 510 LOMEM: 32768: PRINT CHR$
       (21): DIM BD(#0),TD(7,3),
       X(#0),Y(#0),P(63),C(63),D
       (63)
F6 520 TEXT : HOME : VTAB 4: HTA
       B 12: PRINT "LASER CHESS(
       TM)": VTAB 7: HTAB 12: PR
       INT "COPYRIGHT 1987": HTA
       B 7: PRINT "COMPUTE! PUBL
       ICATIONS, INC"
E4 530 VTAB 10: HTAB 10: PRINT "
       ALL RIGHTS RESERVED"
FD 540 PRINT CHR$ (4);"BLOAD LAS
       ER.ML"
91 550 CALL 25315: FOR I = 24400
        TO I + 7: POKE I,128: NE
       XT
E9 560 POKE 6,#0: POKE 7,95: IF
       PEEK (191 * 256) = 76 THE
       N PRINT CHR$ (4)"pr#a$600
       #": GOTO 580
FE 570 POKE 54,#: POKE 55,96: CA
       LL 1002
F2 580 FOR I = # TO #0:Y(I) = IN
       T (I / 9):X(I) = I - Y(I)
        * 9: NEXT
E9 590 FOR I = # TO 63:C(I) = IN
       T (I / 32):P(I) = INT ((I
       - 32 * C(I)) / 4):D(I) =
       I - 32 * C(I) - 4 * P(I)
       : NEXT
2A 600 POKE 28,128: POKE 230,32:
        CALL 62454: POKE 49239,#
       : POKE 49232,#: POKE 4923
       4,#
E6 610 RESTORE : FOR I = # TO 8:
        READ SN(I): NEXT
BB 620 FOR I = # TO 17: READ BD(
       I): NEXT
D# 630 FOR I = 18 TO 62:BD(I) =
       - 1: NEXT
#A 640 FOR I = 63 TO #0: READ BD
       (I): NEXT

CA 650 FOR P = # TO 7: FOR D = #
       TO 3: READ TD(P,D): NEXT
       : NEXT
71 660 CP = #:SP = - 1:PP = 1:MC
       = 2:L1 = 3:L2 = 77
#D 670 FOR I = # TO #0: GOSUB 11
       #: NEXT
CB 680 QF = #:MS = #:K1 = #:K2 =
        #: GOSUB 430
64 690 IF QF THEN 600
25 700 IF K1 + K2 = # THEN 750
7# 710 VTAB 23: IF K1 * K2 THEN
       HTAB 18: PRINT "DRAW": GO
       TO 740
22 720 IF K1 THEN HTAB 14: PRINT
        "ORANGE WINS": GOTO 740
12 730 HTAB 15: PRINT "BLUE WINS
       "
9F 740 GOSUB 490: GOTO 600
7C 750 IF MC = 2 THEN PP = 1 - P
       P:MC = #:MF = #: VTAB 20:
        HTAB 2: PRINT MID$ ("] ]
       [ [",PP * 3 + 1,3): IF SP
        < > - 1 THEN I = SP:SP =
        - 1: GOSUB 110
```

```
5F 770 GOSUB 290
F4 780 GET C$:C = ASC (C$):K = (
      C = 11) + 2 * (C = 21) +
      3 * (C = 10) + 4 * (C = 8
      ) + 5 * (C = 13) + 6 * (C
       = 44 OR C = 60) + 7 * (C
       = 46 OR C = 62) + 8 * (C
       = 27): IF C = 0 THEN 780
21 790 OP = CP: ON K GOSUB 800,8
      20,840,860,880,1130,1170,
      1210:I = OP: GOSUB 110: G
      OTO 690
B9 800 IF CP > 8 THEN CP = CP -
      9
1A 810 RETURN
48 820 IF X(CP) < 8 THEN CP = CP
       + 1
1E 830 RETURN
72 840 IF CP < 72 THEN CP = CP +
      9
22 850 RETURN
58 860 IF X(CP) > 0 THEN CP = CP
       - 1
26 870 RETURN
26 880 IF SP < > - 1 THEN 920
2D 890 IF BD(CP) = - 1 THEN RETU
      RN
94 900 I = CP: GOSUB 180: IF PP
      = C THEN SC = C:SI = P:SD
       = D:SP = CP
1B 910 RETURN
58 920 YM = Y(CP) - Y(SP):XM = X
      (CP) - X(SP):I = SP: GOSU
      B 180:DM = (D < > SD): IF
       ABS (XM) + ABS (YM) + DM
       + MC > 2 THEN RETURN
C8 930 IF CP = SP THEN 1120
9B 940 IF ABS (XM) = 2 AND (BD(S
      P + XM / 2) < > - 1 OR SP
       + XM / 2 = 40) THEN RETU
      RN
5F 950 IF ABS (YM) = 2 AND (BD(S
      P + YM / 2 * 9) < > - 1 O
      R SP + YM / 2 * 9 = 40) T
      HEN RETURN
D8 960 IF YM AND XM AND (BD(SP +
       YM * 9) < > - 1 OR SP +
      YM * 9 = 40) AND (BD(SP +
       XM) < > - 1 OR SP + XM =
       40) THEN RETURN
E9 970 IF CP < > 40 THEN 1000
E9 980 IF MF < > 2 THEN MF = 2:I
       = SP: GOSUB 380: GOTO 11
      20
2B 990 RETURN
9B 1000 IF BD(CP) = - 1 THEN 109
      0
4E 1010 IF (SI < > 0 AND SI < >
      6) OR MF = 3 THEN 1070
6A 1020 IF BD(CP) = 0 THEN K1 =
      1
57 1030 IF BD(CP) = 32 THEN K2 =
      1
CA 1040 IF CP = L0 THEN L0 = - 1
D3 1050 IF CP = L1 THEN L1 = - 1
8F 1060 MF = 3: GOTO 1090
32 1070 IF SI = 2 AND MF < > 1 T
      HEN MF = 1:I = CP: GOSUB
       380: GOTO 1090
F1 1080 RETURN
7E 1090 BD(CP) = BD(SP):BD(SP) =
       - 1
17 1100 IF SP = L1 THEN L1 = CP
2F 1110 IF SP = L2 THEN L2 = CP
D1 1120 I = SP:SP = - 1: GOSUB 1
      10:MC = MC + ABS (XM) +
      ABS (YM) + DM: RETURN
6F 1130 IF SP = - 1 THEN RETURN
3D 1140 I = SP: GOSUB 180:D = D
      - 1: IF D < 0 THEN D = S
      N(SI + 1) - SN(SI) - 1
29 1150 BD(SP) = SC * 32 + P * 4
      + D: IF I < > CP THEN G
      OSUB 110
EB 1160 RETURN
7F 1170 IF SP = - 1 THEN RETURN
C0 1180 I = SP: GOSUB 180:D = D
      + 1: IF D = SN(SI + 1) -
       SN(SI) THEN D = 0
39 1190 BD(SP) = SC * 32 + P * 4
      + D: IF I < > CP THEN G
      OSUB 110
D5 1200 RETURN
03 1210 MS = 1
12 1220 GOSUB 430: GET C$: IF C$
      = CHR$ (21) OR C$ = CHR
      $ (10) THEN MS = MS + 1
      - 3 * (MS = 3)
43 1230 IF C$ = CHR$ (8) OR C$ =
       CHR$ (11) THEN MS = MS
      - 1 + 3 * (MS = 1)
10 1240 IF C$ = CHR$ (27) THEN 1
      270
31 1250 IF C$ < > CHR$ (13) THEN
       1220
8D 1260 ON MS GOSUB 1280,1590,16
      10
43 1270 MS = 0: GOSUB 430: RETUR
      N
34 1280 I = (PP = 0) * L1 + (PP
      = 1) * L2: IF I = - 1 OR
       MF = 4 THEN RETURN
1F 1290 MF = 4:MC = MC + 1:Z = F
      RE (0)
AC 1300 LD = BD(I) - 32 * PP - 4
      :LB$ = CHR$ (I):LX$ = ""
      :EX$ = "":SX$ = ""
8B 1310 FOR Z = 0 TO 1 STEP 0: O
      N LD + 1 GOTO 1320,1340,
      1360,1380
0A 1320 D = 2:J = I - 9: IF I >
      8 AND I < > 49 THEN 1400
DD 1330 GOSUB 230: GOTO 1500
6F 1340 D = 3:J = I + 1: IF X(I)
      < > 8 AND I < > 39 THEN
      1400
E5 1350 GOSUB 230: GOTO 1500
97 1360 D = 0:J = I + 9: IF I <
      72 AND I < > 31 THEN 140
      0
ED 1370 GOSUB 230: GOTO 1500
5A 1380 D = 1:J = I - 1: IF X(I)
      AND I < > 41 THEN 1400
F5 1390 GOSUB 230: GOTO 1500
EE 1400 GOSUB 230:D = LD:I = J:
      GOSUB 230
69 1410 I$ = CHR$ (I):LB$ = LB$
      + I$: IF BD(I) = - 1 THE
      N NEXT
BC 1420 GOSUB 180:LT = LD - D:LT
      = TD(P,LT + 4 * (LT < 0
      )): IF LT = - 1 THEN EX$
       = EX$ + I$: GOSUB 360:D
       = LD: GOSUB 230: GOTO 1
      500
E7 1430 IF LT = 4 THEN LX$ = LX$
      + I$ + CHR$ (D - 1 + 4
      * (D = 0)):LT = 1
F9 1440 IF P < > 3 OR LT THEN 14
      90
D0 1450 IF SX$ = "" THEN 1480
3D 1460 FOR J = 1 TO LEN (SX$):
      IF I$ = MID$ (SX$,J,1) T
      HEN J = 999
61 1470 NEXT : IF J = 1000 THEN
      1500
8A 1480 SX$ = SX$ + I$
7A 1490 LD = D + LT:LD = LD - 4
      * (LD > 3): NEXT
13 1500 IF LX$ < > "" THEN I = A
      SC (LX$):LD = ASC ( MID$
       (LX$,2)):LX$ = MID$ (LX
      $,3): NEXT
2D 1510 Z = 1: NEXT : GOSUB 490:
       IF EX$ = "" THEN 1570
A5 1520 FOR J = 1 TO LEN (EX$):I
       = ASC ( MID$ (EX$,J)):
      GOSUB 470: IF BD(I) = 0
      THEN K1 = 1
3B 1530 IF BD(I) = 32 THEN K2 =
      1
28 1540 IF I = L1 THEN L1 = - 1
AE 1550 IF I = L2 THEN L2 = - 1
F2 1560 BD(I) = - 1: NEXT
88 1570 FOR B = 1 TO LEN (LB$):I
       = ASC ( MID$ (LB$,B)):
      GOSUB 110: NEXT
FB 1580 RETURN
CE 1590 IF SP < > - 1 THEN I = S
      P:SP = - 1: GOSUB 110
0F 1600 MC = 2: RETURN
D0 1610 VTAB 24: HTAB 8: INVERSE
       : PRINT "Q";: NORMAL :
      PRINT "UIT ";
FE 1620 INVERSE : PRINT "R";: NO
      RMAL : PRINT "ESTART OR
      ";
A4 1630 INVERSE : PRINT "C";: NO
      RMAL : PRINT "ANCEL";: G
      ET C$: IF C$ = "R" OR C$
       = "r" THEN QF = 1: RETU
      RN
A2 1640 IF C$ < > "Q" AND C$ < >
       "q" THEN HTAB 8: PRINT
      SPC( 23);: RETURN
6D 1650 TEXT : HOME : END
F9 1660 DATA 0,1,5,6,10,12,14,18
      ,22
9B 1670 DATA 29,29,16,6,0,8,17,3
      0,30
65 1680 DATA 30,26,26,12,20,21,2
      6,26,29
2C 1690 DATA 63,56,56,53,52,46,5
      6,56,60
AF 1700 DATA 60,60,49,40,32,36,4
      8,63,63
FC 1710 DATA -1,-1,-1,-1
01 1720 DATA -1,-1,-1,-1
F3 1730 DATA 0,1,2,3
42 1740 DATA -1,0,4,0
32 1750 DATA 3,2,1,0
E1 1760 DATA 2,1,0,3
95 1770 DATA -1,-1,0,-1
79 1780 DATA -1,-1,1,0
```

Program 5: LASER.ML

Please refer to the "Apple MLX" article elsewhere in this issue before entering the following program.

```
6000: D8 78 85 45 86 46 84 47 ED
6008: A6 07 0A 0A B0 04 10 3E B3
6010: 30 04 10 01 E8 E8 0A 86 B1
6018: 1B 18 65 06 85 1A 90 02 31
6020: E6 1B A5 28 85 08 A5 29 13
6028: 29 03 05 E6 85 09 A2 08 EA
6030: A0 00 B1 1A 24 32 30 02 65
6038: 49 7F A4 24 91 08 E6 1A E8
6040: D0 02 E6 1B A5 09 18 69 63
6048: 04 85 09 CA D0 E2 A5 45 DC
6050: A6 46 A4 47 58 4C F0 FD D2
6058: 80 FC E6 E6 FE E6 E6 80 C5
6060: 80 BE E6 E6 BE E6 FE 80 6C
6068: 80 BC E6 86 86 E6 BE 80 AB
6070: 80 BE E6 E6 E6 E6 BE 80 3D
6078: 80 FE 86 86 BE 86 FE 80 01
6080: 80 FE 86 86 BE 86 86 80 18
6088: 80 BC E6 86 F6 E6 BE 80 4F
6090: 80 E6 E6 E6 FE E6 E6 80 78
6098: 80 98 98 98 98 98 98 80 35
60A0: 80 E0 E0 E0 E0 E6 BC 80 A0
60A8: 80 E6 E6 B6 9E E6 E6 80 8A
60B0: 80 86 86 86 86 86 FE 80 59
60B8: 80 E6 FE E6 E6 E6 E6 80 E2
60C0: 80 BE E6 E6 E6 E6 E6 80 DD
60C8: 80 BC E6 E6 E6 E6 BC 80 11
60D0: 80 BE E6 E6 BE 86 86 80 6A
60D8: 80 BC E6 E6 E6 B6 EC 80 C0
60E0: 80 BE E6 E6 BE E6 E6 80 BC
```

```
60E8: 80 BC E6 8C B0 E6 BE 80 DD
60F0: 80 FE 98 98 98 98 98 80 27
60F8: 80 E6 E6 E6 E6 E6 BE 80 CF
6100: 80 E6 E6 E6 E6 E6 98 80 8C
6108: 80 E6 E6 E6 E6 FE E6 80 91
6110: 80 E6 E6 E6 BC E6 E6 80 E7
6118: 80 E6 E6 E6 BC 98 98 80 1A
6120: 80 FE B0 98 8C B6 FE 80 7F
6128: 95 95 95 95 95 95 95 95 EA
6130: 00 00 00 00 00 00 00 00 F2
6138: AA AA AA AA AA AA AA AA FA
6140: 05 D3 BD DE BF 5F 9D EE C5
6148: EC BF 5F 82 94 BA 52 97 51
6150: 4A 52 94 A5 72 A9 54 01 9D
6158: DC EE 56 BA 9D D2 EA 56 20
6160: EE 77 00 AA 55 2B 94 A5 49
6168: 29 4B A5 29 74 A5 00 5F 73
6170: 2F 9C FF 75 EF F7 2F 97 9E
6178: C0 2E 5A D6 B5 AD 6B 5A 75
6180: D6 B5 C0 0A 57 2B 97 AA 16
6188: 95 FA AE 00 75 CE 9D 6B 38
6190: FD AF 5E D7 FB AD D3 AE 74
6198: 00 55 CA FD 54 BD 5C AE EA
61A0: 50 01 7D 2A D7 72 EF 57 DD
61A8: AF 5E AF 5D EB B9 56 BE 58
61B0: 80 AD 5B A7 4A F5 7B 6A 90
61B8: B5 5B F7 EE 70 39 CF DF 03
61C0: AA D5 6D AB D5 EE 9D 2B 87
61C8: 56 00 2E 5C 00 36 DB 6D 40
61D0: B6 DB 60 B9 72 A7 A5 4F 5A
61D8: 4A 9E 95 3D 02 76 BB 4E 9E
61E0: D7 69 DA ED 3B 5D A7 60 23
61E8: 95 3D 2A 7A 54 F4 A9 EB 68
61F0: 97 03 E4 EA 7C 9D 4F 93 18
61F8: A9 F2 75 3E 40 AD 5B A7 F6
6200: 56 EA 9D 6B F9 7D AF 5D 97
6208: 6A 95 7A 76 0E D2 AF 54 8D
6210: BA D7 AF B5 FC EB 75 4E 4A
6218: AD D2 B5 60 F9 2B B7 BE CF
6220: D5 F2 AA 7F 2B A5 56 FA 71
6228: 7E E7 03 9C FD F4 AA D5 3F
6230: D3 F9 55 2B E7 DB D5 DA EF
6238: 7C 00 35 FF D7 FF FF F7 98
6240: FF DF F7 D7 D7 DF DF F7 6F
6248: D7 FF DF FF FF FF FF DF D4
6250: FF D7 D7 DF F7 F7 FF FF A3
6258: FF DF FF FF D7 DF FF DF 33
6260: FF D7 D7 D7 FF FF FF F7 8B
6268: D7 DF F7 D7 DF F7 D7 DF FB
6270: FF EB EB FB FF FF AE BA 85
6278: FF AA AE FE BA AE BF AA 68
6280: AF EA FA EE AE FF AA BA AB
6288: FA FF EE BF EA EB FB AB 4F
6290: AF AE AE AF BA BA BB EF D1
6298: AB AB BA BF EA EB EB FA 4B
62A0: FE FF AE BA BB BF B5 BF 6E
62A8: BF B7 B5 B7 BD BF B5 BF 85
62B0: BF B5 B7 BD BF BF BF BF D1
62B8: BF B5 B7 BD BD BF BF BD C7
62C0: B5 B5 B5 B5 B7 B7 BD BD B5
62C8: BD BD BD BD BD BF BF BF 9B
62D0: BF BF BF BF BF BF BF B5 8B
62D8: B7 BF 4C E4 63 4C 14 64 1A
62E0: 4C 5C 64 A9 00 85 E3 18 FF
62E8: A9 3B 6D 3A 62 8D B6 63 BC
62F0: 90 07 EE B7 63 EE BD 63 CE
62F8: 18 6D 3A 62 8D BC 63 90 49
6300: 03 EE BD 63 A9 16 85 F9 9C
6308: A9 7C 85 EC A9 65 85 ED 1E
6310: A5 EC 85 EE 48 A5 ED 85 BE
6318: EF 48 20 9A 63 20 31 63 F7
6320: 20 7E 63 68 85 EF 68 85 CB
6328: EE 20 7E 63 C6 F9 D0 E0 15
6330: 60 A9 12 85 EB 0E 7D 63 22
6338: 2E 7C 63 2E 7B 63 B0 1F 6E
6340: A0 FF A2 80 C8 8A 0A B1 E9
6348: EE 2A AA 09 80 91 EC C0 DB
6350: 02 D0 F1 29 BF 91 EC 20 5B
6358: 56 65 C6 EB D0 E2 60 A0 AE
6360: 02 B1 EE 09 40 18 2C B1 6F
6368: EE B0 02 69 80 6A 91 EC 67
6370: 88 10 F4 20 56 65 C6 EB E1
6378: D0 E5 60 2C 3F FC A9 12 43
6380: 85 EB A0 02 B1 EE 49 40 55
6388: 2C B1 EE 49 7F 91 EC 88 E8
6390: 10 F7 20 56 65 C6 EB D0 B5
6398: E9 60 A9 12 85 EB A9 01 F2
63A0: 20 CC 63 90 06 20 CC 63 CD
63A8: 2A D0 F5 AA CA A0 00 BD B8
63B0: 3B 62 91 EC C8 BD 3B 62 C4
63B8: 91 EC C8 BD 3B 62 91 EC EB
63C0: A9 03 A2 EC 20 61 65 C6 58
63C8: EB D0 D3 60 C6 E3 10 11 31
63D0: A2 07 86 E3 AE 40 61 86 79
63D8: 19 EE D5 63 D0 03 EE D6 20
63E0: 63 06 19 60 20 1C 65 8D CD
63E8: 76 65 20 1C 65 8D 77 65 BF
63F0: 20 9E 64 20 D8 64 A2 12 AD
63F8: 8E 79 65 A0 02 B1 FC 91 7E
6400: FE 88 10 F9 A9 03 A2 FC A7
6408: 20 61 65 20 27 65 CE 79 CD
6410: 65 D0 EB 60 20 9E 64 AD D4
6418: 7B 65 C9 04 D0 09 CD 7A 32
6420: 65 D0 04 A9 AA D0 0A 4D E4
6428: 7A 65 4A A9 80 B0 02 A9 DF
6430: FF 8D 3C 64 A2 12 8E 79 1E
6438: 65 A0 FF A9 00 C8 18 2C F5
6440: 5B 64 F0 01 38 2A 09 80 FA
6448: 91 FE C0 02 D0 EF 29 BF 2A
6450: 91 FE 20 27 65 CE 79 65 D6
6458: D0 DF 60 08 20 1C 65 8D D7
6460: 76 65 20 1C 65 49 01 8D 63
6468: 77 65 20 9E 64 20 D8 64 ED
6470: A2 11 A9 02 2C 77 65 D0 FE
6478: 02 A2 31 8E 87 64 A2 12 1F
6480: 8E 79 65 A0 02 B1 FC 11 87
6488: FE 91 FE 88 10 F7 A9 03 54
6490: A2 FC 20 61 65 20 27 65 63
6498: CE 79 65 D0 E6 60 20 1C F5
64A0: 65 8D 7A 65 A0 03 20 FA 71
64A8: 64 18 69 06 48 20 1C 65 97
64B0: 8D 7B 65 A8 B9 C6 64 85 8D
64B8: FE B9 CF 64 85 FF 68 A0 4D
64C0: 00 A2 FE 4C 6B 65 00 00 C7
64C8: 00 80 A8 A8 A8 50 50 28 A0
64D0: 31 3A 23 28 31 3A 20 29 83
64D8: A9 7C 85 FC A9 65 85 FD 02
64E0: AD 76 65 A0 D8 20 FA 64 76
64E8: A2 FC 20 6B 65 AD 77 65 33
64F0: A0 36 20 FA 64 A2 FC 4C 3F
64F8: 6B 65 A2 08 85 19 84 E3 23
6500: A9 00 85 F9 CA 30 12 0A 35
6508: 26 F9 06 19 90 F6 18 65 AC
6510: E3 90 F1 E6 F9 D0 ED A9 36
6518: 00 A4 F9 60 20 B1 00 20 39
6520: 05 E1 A5 A1 A4 A0 60 18 35
6528: A5 FF 69 04 85 FF 2C 55 0D
6530: 65 F0 01 60 E9 1F 85 FF E6
6538: A5 FE 69 7F 85 FE B0 01 45
6540: 60 E6 FF A5 FF 29 03 F0 EA
6548: 01 60 A5 E6 85 FF A5 FE 45
6550: 69 27 85 FE 60 1C A9 03 04
6558: A2 EE 20 61 65 A9 03 A2 C4
6560: EC 18 75 00 95 00 90 02 26
6568: F6 01 60 18 75 00 95 00 53
6570: 98 75 01 95 01 60 31 20 6A
```

©

BE A WINNER IN OCTOBER.

Join Computer Learning Month

WHY A COMPUTER LEARNING MONTH?

Learning with computers is giving America's youth the winning edge. Never before has there been such an exciting event devoted exclusively to bringing parents, teachers and kids together for a month-long celebration of learning with software and computers. It's called Computer Learning Month, and it's happening this October. Computer Learning Month is so important that a commemorative bill has been introduced into both Houses of Congress declaring October 1987 as Computer Learning Month. Sponsored by major educational software and computer companies and supported by national educational organizations and State Departments of Education, Computer Learning Month will be an event in which everyone will be a winner – an event you won't want to miss.

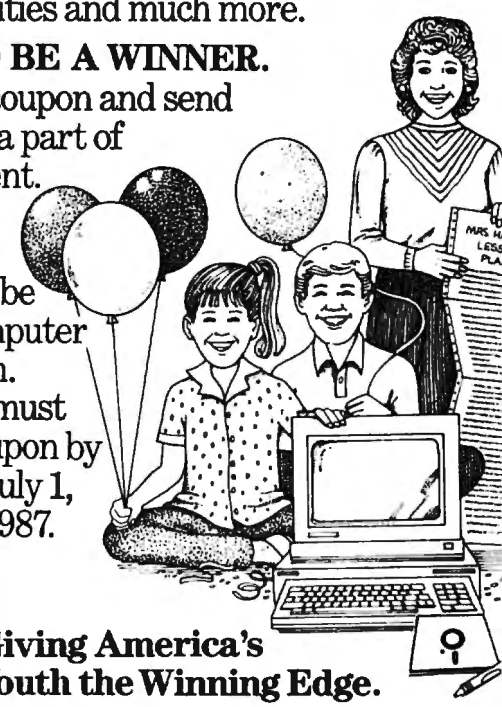
EVERYONE WILL BE A WINNER... Contests! Contests! Contests! Thousands of dollars worth of great prizes will be given away in October. Essay and art contests for families and kids. Lesson plan contests for teachers. And if you are a prize winner, your school will be a prize winner, too.

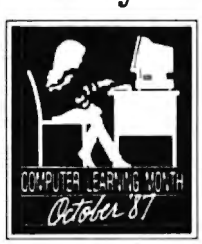
Exciting programs for families. Now parents can learn how to help their children benefit from using a computer, whether they own one or not. We're giving away a free booklet designed just for parents. There will be fairs and special events sponsored by local retailers. And we know schools will be hosting special parent nights.

A month full of activities for teachers. Teachers can join in the celebration easily and learn new ways to help their students benefit from using computers. Every teacher can receive a celebration poster, chock-full of lesson plan ideas, activities and much more.

IT'S EASY TO BE A WINNER.

Just clip this coupon and send it in today to be a part of this national event. We'll send you everything you need so you can be a winner in Computer Learning Month. Don't delay. We must receive your coupon by July 1, 1987.





Giving America's Youth the Winning Edge.

YES! I WANT TO BE A WINNER IN OCTOBER!

☐ Please send me contest rules.

☐ Please add my name to the Computer Learning Month Mailing List.

I am a ☐ teacher ☐ parent ☐ student

I own a computer ☐ yes _____ ☐ no
(type of computer)

NAME _____

SCHOOL _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE _____

Mail Coupon to: Computer Learning Month
P.O. Box 19763
Washington, D.C. 20036-0763
(202) 452-1600

Computer Learning Month 1987 is Sponsored by: Software Publishers Association, Advanced Ideas, Britannica Software (Publishers of DesignWare,® EduWare,® Blue Chip Software™), Broderbund Software,™ COMPUTE! Publications, Inc./ABC Publishing, Davidson and Associates, Inc., The Learning Company, Mindscape, Inc., Random House Media, Scholastic, Inc., Spinnaker Software, Springboard Software, Inc., Weekly Reader Family Software.

Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

Rooting Around

I have a question about math on my Commodore 128. I know how to get the computer to give square roots (with the SQR function), but how do I get other roots? For example, how do I get the fifth root of 78?

Neal Hatton

Although your question refers to BASIC 7.0 on a Commodore computer, the answer applies to most computers with BASIC. You can take advantage of the fact that the fifth root of 78 is the same as 78 raised to the 1/5 power. Try this:

```
ROOT = 78↑(1/5) : PRINT ROOT
```

You can check your answer like this:

```
PRINT ROOT↑5
```

You'll get a number near 78. The number is slightly off because of the limited accuracy of the exponentiation algorithm.

Applesoft Memory Management

I have written an 8K program in Applesoft BASIC and I need to use hi-res graphics page 1, but my program interferes with this page. Can you tell me how to move my program into the open area above page 1?

J. Larry Gaither

For a program stored on disk with the name PROGRAM, include the following as the first line of the program:

```
5 IF PEEK(104)<>64 THEN POKE104
  ,64:POKE 16384,0:PRINT CHR$(4
  )"LOAD PROGRAM"
```

When you're running a program that uses only the text screen, you have 36,348 bytes of memory available for storing the program and all its variables. But using hi-res screen 1 limits your program and its numeric variables, which are normally stored before the screen in memory, to 6140 bytes. String variables, which are stored after hi-res screen memory, have 22,016 bytes available. The worst part is that Applesoft BASIC won't tell you when your program and variables fill up their area, as you've no doubt already discovered. If you move your program, all the code and variables will be stored in the larger area above the screen. (Since Applesoft uses 4 bytes to indicate the start and end of program storage, you'll get only 22,012 usable bytes.)

Short programs (less than 6K) which use large array variables might work better if you moved only the numeric variables, leaving the program alone in the lower area. The command to do this is LOMEM: 16384. If the program permits, you might also try switching to hi-res page 2. When using this page, the memory area before the screen is 14,332 bytes long; the area after the screen is 13,824 bytes long. For larger programs, the numeric variables can be moved to the second area with the command LOMEM: 24576.

What You See Is Not...

Why doesn't this program work on my Commodore computer?

```
10 A = SQR(9)
20 IF A = 3 THEN GOTO 10
30 PRINT A
```

When I run this program, it falls through to line 30, printing the value of A, rather than branching back to line 10 as you would expect. Line 30 shows that A is equal to 3, so why doesn't the IF test work?

Christoph Khouri

There are two causes for this discrepancy: the way that the SQR function computes a square root and the way that the computer rounds off some numbers. SQR derives the square root of a number by finding the number's natural logarithm and then dividing that quantity by 2. The result is accurate enough for most practical purposes, but in this case, it yields a number that is very close to 3—but not exactly 3.

When it stores the result of SQR(9) in the variable A, the computer holds the number with a considerable degree of accuracy, including the very small fraction that prevents the result from being exactly 3. The IF test in line 20 compares the number 3—which is exactly 3—with the result of SQR(9), which is a little bit off. Since the two values aren't exactly the same, the IF test fails, and the computer proceeds to line 30.

In that case, you might ask, why does the PRINT statement in line 30 display exactly 3 instead of some fractional value? PRINT is generally accurate, but it rounds off very small fractions. This isn't a defect in the computer, but simply a consequence of the fact that some fractions can't be represented with complete precision in a given numbering system. For instance, in decimal notation, the fraction ⅓ is represented as .33333333... with the 3s carried out an infinite number of places. In this case, the fractional component of the result is so small that PRINT rounds it off rather than attempting to represent it.

Similar anomalies occur in every version of BASIC. Fortunately, few of them cause any real problems unless you are doing serious math which requires a high degree of precision.

Atari Miscellany

I have a few questions about Atari computers. What does the XIO statement do, and what is its syntax? How can I simulate the MOD function? How can I make cartridges for the Atari VCS videogame machine work on my computer?

D. Botha

The XIO statement is a general I/O (Input/Output) statement. Atari BASIC already has many I/O statements such as OPEN, CLOSE, GET, and PUT, but there are some tasks that you cannot do without XIO. For example, the following statement deletes a disk file named TEMP.BAS:

```
XIO 33,#1,0,0,"D:TEMP.BAS"
```

This can be done in program mode or direct mode. You can learn more about XIO in the Atari 400/800 BASIC Reference Manual *and similar sources.*

MOD is the remainder of an integer (whole number) division. Here is an example of how to translate the expression LOW = LOCATION MOD 256 into Atari BASIC:

```
HI=INT(LOCATION/256)
LOW=LOCATION-HI*256
```

From the publishers of *COMPUTE!*

June 1987 *COMPUTE!* Disk

All the exciting programs from the past three issues of *COMPUTE!* are on one timesaving, error-free, floppy disk that is ready to load on your Atari 400/800, XL, and XE. The June 1987 *COMPUTE! Disk* contains the entertaining and useful Atari programs from the April, May, and June 1987 issues of *COMPUTE!*.

The June 1987 *COMPUTE! Disk* costs $12.95 plus $2.00 shipping and handling and is available only from COMPUTE! Publications.

For added savings and convenience, you may also subscribe to the *COMPUTE! Disk*. At a cost of only $39.95 a year (a $12.00 savings), you'll receive four disks, one every three months. Each disk will contain all the programs for your machine from the previous three issues of *COMPUTE!*. To order a subscription, call toll free 800-247-5470.

This is an excellent way to build your software library while you enjoy the quality programs from *COMPUTE!*.

Disks and subscriptions are available for Apple, Atari, Commodore 64 and 128, and IBM personal computers. Call for details.

For more information or to order individual issues of the June 1987 *COMPUTE! Disk*, call toll free 1-800-346-6767 (in NY 212-887-8525) or write *COMPUTE! Disk*, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

COMPUTE! Publications, Inc. abc

Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019

Publishers of COMPUTE!, COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, COMPUTE!'s Apple Applications, and COMPUTE!'s Atari ST Disk & Magazine.

Finally, it is impossible to usefully connect VCS cartridges to your Atari computer. Although both devices have a 6502-based microprocessor, their video display hardware is totally incompatible. Rigging up the necessary hardware would probably cost you more than the combined price of a VCS and a 65XE computer.

Reading Amiga Joysticks

I am programming in Microsoft Amiga BASIC, and I want to write a program that uses two joysticks. How can I read a joystick plugged into joystick port 1 (the port where the mouse connector normally goes)?

H. Manson

Amiga BASIC's STICK function works for port 2, but not for port 1. Since the mouse is used to access BASIC's own menus, the designers may have assumed that you would never want to unplug it from port 1. However, it is possible to read a joystick in port 1 by PEEKing a location in memory. Location 14675978 ($DFF00A) is a 16-bit register that contains the information you need. The position of the joystick is returned in bits 0–1 and 8–9 of this location. Different bits are set to a value of 1 depending on which direction you press the joystick:

right:	*bit 1 = 1*
left:	*bit 9 = 1*
down:	*(bit 0 XOR bit 1) = 1*
up:	*(bit 8 XOR bit 9) = 1*

Location 14675980 ($DFF00C) contains corresponding information for joystick port 2.

BASIC's BUTTON function also works only for port 2. Here is a program that prints the directions and button status of a joystick in port 1.

```
Stick1:
jp1%=PEEKW(14675978&)
bit1%=jp1% AND 2
bit9%=jp1% AND 512
IF bit1%=2 THEN PRINT "right ";
IF bit9%=512 THEN PRINT "left ";
IF (jp1% AND 1) XOR bit1%/2 THEN
PRINT "down ";
IF (jp1% AND 256) XOR bit9%/2 TH
EN PRINT "up ";
IF (PEEK(109528958) AND 64)=0 TH
EN PRINT "fire";
PRINT
GOTO Stick1
```

The last IF statement in the program checks the joystick button in port 1. If for some reason you don't want to use the BUTTON command for port 2, you can test the button with this statement:

```
IF (PEEK(109528958) AND 128) = 0
THEN PRINT "Port 2 button"
```

Dynamic Keyboard For The IBM

I have seen a few programs for the Commodore 64 employ a technique you call the dynamic keyboard. The technique adjusts memory locations to make the computer think that someone has typed something on the keyboard. I would like to know how to do this on an IBM PC compatible like my Tandy 1000.

Kevin O'Donovan

The dynamic-keyboard technique can be implemented on most computers which feature a keyboard buffer, including the IBM PC and compatibles. A keyboard buffer is an area of memory where information received from the keyboard is held while awaiting processing. In the IBM PC, PCjr, and true compatibles the buffer normally resides at locations 1054–1085 (&h41E–&h43D). Each time a key is pressed, two bytes are placed into the buffer. The first is the ASCII character code for the key, and the second is the keyboard-scan code for the key. For keys like the function keys which have no corresponding ASCII code, the first byte will be 0, and the second will be the extended keyboard-scan code for the key. Tables of ASCII and extended keyscan codes can be found in your BASIC manual.

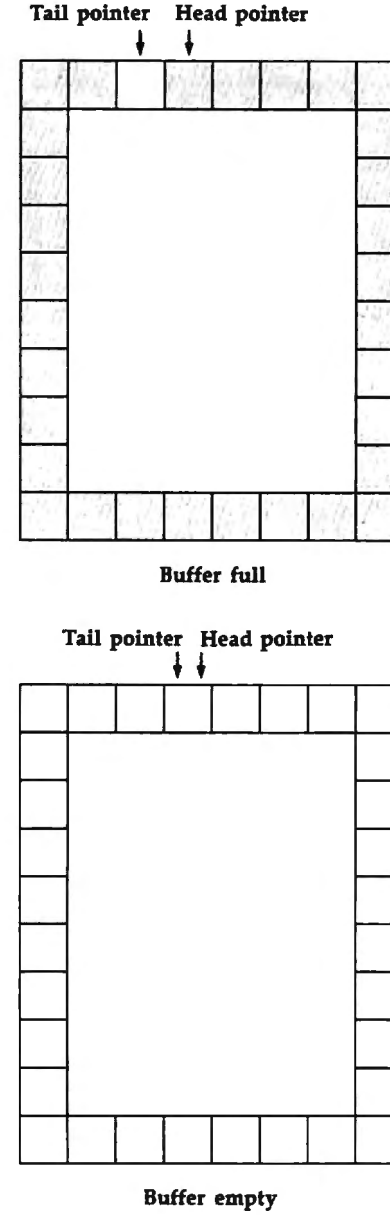
Locations 1050 (&h41A) and 1052 (&h41C) are used as buffer pointers. Location 1050 points to the head of the buffer (the first entry waiting to be read), and location 1052 points to the tail of the buffer (one byte beyond the last entry). Each time a keypress is added to the buffer, the value in location 1052 is incremented by 2. Each time an entry is read from the buffer, the value in 1050 is decremented by 2. The buffer is circular; that is, after a keypress has been recorded in the last two bytes of the buffer, the next keypress will be recorded in the first two bytes. The buffer is considered empty when the value in location 1052 equals the value in 1050, and the buffer is considered full when the value is 1052 is 1 less than the value in 1050. Thus, only 15 of the 16 possible entries in the 32-byte buffer can be used (the maximum number of keypresses the buffer can hold is 15).

To use the dynamic-keyboard technique, POKE the ASCII codes for the desired characters into the buffer. (Remember to use only every other buffer location; it is not necessary to POKE the keycode bytes unless you are trying to enter a key with an extended keyscan code.) If you want the text to be acted upon automatically, remember to include a carriage return (character code 13) as the final character. Then POKE location 1050 with the offset (from location 1024/&h400) to the first character of the entry (use 30/&h1E for the first byte of the buffer) and POKE location 1052 with the offset immediately beyond the last character of your entry. Once you have set up the buffer and pointers, the next time BASIC looks to the keyboard buffer for input, it will find the POKEd characters and process them just as if they had been typed.

One possible use for this technique is to provide default entries for INPUT statements. The following example program illustrates this:

```
100 SCREEN 0:WIDTH 80:CLS:DEF
    SEG=0
110 PRINT:PRINT
120 DEFAULT$="Greensboro":GOS
    UB 1000
130 INPUT "City:  ",CITY$
140 DEFAULT$="NC":GOSUB 1000
150 INPUT "State: ",STATE$
```

IBM Keyboard Buffer



```
160 PRINT:PRINT CITY$;", ";ST
    ATE$:PRINT
170 DEFAULT$="y"+CHR$(29):GOS
    UB 1000
180 INPUT "Another entry";OPT
    $
190 IF OPT$="Y" OR OPT$="y" T
    HEN 110
200 END
1000 DEFAULT$=LEFT$(DEFAULT$,
     15):DBL=2*LEN(DEFAULT$)
1010 FOR II=1 TO DBL STEP 2
1020   POKE &H41D+II,ASC(MID$
     (DEFAULT$,(II+1)/2,1))
1030 NEXT
1040 POKE &H41A,&H1E
1050 POKE &H41C,&H1E+DBL
1060 RETURN
```

The subroutine beginning at line 1000 places whatever is in the variable DEFAULT$ into the buffer (limited to 15 characters by the LEFT$ function in 1000), and then sets the buffer pointers appropriately. Line 170 illustrates how cursor-movement characters can be added: The CHR$(29) is a cursor-left character, which moves the cursor back onto the y after it is printed.

A more common use of the technique is to make programs self-modifying. This scheme relies on BASIC's full-screen editor. You can make a program add lines to itself by printing the desired new program lines at a known location on the screen, and then placing a carriage return (character 13) into the buffer for each line to be added. Here is a very simple address-file program which illustrates this method:

```
10 NEWLINE=300 :ENTRIES=0
20 SCREEN 0:WIDTH 80:CLS
30 PRINT:PRINT"Micro Filer":P
   RINT
40 PRINT "Select: [1] Review
   addresses":PRINT TAB(9)"[2
   ] Add addresses":PRINT
50 K$=INKEY$:IF K$="" OR K$<"
   1" OR K$>"2" THEN 50
60 IF K$="2" THEN 100
70 RESTORE:FOR I=1 TO ENTRIES
   :READ PERSON$,ADDRESS$,CIT
   YSTATE$
80 PRINT PERSON$:PRINT ADDRES
   S$:PRINT CITYSTATE$:PRINT
90 NEXT:GOTO 30
100 INPUT "Name:       ",PERSON
    $
110 INPUT "Address:    ",ADDRES
    S$
120 INPUT "City:       ",CITY$
130 INPUT "State:      ",STATE$
140 INPUT "Zip code:  ",ZIP$
150 CITYSTATE$=CHR$(34)+CITY$
    +", "+STATE$+" "+ZIP$+CHR
    $(34)
160 CLS:LOCATE 2,1
170 PRINT STR$(NEWLINE)+"data
     "+PERSON$+","+ADDRESS$+"
    ,"+CITYSTATE$
180 PRINT "10 newline=";NEWLI
    NE+10;":entries=";ENTRIES
    +1
190 PRINT"run"
200 DEF SEG=0:POKE &H41E,13:P
    OKE &H420,13:POKE &H422,1
    3
210 POKE &H41A,&H1E:POKE &H41
    C,&H24
220 LOCATE 1,1:END
```

After accepting name and address information, the program creates a DATA statement containing the new entry and prints it on the screen (line 170). It also prints an updated line 10 and a RUN command, then places three return characters into the buffer. Finally, the program relocates the cursor one line above the printed lines and ends. When a BASIC program ends, BASIC prints an Ok prompt, moves the cursor to the next line, and looks for keypresses. In this case, the return characters in the buffer will cause the three printed lines to be entered, and the RUN command in the last of the three lines will restart the newly modified version of the program. The major drawback to using this technique is that all previous variable values are lost when the program is run again. This is why it is necessary to update line 10 to reflect the new values for NEWLINE and ENTRIES.

Refer to Chapter 2 in COMPUTE!'s Mapping the IBM PC and PCjr *for more information on the keyboard buffer, including ways to relocate the buffer and increase its size. The example programs should work on most IBM-compatible computers (they were tested on a Tandy 1000EX), but we cannot guarantee that they will work as listed on every variety. It is possible that some machines may place the buffer at locations other than those used by IBM.*

Stay Out Of My Zero Page

I am writing a simple graphics routine for the Commodore 64. I have been trying to fill the screen randomly with the ball character, using this BASIC line:

```
10 X=INT(RND(1024)*2023)+1:POK
   EX,81:GOTO10
```

The computer puts a few balls on the screen and then locks up. What causes this problem?

Kevin Bentley

*You're aiming at the right area (screen memory, locations 1024–2023), but you're using a faulty formula to calculate a random position within that zone. No matter what value you put inside its parentheses, the RND function always returns a fractional value between 0 and 1. The result of RND(1024) is the same as RND(1) or RND with any positive number inside parentheses. Thus, the expression INT(RND (1024)*2023)+1 generates random numbers in the range 1–2023. Some of these POKEs may go in the screen memory area, but many of them will go below the screen, into sensitive memory zones such as the zero page—the lowest 256 bytes of memory. Nearly all of the locations in the zero page are reserved for the computer's own internal use, and you can easily lock up the machine with careless POKEs in this zone.*

Fortunately, no real harm occurs when you crash the computer with a misdirected POKE. Turn the machine off and on to reset it. Here is a program that properly POKEs randomly colored ball characters at random locations on the screen.

```
10 PRINT CHR$(147) CHR$(142)
20 POKE 53280,0:POKE 53281,0
30 JUNK=RND(-TI):BALL=81
40 SCREEN=INT(RND(1)*1000)+102
   4
50 COL=INT(RND(1)*15)
60 POKE SCREEN+54272,COL
70 POKE SCREEN, BALL
80 GOTO 40
```

To obtain better randomness, line 30 seeds the random number generator with the expression RND(−TI). The expression in line 40 generates a random number in the range 1024–2023, the correct area for screen POKEs, and line 50 generates a color number in the range 0–15. Line 60 POKEs the randomly selected color value into the color memory location that matches the screen location we chose. Color memory is located at 55296–56295, exactly 54,272 bytes above screen memory. Once the new color has been set, line 70 places the ball on the screen. When POKEing to screen memory, it's usually best to POKE into the matching color memory location first.

Modified Numeric Keypad

Thank you for "Numeric Keypad in 64 Mode" ("Readers' Feedback," October 1986), which allows Commodore 128 owners to use the computer's numeric keypad when in Commodore 64 mode. That program is very useful. However, I would like to change the decimal point on the keypad to a comma, so that when I'm typing DATA statements, I will have the comma in a convenient location.

S. A. Seekins

Load the program, add this line, and save the program under a new name.

```
125 POKE 927, 47
```

This POKE makes the period key produce a comma. POKE 927,44 restores the key to normal. C

Murder Party

Neil Randall

Requirements: Commodore 64, Apple II series, and IBM PC and compatible computers. A printer is required.

As everyone knows, computers provide capable, tireless, and socially unobjectionable opponents for games of all types. After a while, though, it's possible to grow tired of being trounced by computers. And it's also possible to grow tired of playing solitaire games.

Remember those long multiplayer games of the past—before computers—in which socializing was as important as getting the game finished? They had to appeal to the greatest number of people possible. They had to be easy to play, yet challenging enough to satisfy. And they had to have a subject which captivated us.

More often than not, we chose murder.

Murder mysteries. Sherlock Holmes. Hercule Poirot. Perry Mason. Jessica Fletcher. Miss Scarlet in the kitchen with the candlestick. And so on.

Electronic Arts' *Murder Party* allows you quickly and easily to create a party in which you and your guests will try to solve a murder. You need anywhere from five to seven guests. Each of them, in addition to yourself, will play the role of a suspect in the case, with all of you picking up clues from conversations and distributing clues you are given. At the end of the evening, whoever solves the mystery wins the game.

Verdict, Please

The rules are almost nonexistent. The game is divided into four rounds. During each round, every guest must give out some information to the others as discreetly as possible. As host, you try to determine when everyone has done so, and then you move them into the next round. Once the four rounds have ended, the guests fill out a Verdict Sheet, and together you figure out who has won.

Noncomputerized versions of this type of game exist, but where the computer helps is its ability to keep information hidden from everyone, even the host. In fact, the host does not just run the event; he also plays the game. *Murder Party* scrambles the clues and the murderer's identity, so not even the host knows who did it.

Creating a party is extremely easy. Instructions on the screen take you through the creation step by step, while information in the manual helps you get the party going. First, you choose from one of two scenarios: The Big Kill, or Empire. The disk instructions introduce you to *Murder Party*, then provide information about either scenario. After choosing one, you can then read the invitation that will be sent to the guests, learn more about the suspects to help you assign roles, or, finally, create or edit a party.

To create the party, you fill in the blanks on the screen. You must specify the date, time, and place of the get-together, assign the guests to their roles, and then print out the materials. Casting roles involves choosing which guest will play which role, then entering that guest's name and address. Figuring out who will best suit each part is an extremely interesting exercise, since it means that you should have a reasonable grasp of your guests' personalities.

By Invitation Only

Printing the party materials takes up to two hours with a dot-matrix printer, and it consumes roughly 100 sheets of paper. But it's worth it. *Murder Party* individualizes each packet of information, providing each player with different, sometimes conflicting, information. A few weeks before the party starts, you print out the invitations and envelopes, sending them to the guests without examining them yourself (except your own, of course). Just before the party, you print out the materials for the host (police records, personnel files, coded notes, even a check) and the clues for the guests. The program warns you against the temptation to examine the clues, since doing so will give it all away. Instead, you put them in sealed envelopes and hand them to the guests when the party begins.

The software does all this, while the manual helps you organize the party itself. It provides suggestions and hints for making the guests comfortable (even those who might not enjoy games), and it goes so far as to suggest recipes for food to serve at dinner, music to play in the background, and clothes to capture the ambience. The Big Kill, for instance, is all about a group of ex-Berkeleyites from the sixties getting together just after a friend's death (yes, you've heard this one before), and the manual suggests pulling out your old Woodstock, Canned Heat, or Bob Dylan albums and putting up posters of Joplin and Hendrix. Exceptionally useful, the manual should get you through your first party quite effortlessly. Electronic Arts plans to publish party disks, each with two or more additional scenarios, but the two provided are variable and can be re-used.

For anyone interested in staging such a party, *Murder Party* is exactly right. Not only does it take the drudgery out of producing the needed materials; it also allows you to play the game, not just to organize it and watch it run. Unlike the game master in a standard role-playing game, who spends hours learning rules, drawing maps, and anticipating responses from the players, the host of *Murder Party* simply decides whom to invite, prints out the materials, and plans for an entertaining evening.

Murder Party
Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
$32.95 Commodore 64 version
$39.95 Apple II series and IBM PC and compatibles versions

Gridiron

Ervin Bobo

Requirements: Amiga or Atari ST. 512K required.

When you've booted *Gridiron* on the Amiga, the title screen presents you with an excellent picture of a ball carrier twisting out of double coverage; the last few bars of "The Star-Spangled Banner" are heard; and the crowd roars. But be ready for a surprise—this is the last you'll see of anything that looks like real football players.

Instead, your view of the field will show you colored dots moving through the play action, resembling nothing so much as an animated chalkboard—but without the arrows. After just a few minutes, you'll realize that this is a pretty good way to play the game.

From opening menus, you choose whether you wish a one- or two-player game, and you set the length of the quarters for 5, 10, or 15 minutes. The shorter times do not allow for a long-range strategy because there will not be as many possessions per team. With the longest quarters, the average game can be played in two hours.

Practice First

Even in the Beginner level, the computer will be a tough opponent. With difficulty being based on factors of speed, strength, and intelligence, you'll want to do a lot of practice before challenging it at any of the three higher levels.

Next you choose whether to use the Standard Playbook or one that you have created using the Play Creation Utility; and whether to use the default team or draft a new team that you have created by altering Player Ability Points. Go through the same steps to select the opposing team, whether controlled by the computer or a friend, and you're ready for the coin toss and the beginning of the game. All selections are made with the mouse. In fact, in playing the game you'll use the keyboard only when entering the names of your teams and the validation code that is a familiar form of copy-protection.

The overhead view will show you about 70 percent of the football field. When action spills across the screen border, the game will momentarily freeze, and then the field will be redrawn to show the missing end. This is a rather nice feature, since scrolling might not allow enough time for you to get the mouse pointer into position—and in this contest you're going to need every break you can get.

In any play, one of your players is controlled by the mouse pointer. He's easy to spot because, a moment before the ball is snapped, he changes color. Hopefully, you'll have your pointer in position, because this player will follow it wherever you drag it—providing no one gets in his way. If you're playing offense, he'll be the quarterback; you'll want to drag him back a bit, then click the left mouse button when you're ready for him to throw. Assuming the pass is completed, the receiver now changes color and will be controlled by the dragging of the pointer, allowing you to weave around the opposing players and make the touchdown—maybe.

It is this single-player control that makes *Gridiron* unique among computerized football games. Others allow some degree of control, but only for the team as a unit, after which you can put your hands in your lap and wait to see how it turns out. *Gridiron*, by allowing this player control, allows you to alter your strategy to fit the moves of the other team.

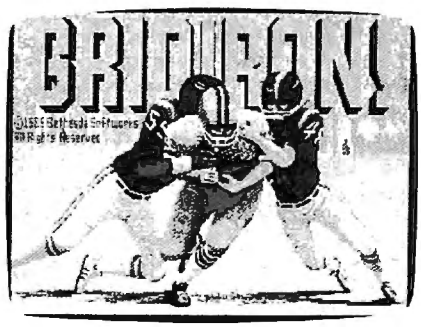
For setting up a play or a defense, a small box appears in the upper left of the screen and allows you to select a number between 0 and 19; these numbers correspond to the plays in the playbook, or to plays you may have built on your own. While you are doing this, a larger scorecard fills the center of the screen and keeps you from seeing the formation selected by the opposing team. This situation lasts until 20 seconds before the play. If you're on offense, you decide when to snap the ball; otherwise you wait and try to react.

When doing this, you need to remember that all your players will follow the playbook pattern exactly, and that only one man, the one controlled by the mouse, will have the freedom to deviate from the plan.

Play Creation And Player Draft Options

Your own playbook can be created with the Play Creation Utility, in which you can modify either offensive or defensive plays. First, select any of the 20 listed plays from the menu and, if you wish, change the name of the play. Select a player by pointing and clicking; place him on the field; and, from the new edit menu, select the action you wish the player to take. (This explanation is somewhat simplified for space considerations.)

In building players capable of carrying out your schemes—or just better qualified to run the plays in the default playbook—you select the Player Draft option and alter the Player Ability Points for each selected player. Speed and strength of each player is given a numerical value that you may raise or lower as you see fit and, during play, the computer takes all this into account: A ball carrier with high strength will not necessarily be brought down or even delayed unless the tackler is of equal strength.



The title screen of Gridiron *(Amiga version) from Bethesda Softworks.*

Both the altered players and the altered plays can be saved to a separate data disk for future use. (It occurred to me that you could create a team of wimps who run the wrong way and then force the computer to use that team. So far, it has only occurred to me: It's up to you to figure out the ethics of such a move.)

With the ability to alter players, teams, and the playbook, *Gridiron* presents some options never before seen in such a simulation. The fact that it uses moving dots rather than sprites and blitter objects to represent the players is no handicap to the action and strategy—and may in fact be a blessing for some.

Documentation is very good. And the sound continues throughout the game—not only will you hear the crowd roar; you'll also hear the quarterback making his "hut...hut" calls; hear the grunts as players collide; and, through synthesized speech, hear the referee calling the results of plays.

With all these things going for it, *Gridiron* should become a favorite of die-hard football aficionados as well as offer some appeal for those who just like to play a good game.

Gridiron
Bethesda Softworks
9208 Burning Tree Road
Bethesda, MD 20817
$59.95 Amiga version
$49.95 Atari ST version

Destroyer

Scott Thomas

Requirements: Commodore 64, Apple II series with a minimum of 128K, IBM PC and compatibles.

Destroyer, a new game simulation by Epyx, is an interesting blend of strategy and arcade action. Obviously a counter to the numerous submarine simulations that have infested the silicone seas over the last year, *Destroyer* is a simulation of the Fletcher-class US Destroyer from World War II. Besides having some excellent arcade action and strategy elements, the program features some of the best graphics I've seen on a Commodore 64. It puts you in command of a 3000-ton fighting ship with a 325-man crew. Destroyers like the Fletcher-class were the workhorses of the Navy during World War II. Besides providing valuable submarine and aircraft screen for naval and merchant convoys, destroyers ran blockades, rescued downed pilots, bombarded coastal positions in support of invasion forces, and engaged in surface-to-surface combat with other naval vessels. The program provides seven missions of increasing difficulty, each one providing a different combination of tasks that permit you to experience the various activities that destroyers performed during the war.

Destroyer requires a joystick, and loads from a single disk. Due to the numerous graphics screens for the ship stations, the program must access the disk at various points through the game. The program's loading time and subsequent disk-access time, however, is kept to a minimum by the implementation of Epyx's fast *Vorpal* load utility.

Missions And Orders

You begin the program by providing your name and one for the ship in your command. The next step permits you to select from the seven different missions and select from three difficulty levels (easy, intermediate, or advanced). Once your choices are made and your mission selected, you will receive your orders, which describe the mission you must undertake and its objective. Thereafter you enter the navigation station, where plotting the ship's course is accomplished.

With the course plotted, you are then confronted with a screenful of instrumentation known as the Bridge station. From the Bridge, you have a certain amount of control over the other ship stations. Those stations include radar, sonar, surface guns, anti-aircraft guns, torpedoes, and depth charges, as well as damage control. While you can go to each of these stations by giving a two-letter command, you can receive information from and give certain limited commands to some or all of those stations from the Bridge. Also from the Bridge, as well as from several other stations, you can change the heading and speed of the ship from the plotted course.

Destroyer, *from Epyx, is a fast-paced blend of action and strategy with excellent graphics.*

It is from the Bridge that you first become aware of the strategy/arcade dichotomy of *Destroyer*. The ship does indeed have a large crew, and all of the stations under your command will, if ordered, operate automatically. On the other hand, you can take manual control of any station—which, with respect to the weapons stations, puts you into an arcade-style action mode. Taking manual control of these stations will be too tempting for some arcade enthusiasts, as the graphics and action are very good. However, even arcade enthusiasts will quickly learn how fast a vessel can be destroyed and the mission lost if they spend too much time manually shooting down aircraft or torpedoing enemy ships.

The reverse situation also is true. You cannot command your ship from the Bridge and succeed in your mission. While your crew can perform with minimum levels of adequacy at each station, an attack on your vessel or a command by you requiring the capabilities of a particular station will most often require your direct attention for a period of time. Accordingly, *Destroyer* demands a precise blend of the commander's use of the strategy and action aspects to succeed on the mission.

One of the most interesting and challenging stations is damage control. By typing the letters *DA*, you are transferred to the damage-control station, which gives you a cross-section view of your ship and a list of the areas of the ship that can incur damage. Also listed are four damage-control parties, identified as Alpha, Baker, Charlie, and Delta teams. The teams are listed in order of their speed of repair, with Alpha team being the fastest. When damage is incurred, the teams are assigned automatically in descending order to fix the repairs. However, you can manually reassign the parties to any damaged stations. The reassignment capability will be critical to your success—particularly in the more difficult missions, where some damage to your ship is almost certain—since the automatic assignment routine may assign the fastest teams to the least important stations for repairs. Depending upon your mission and the circumstances of the moment, the repair of damage to the steering system or sonar may be much more critical than damage to the surface guns or the torpedo stations.

The instruction manual that accompanies the program is brief, but complete and informative. It includes a breakdown of each station, as well as some interesting and useful information about the Fletcher-class Destroyer.

In summary, *Destroyer* is one of those hybrid strategy/arcade simulation games that excels by performing well in both areas and by requiring skills in both areas for player success. The program maintains Epyx's reputation for superb graphics, and is fast-paced, with plenty of action. If you fancy yourself to have "the right stuff" to be captain of a fighting ship, *Destroyer* can certainly provide you with a taste of the rewards and rigors of command.

Destroyer
Epyx
P.O. Box 8020
600 Galveston Rd.
Redwood City, CA 94063
$39.95 all versions

All programs listed in this magazine are available on the GAZETTE Disk. Details elsewhere in this issue.

SILICON EXPRESS DISKS SS/DD 6.95 DS/DD 7.95

VERBATIM OR MAXELL SS/DD 13.95 DS/DD 16.95

ELEPHANT SS/DD 7.95 DS/DD 8.95

DATACASE 7.95

INSIDE OHIO 1-800-225-0755

YOUR TOLL FREE ACCESS OUTSIDE OHIO

1-800-228-0755

CALL...FOR THE BEST PRICES... BEST DELIVERY...

We have hundreds of other programs in stock at Fantastic savings.

SILICON EXPRESS

LASER 128—APPLE COMPATIBLE COMPUTER $389.00

	IBM
Algeblaster	27.89
American Challenge	24.89
ASC II Express	74.89
Balance of Power	33.89
Bank Street Writer+	58.89
Breakers	28.89
Certificate Maker	41.89
Library Vol. 1	19.89
dBase III+	419.89
Dollars & Sense	104.89
Easy	88.89
Flight Sim	28.49
Fontrix	88.89
Fontasy	48.89
Fntpks 1 thru 13	ea. 13.89
Framework II	419.89
Fully Powered PC	24.89
Gamecard III	39.89
Gato	21.89
Hacker I or II	24.89
Helicat Ace	24.89
Jet	31.49
King's Quest I or II	31.89
Load Runner	22.89
Locksmith	56.89
Lotus	319.89
Mean 18	34.89
Mngng Yr Money	114.89
Micro Cookbook	27.89
Microsoft Word	279.00
Multiplan	110.89
Newsroom Pro	74.89
Norton Utilities	58.89
Option Board	79.89
PC Tools	18.89
PFS: Access	59.89
PFS: Files Profes.	148.89
PFS: First Choice	104.89
PFS: Plan Profes.	148.89
PFS: Write Profes.	148.89
Print Shop	34.89
Print Shop Companion	32.89
Print Shop Gr. #1	21.89
Reflex Database	86.89
Report Card	32.89
Sidekick	50.89
Sideways Ver. #3	38.89
Silent Service	22.89
Speedkey	104.89
SuperCalc 4	279.89
Superkey	40.89
Symphony	494.89
Think Tank	99.89
Toy Shop	46.89
Traveling Sidekick	40.89
Turbo Database Tlbx	40.89
Turbo Lighting	58.89
Turbo Pascal 3.0	59.89
Turbo Prolog	58.89
Turbo Tutor	23.89
Ultima III	40.89
Visable 8088	47.89
Volkswriter Dlx #3	141.00
Web. Spell Checker	39.89
Wizardry	36.89
Word Perfect	289.00
Wordstar 2000+	284.89

APPLE SOFTWARE	
Adv. Construction	33.89
Airheart	22.89
Algeblaster	27.89
Alpha Plot	20.89
ASC II Express	74.89
Auto Works	25.89
Bag of Tricks 2	31.89
Bank Street Filer	42.89
Bank Street Mailer	42.89
Bank Street Writer	42.89
Bank Street Speller	42.89
Bard's Tale	39.89
Beagle Bag	16.89
Beagle Basic	18.89
Beagle Compiler	41.89
Beagle Graphics	30.89
Beneath Apple DOS	12.89
Big U	18.49
Bookends	66.89
Bop & Wrestle	18.89
Captain Goodnight	22.89
Carmen Sandiego	25.89
Certificate Maker	29.89
Library Vol. 1	19.89
Chart'n Graph Tlbx	23.89
Classmates	27.89
Database Toolbox	23.89
Dazzle Draw	37.89
D-Code	20.89
DiskQuick	15.89
Dollars & Sense	71.89
DOS Boss	12.89
Double-Take	18.49
Eidolon	24.89
Essential Data Dup	20.89
Extra K	20.89
Fantavision	31.89
Fat Cat	18.49
Flex Type	15.49
Flight Sim II	34.89
Fontrix 1.5	58.89
Fntpks 1 thru 15	ea. 13.89
Font Works	33.89
Forecast	37.89
Frame-Up	15.49
Gato	21.89
G I Joe	24.89
Graphics Department	28.89
Graphic Magician	25.89
Graphworks	44.89
GPLE	25.89
Handler Package	58.89
Impossible Mission	24.89
I.O. Silver	15.89
Jeeves	28.89
Karateka	22.89
Knight of Diamonds	21.89
Kid Writer	24.89
King's Quest I, II or III	31.89
Koronis Rift	24.89
Legacy of Llylgamyn	24.89
List Handler	36.89
Locksmith	38.89
Macroworks	18.89
Mngng Yr Money	98.89
Megaworks	CALL
Merlin	41.89
Merlin Pro	64.89
Merlin Combo	68.89
Micro Cookbook	27.89
Millionaire	31.89
Minipix 1,2 or 3	15.49

APPLE SOFTWARE	
Moebius	40.89
Mouse Talk	69.89
MouseWrite	88.89
Munch-A-Bug	23.89
NATO Commander	22.89
PFS: Plan	75.89
PFS: Write	75.89
PFS: File	75.89
Piece of Cake Math	23.89
Pinpoint	48.89
Prince	48.89
Printographer	23.89
Printmaster	24.89
Print Shop	31.89
Prt Shp Gr. 1,2,3 or 4	15.89
Print Shop Companion	24.89
Probyter	18.49
Pronto DOS	15.49
Program Writer	33.89
Power Print	20.89
Quicken	34.89
Report Card	32.89
Rescue on Fractalus	24.89
Reportworks	CALL
Rescue Raiders	21.89
Science tool kit	44.89
Sensible Grammar	55.89
Sensible Speller	58.89
Shape Mechanic	20.89
Sideways	38.89
Silicon Salad	12.89
Skyfox	27.89
Spellworks	29.89
Springboard Publisher	94.89
Star Fleet I (Flight)	33.89
Sticky Bear ABC	22.89
Car Builder	23.89
Drawing	22.89
Math 1 or 2	22.89
Math Word Prob.	22.89
Music	22.89
Numbers	22.89
Printer	22.89
Reading	22.89
Rd. Comprehension	22.89
Spellgrabber	22.89
Townbuilder	22.89
Typing	22.89
Summer Games I or II	24.89
Super Macroworks	32.89
Terrapin Logo	63.89
Think Tank	79.89
Thinkworks	59.89
Tip Disk #1	10.89
Toy Shop	38.89
Transylvania	13.89
Trinity	24.89
Triple-Dump	20.89
Turbo Database Tlbx	44.89
Turbo Pascal 3.0	44.89
Turbo Tutor	21.89
Type	28.89
Ultima III or IV	40.89
Understanding IIE	17.89
Utility City	15.49
Video Toolbox	23.89
Visable 6502	31.75
Wilderness	34.89
Wizardry	31.89
Wizard's Toolbox	23.89
Wizprint	16.89
Word Handler	36.89
Word Perfect	84.89
World's Grtst Ftball	24.89
Write Choice	24.89

COMMODORE	
Acrojet	22.89
Air Rescue I	24.89
Bard's Tale	30.89
Ballblazer	24.89
Back to Basics	134.89
Bank Street Storybook	27.89
Bank Street Speller	34.89
Bank Street Writer	34.89
Basic Toolkit	29.89
Blazing Paddles	24.89
Borrowed Time	20.89
Carmen Sandiego	22.89
Certificate Maker	29.89
Color Me	20.89
Cntdwn to Shtdwn	20.89
Fast Tracks	20.89
Gato	20.89
Gertrude's Secrets	20.89
G I Joe	24.89
Graphics Magician	27.89
Grt Amer Road Race	19.89
Hacker	20.89
Jet	27.89
Jet Combat Simulator	20.89
Karateka	22.89
Kennedy Approach	24.89
Kung Fu	20.89
Little Comp. People	24.89
Mail Order Monsters	24.89
Merlin	34.89
Mig Alley Ace	24.89
Multiplan 64/128	43.89
Nato Commander	22.89
Paper Clip	41.89
Print Shop	29.89
Print Shop Comp.	21.89
Pr. Shop Gr. 1,2,or 3	15.89
Rescue on Fractalus	24.89
Scrabble	27.89
Sideways	20.89
Silent Service	22.89
Spitfire Ace	20.89
Spreadsheet	34.89
Sticky Bear ABC	20.89
Numbers	20.89
Opposites	20.89
Shapes	20.89
Summer Games I or II	24.89
Temple of Apshai Tril.	24.89
Toy Shop	39.89
Transformers	24.89
World Games	24.89
World's Grtst Bsball	24.89
World's Grtst Ftball	24.89

IBM APPLE COMMODORE	
Ballyhoo	24.89
Clip Art (Vol 1 or 3)	17.89
Clip Art (Vol 2)	24.89
Copy II+	19.89
Crossword Magic	31.89
Crusade in Europe	24.89
Cutthroats	24.89
Early Games	20.89
Easy as ABC's	23.89
Enchanter	24.89
Family Roots	149.89
Fraction Factory	19.89
F-15 Strike Eagle	21.89
Graphics Expander	24.89
Ghost Busters	24.89
Hitchhiker's Guide	24.89
Leather Goddesses	24.89
Load Runner	22.89
Magic Spells	25.89
Mastertype	26.89
Math Blaster	27.89
Micro Lg. Baseball	24.89
Micro Lg. Manager	24.89
Micro Lg. Team	13.89
Mind Forever Voy.	24.89
Music Construction	12.89
Newsroom	34.89
One-On-One	12.89
Pitstop II	24.89
Planetfall	24.89
Reader Rabbit	24.89
Sargon III	31.89
Sat (Harcort Brace)	27.89
Sorcerer	27.89
Speed Reader II	38.89
Spellbreaker	30.89
Spell It	27.89
Starcross	30.89
Star Trek I or II	27.89
Suspect	27.89
Tass Times	24.89
Ultima III	40.89
Temple of Apshai Tri	24.89
Trinity	24.89
Typing Tutor III	31.89
Will Writer	28.89
Winter Games	24.89
Wishbringer	24.89
Witness	24.89
Word Attack	27.89
World's Grtst Bsball	24.89
Zork I	24.89
Zork II or III	27.89
Zork Trilogy	46.89

HARDWARE	
Compuserve Starter	24.89
Datacase	7.89
Disk Notcher	4.89
Flip-n-File	10.89
Lemon	27.89
Lime	48.89
APPLE HARDWARE	
80 Col. 64K Card (II E)	49.89
Apple Cat II	164.89
Disk Drive Cont	47.00
Disk Drive Hi Tech	119.89
Gibson Light Pen	148.00
Grappler Buffered	118.89
Grappler Pro	74.89
Grappler (serial)	74.89
Grappler C	74.89
Hotlink	44.89
Kraft Joystick IIE IIC	23.89
Koala Pad +	78.89
Laser 128 Computer	389.00
Mach II Joystick	28.89
Mach III Joystick	34.89
Mach IV Joystick	59.89
Micromodem IIE	139.89
Mockingboard A or B	64.89
Mockingboard C	114.89
Multiram CX 512K	174.00
Multiram IIE 80/64K	119.00
Mouse House	7.49
Mouse Mat	6.49
Paddlesticks	26.50
Parallel Printer Card	44.89
Prometheus 1200A	254.89
Print-it	122.89
Ramfactor 512K	219.89
Ramfactor 1 MEG	289.89
Ramworks 64K	139.89
Ramworks 512K	199.89
Ramworks 1 MEG	269.89
Super Serial Card	89.89
System Saver Fan	58.89
Thunder Clock	109.89
Wildcard II	74.89
ZEE 80A Checkmate	54.89
Zoom IIE	102.89
PRINTERS	
Citizen 120D	174.89
Star LV1210	189.89
Star NX-10	249.89
Printer Stand	16.89
Okidata 182P	229.89
Okimate 20	209.89
Printer Ribbons	CALL
Imagewriter Black	4.49
Imagewriter Color	4.99
Imagewriter II Color	11.99
Color Paper Pack	12.89
Print Shop Refill	12.89
MONITORS	
Magnavox Monitors	CALL
Amdek Color 600	414.99

SILICON EXPRESS

5955 E. Main St. Columbus, Ohio 43213

1-614-868-6868

SILICON EXPRESS

5955 E. Main St. Columbus, Ohio 43213

Name _____

Address _____

City _____ State _____ Zip _____

Charge # _____ Exp. Date _____

QTY.	DESCRIPTION	PRICE

Computer Type _____ SHIPPING

Phone No. _____ TOTAL

Add $3.00 min. U.S. shipping. C.O.D. $5.00 extra. Hawaii and Alaska $4.95 min. Orders outside U.S. are not insured. Canada & Mexico 10% min. $10.00. All other countries 25% min. $30.00. MasterCard, Visa and school purchase orders accepted. Personal checks allow 3 weeks. 5.5% sales tax for Ohio residents. Defective replaced within 20 days. 20% restocking fee if not replaced with same item. Compatibility not guaranteed. Prices subject to change without notice.

Defender Of The Crown For Amiga

Neil Randall

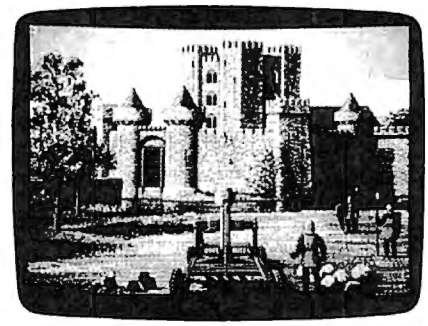
Requirements: Amiga with 512K minimum of memory.

Defender of the Crown is the Amiga introduction to Mindscape's new Cinemaware series. Labeled "interactive movies," the Cinemaware products attempt to capture the heroism of Hollywood in the form of a graphics adventure. *Defender of the Crown* places you in the role of an English noble in the swashbuckling world of Robin Hood, a world in which you lead armies, joust in tournaments, and rescue ladies.

By far the strongest feature of the Cinemaware products is the exceptional graphics. From its opening credits screen to the spectacle of the tournament, *Defender* makes full use of the Amiga's enormous graphics potential. The pictures almost leap off the monitor with clarity, color, and sheer beauty, and the animation is frequently good enough to achieve the game's goal of having the viewer feel part of a movie. The 3-D map of Britain, the pictures of the castles, the splendid portrayal of the tournament—all represent the finest artwork produced so far in a microcomputer game.

At various points in the story, for example, you set out to rescue a Saxon lady from the dastardly Normans. The game places your character in the courtyard of a castle, in which each brick on the wall and each star in the sky is clearly delineated. Your character, along with two allies, advances toward three defending swordsmen. If you win the sword fight, you move inside the castle, thrusting, slashing, and parrying your way up a flight of stone steps as the torches flicker on the walls behind you. Once past this guard, you enter the castle's interior, where the frightened damsel awaits you. The sequence places you, finally, in front of a fireplace, and you watch two silhouettes embrace and kiss. True to old Hollywood, the scene tastefully fades.

Unfortunately, *Defender*'s gameplay is less impressive. In one turn you can recruit your home army, build the campaign army, ride in search of conquest, besiege an enemy castle, or launch a raid against an enemy's possessions. You are constantly endangered by the armies of the other nobles, and as the game moves to a contest between you and one other noble, the opposing army can grow very large indeed. But where a good war game would use these medieval essentials as the basis for a thoughtful simulation,



Mindscape's Defender of the Crown *makes full use of the Amiga's graphics potential.*

Defender oversimplifies each step. Power politics, in the form of alliances and treachery, are nowhere to be seen, and battles offer nothing of the strategies and tactics of the period. Even the jousting and sword fighting, on which the game's action relies, are not particularly enjoyable. *Defender of the Crown* rests on its graphics, and this fact is both its strength and its weakness.

These criticisms are perhaps unfair, since *Defender* is not an attempt at historical simulation. But after the initial dazzle of the graphic excellence subsides a bit (it never leaves completely), the game isn't nearly as enjoyable as it could be. It is fun, and its historical flavor is undeniable, but it leaves me with a strong hope that future Cinemaware productions will complement the enormous graphic success with a more solid and replayable game. *Defender of the Crown* is a must for any Amiga owner, because with it you begin to understand your computer's capabilities. But it could have been even better. When superb graphics and superb game finally merge, as they surely will in a future Cinemaware product, computer gaming will never be the same again. For now, though, buy *Defender*. As Hollywood fantasy, it's as good as you can get.

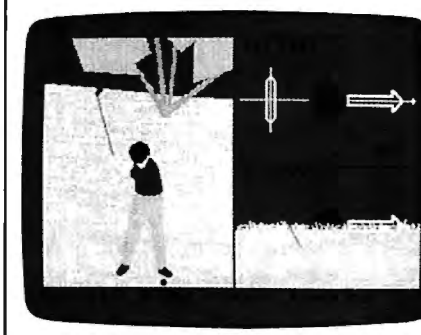
Defender of the Crown
Mindscape
3444 Dundee Rd.
Northbrook, IL 60062
$49.95

Championship Golf

Neil Randall

Requirements: Amiga (reviewed here), and IBM PC and compatible computers.

Among the sports games developed early in the Amiga's life cycle, golfing simulations seem to dominate. *Leader Board* and *Mean 18* were first off the tee, each with its own touches of excellence and each gaining its own following. Now into the race comes Gamestar (a subsidiary of Activision), whose *Star League Baseball* and *On-Court Tennis* showed Commodore 64 and eight-bit Atari owners what their machines were capable of. *Championship Golf* for the Amiga goes a long way toward demonstrating the kind of simulations we can begin to expect on the new, higher-powered personal computers.



Graphics, animation, and excellent playability recommend Gamestar's Championship Golf *(PC version shown here).*

Remarkable Graphics

Championship Golf uses a split screen to simulate the intricacies of the famous Pebble Beach course. On the left is an overhead shot of the hole you are playing, while the right side shows the same hole from the golfer's ground-level point of view. Both views are variable, though; the function keys let you zoom in on the overhead screen and step back to a bird's-eye view on the other. The graphics detail on both screens is remarkable, but especially interesting are the fractal techniques used in the ground-level view. Hills and slopes are all visible, and the complexity of the course becomes obvious. In fact, you can position the golfer anywhere you like on the hole, and the graphics will be redrawn to show your new perspective. In this way, you can walk the course.

Once you've aimed your shot, the left screen changes to allow you to choose the club you will use and the

type of shot you will take (full swing, chip, or putt). For all shots, the computer acts as your caddy, suggesting club and shot type. After this, you set your stance and the alignment of the clubface, working with the wind to hook or slice your shot. Finally, the golfer appears, ready to swing away (under your control). The graphics are superb here, with the golfer's animation clear and accurate right down to wrist movement, and once the shot is taken, you watch the ball in the ground-level screen. All in all, the graphics are fully worthy of the Amiga's capabilities.

In addition to addressing stance and clubface alignment, you have the option of adding arm speed, wrist action, and body action as you take the shot. As in real golf, these activities must be learned and practiced, and also as in real golf, success makes you feel extremely good. After some initial ineptness in which you watch shot after shot dribble off the tee, you will suddenly see the golfer get under the ball perfectly, and you will watch the ball disappear above the height of the screen and land farther down the fairway than you've managed so far. With the right amount of practice, you can make these shots fairly regularly.

Game Options

To help you perfect your swing, *Championship Golf* includes a driving range. Here you can practice teeing off, hitting from the rough or off the fairway, chipping out of two types of bunkers, or putting on a practice green. The driving range is a must for those wanting to become expert players, even those experienced in real golf. While *Championship Golf* seems to be realistic, it is inevitably different from actually swinging a golf club.

The game offers several other options. You can hit from three different tee placements, you can carry clubs not normally used (2-wood and 1-iron, for example), and you can save the current game to disk. Since the full Pebble Beach course takes a fairly long time to play through, saving is extremely convenient. For those who wish to see how the game works, pressing RETURN and F10 through the shot sequence will provide a perfectly respectable, but hardly awe-inspiring, round.

Championship Golf is subtitled *Volume One: Pebble Beach.* The implication, clearly, is that Gamestar intends to issue other volumes, each with its own course. If each course is as detailed as Pebble Beach, I can easily imagine a golfing fan happily buying every volume.

The system's only drawbacks are the amount of disk access, the time taken to redraw the ground-level views, and the strangely undetailed putting game. The gamemakers could help disk access by allowing those with extra memory to load parts of the program into RAM; they could also improve the putting. Whether or not Gamestar can speed up the redrawing is relatively unimportant; the ground-level and bird's-eye views are spectacular, and well worth the wait. *Championship Golf* is indeed a very good game.

Championship Golf
Gamestar
P.O. Box 7287
Mountain View, CA 94039
$39.95 Amiga and MS-DOS versions

Super Hi-Res Graphics And Sound On The Apple IIGS

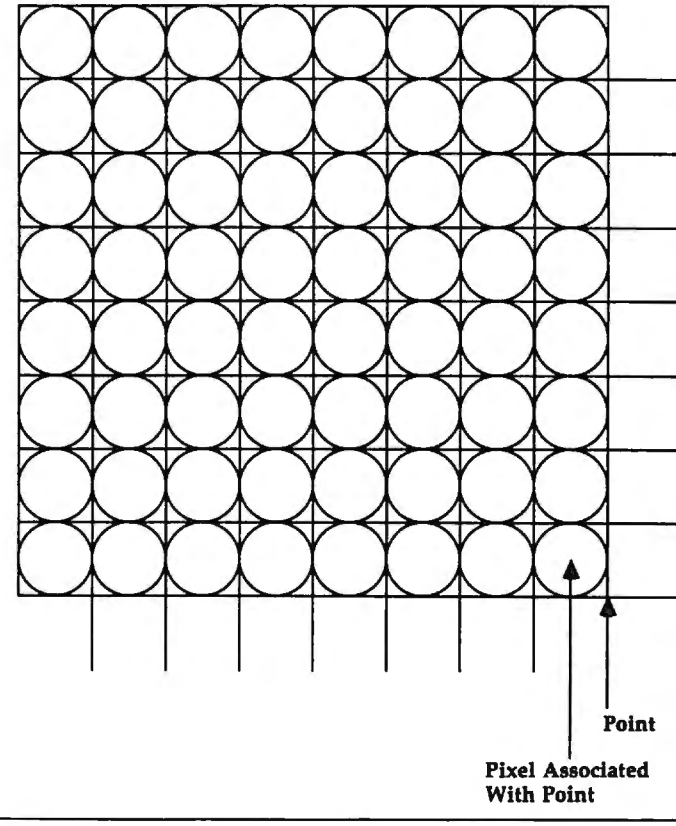
William B. Sanders

The IIGS, the newest computer in the Apple II line, is aptly named. Its superb graphics and sound—hence, GS—make this computer the most exciting machine in the Apple II line. Commercial software developers are moving quickly to take advantage of these features. For amateur programmers, the GS also opens up new areas to explore. Here's an overview of the IIGS's amazing super high-resolution color graphics and powerful sound capabilities, excerpted from William B. Sanders' recently published book, The Elementary Apple IIGS, *from COMPUTE! Books.*

Your IIGS has made a quantum leap over the other Apple IIs when it comes to graphics and sound. The reason is that it can place pixels on a 640 × 200–dot matrix for four-color programs to gain the highest resolution on an Apple II yet. And it has an Ensoniq sound chip for creating unsurpassed digitized sound. There are some excellent programs available for accessing these new features, but you cannot easily work with them directly from Applesoft BASIC. Applesoft BASIC remains the same as it has always been, so the thousands of programs written in it can also run on the IIGS.

Apple has made it simple for advanced programmers to use the super high-resolution graphics and sound. However, because of the advanced skills required for accessing the "toolboxes," we'll introduce just the concepts here so that you can get an idea of whether you're interested in pursuing the advanced programming levels required to access the IIGS's super high resolution and sound. Remember, there are application programs available for you to do all of this without any programming skills at all. However, you might find doing the programming yourself to be a real adventure.

Figure 1: Pixel Matrix



The QuickDraw II Toolbox: Super Hi-Res Graphics

The collective set of the subroutines built into the IIGS ROM have been named QuickDraw II. By making *calls* to these routines, you can create lines, rectangles, polygons, arcs, and other figures. With the toolbox routines in QuickDraw II, much of the work has been done for you.

The basic building block of the routines is the pixel matrix made up of points and associated pixels. Each point referenced through the toolbox routines has an associated pixel above and to the left of the point. A pattern is composed of an 8 × 8 matrix of points and pixels (Figure 1).

Depending on the values associated with a given pixel matrix, different patterns, shapes, and figures can be created. The boundaries of the drawings created in QuickDraw II are (−16384,−16384) and (16383,16383) on a Cartesian matrix with memory space of 32768 × 32768 pixels. Various calls made in assembly language, C, or another language that can easily access the QuickDraw II tools are used. The calls are given various names associated with addresses in memory. Let's look at some of these calls to get an idea of what is available in the graphics toolbox inside your IIGS.

QuickDraw II Calls

The first set of functions in the QuickDraw II toolbox is intended for housekeeping purposes. These functions set up the various registers and pointers to allow access to the graphics tools. They include QDBootInit, which initializes the QuickDraw II tools when the system is booted; QDStartup, which initializes QuickDraw II and sets the standard port and clears the screen; and QDShutDown, which turns QuickDraw II off and frees the buffers. QDVersion and QDStatus, respectively, provide information on the version of QuickDraw II and specify whether or not it is active.

A second set of QuickDraw II routines allows you to control the characteristics of the screen display. Each of the 200 horizontal *scan lines* that make make up the display has its own scan-line control byte (SCB), so the characteristics of each line can be specified independently. The GetStandardSCB routine returns information about the SCB. The first four bits (0–3) are used for color table 0; bit 4 is reserved; bit 5 controls the fill option; bit 6 specifies whether an interrupt can occur; and bit 7 controls the color mode (320 pixels or 640 pixels). The call SetMasterSCB sets the low byte of the master SCB, and GetMasterSCB returns the same information. SetSCB, GetSCB, and SetAllSCBs are further scan-line control-byte calls. For setting the color table, either in the 320 or 640 mode, the InitColorTable call is used. Table 1 lists the values (in hexadecimal) for the two modes.

Calls to SetColorTable, GetColorTable, SetColorEntry, and GetColorEntry all access the routines to set and get information about the colors.

The other global calls deal with the fonts, clearing the screen, and turning the super hi-res graphics mode on and off. The calls include SetSysFont, GetSysFont, ClearScreen, GrafOn, and GrafOff. The functions of the calls are fairly self-explanatory, making it easier to use the graphics than to use more obtuse codes.

Port Of Calls

Now you have some idea of a few of the functions and how the name of the call is connected to the functions. For the rest of the QuickDraw II functions, refer to Table 2, just to

Table 1: Hexadecimal Values For Setting Color Table

320 Mode			640 Mode		
Pixel	Color	Code	Pixel	Color	Code
$0	Black	0 0 0	$0	Black	0 0 0
$1	Dark gray	7 7 7	$1	Red	F 0 0
$2	Brown	8 4 1	$2	Green	0 F 0
$3	Purple	7 2 C	$3	White	F F F
$4	Blue	0 0 F	$4	Black	0 0 0
$5	Dark green	0 8 0	$5	Blue	0 0 F
$6	Orange	F 7 0	$6	Yellow	F F 0
$7	Red	D 0 0	$7	White	F F F
$8	Flesh	F A 9	$8	Black	0 0 0
$9	Yellow	F F 0	$9	Red	F 0 0
$A	Green	0 E 0	$A	Green	0 F 0
$B	Light blue	4 D F	$B	White	F F F
$C	Lilac	D A F	$C	Black	0 0 0
$D	Periwinkle blue	7 8 F	$D	Blue	0 0 F
$E	Light gray	C C C	$E	Yellow	F F 0
			$F	White	F F F

Table 2: GrafPort Calls

OpenPort	InitPort	ClosePort	SetPort	GetPort
SetPortLoc	GetPortLoc	SetPortRect	GetPortRect	SetPortSize
MovePortTo	SetOrigin	SetClip	GetClip	ClipRect
HidePen	ShowPen	GetPen	SetPenState	GetPenState
SetPenSize	GetPensize	SetPenMode	GetPenMode	SetPenPat
GetPenPat	SetSolidPenPat	SetPenMask	GetPenMask	SetBackPat
GetBackPat	SetSolidBackPat	SolidPattern	PenNormal	MoveTo
Move	SetFont	GetFont	SetFontID	GetFontID
GetFontInfo	GetFGSize	GetFontGlobals	SetFontFlags	GetFontFlags
SetTextFace	GetTextFace	SetTextMode	GetTextMode	SetSpaceExtra
GetSpaceExtra	SetCharExtra	GetSpaceExtra	SetForeColor	GetForeColor
SetBackColor	GetBackColor	SetBufDims	ForceBufDims	SaveBufDims
RestoreBufDims	SetClipHandle	GetClipHandle	SetVisRgn	GetVisRgn
SetVisHandle	GetVisHandle	SetPicSave	GetPicSave	SetRgnSave
GetRgnSave	SetPolySave	GetPolySave	SetGrafProcs	GetGrafProcs
SetUserField	GetUserField	SetSysField	GetSysField	

see the extent of the toolbox. Further description here would do little good, since you may not yet have the programming skills necessary to use the calls.

Drawing Calls

The table below will give you a better idea of what kinds of drawing shapes are supported by the QuickDraw II toolbox. This set of calls is what most programmers will use often when they're creating graphics directly or when they're writing a graphics drawing program.

By looking over these calls from the QuickDraw II toolbox, you can become acquainted with what is available and get a clue as to the graphics power of your Apple IIGS computer.

The Sound Of The IIGs

The Apple IIGS houses a 5503 Ensoniq Digital Oscillator Chip (DOC). The DOC has 32 digital oscillators that give you everything from beeps and buzzes to a talking computer and symphonic orchestra. However, as with the super high-resolution graphics, you must use the DOC toolbox to take full advantage of this feature.

To get started, let's look quickly at the registers used to control the sounds in DOC.

Frequency control (low and high). Two registers control frequency; joined together they form a 16-bit value used for the 24-bit accumulator. The value of this register pair is added to the current value stored in the 24-bit accumulator.

Address: $00–$1F (low)
$20–$3F (high)

Volume. This register set controls the volume level of the sound created.

Address: $40–$5F

Waveform data sample. This reads the last value from the waveform table.

Address: $60–$7F

Address pointer. These registers are used to determine where in RAM the waveform tables are located. Each waveform table begins with the first address of a page and must continue upward through RAM and cannot wrap around over 64K. The register keeps track of where the table ends.

Address: $80–9F

Control register. Channel assignment, oscillator mode, and halt bit are all controlled by this register. Bits 4–7 make up the channel assignment. Those four bits can assign up to 16 channels for sound. Bit 3 is the interrupt enable used for ordering output when more than a single oscillator has generated output. It helps keep all the different sounds organized. Bits 1 and 2 set the oscillating mode for each oscillator, and bit 0 is the halt bit indicating when an oscillator has been stopped by the microprocessor or DOC.

Address: $A0–$BF

Bank select/resolution/waveform registers. Each register uses seven bits for controlling three major functions (bit 7 is not used). Bit 6 determines whether the DOC address range is 0–64K (0) or 65–128K (1). Bits 3–5 specify the size of the waveform table, ranging from 256 bytes to 32K bytes. Finally, bits 0–2, called the *resolution determination bits*, actually determine the final address for the waveform table.

Address: $C0–$DF

Table 3: Drawing Calls

Lines

LineTo	Line			

Rectangles

FrameRect	PaintRect	EraseRect	InvertRect	FillRect

Regions

FrameRgn	PaintRgn	EraseRgn	InvertRgn	FillRgn

Polygons

FramePoly	PaintPoly	ErasePoly	InvertPoly	FillPoly

Ovals

FrameOval	PaintOval	EraseOval	InvertOval	FillOval

Rounded-Corner Rectangles

FrameRRect	PaintRRect	EraseRRect	InvertRRect	FillRRect

Arcs

FrameArc	PaintArc	EraseArc	InvertArc	FillArc

Pixel Transfers

ScrollRect	PaintPixels	PPToPort		

Text Drawing and Measuring

DrawChar	DrawText	DrawString	DrawCString	CharWidth
TextWidth	StringWidth	CStringWidth	CharBounds	TextBounds
StringBounds	CStringBounds			

Mapping and Scaling Utilities

MapPt	MapRect	MapRgn	MapPoly	ScalePt

Miscellaneous Utilities

Rectangle Calculations

SetRect OffsetRect	InsetRect	SectRect	UnionRect
PtInRect	Pt2Rect	EqualRect	EmptyRect

Point Calculations

AddPt	SubPt	SetPt	EqualPt
LocalToGlobal			
GlobalToLocal			

Region Calculations

NewRgn	DisposeRgn	CopyRgn	SetEmptyRgn
SetRectRgn	RectRgn	OpenRgn	CloseRgn
OffsetRgn			
InsetRgn	SectRgn	UnionRgn	DiffRgn
XorRgn			
PtInRgn	RectInRgn	EqualRgn	EmptyRgn

Polygon Calculations

OpenPoly	ClosePoly	KillPoly	OffsetPoly

Other

Random	SetRandSeed	GetPixel

Oscillator interrupt, oscillator enable, and A/D converter registers. These three registers *(not* bits) control the oscillators and analog-to-digital conversion.

Address: $E0–$E2

Sound Tools

From the above brief description of the sound registers and the digital oscillator chip, you can see that it's not simple to crank up the kinds of sound heard in musical demonstrations on the IIGS. As an aid to programmers, the sound tools have been provided. There are 18 sound function calls and six low-level routines for accessing the power of DOC. The sound toolkit works through a sound tool set with a specified number. The tool locator finds this number in order to use the sound tools. Again, this requires a higher level of programming skills than you may now possess, but to give you an idea of what's in the sound tool set, the calls in Table 4 are available.

Table 4: Sound Tool Calls

Function Calls

SoundBootInit	SoundStartup
SoundShutdonw	SoundVersion
SoundReset	SoundToolStatus
WriteRamBlock	ReadRamBlock
GetTableAddress	GetSoundVolume
SetSoundVolume	FFStartSound
FFStopSound	FFSoundStatus
FFGeneratorStatus	SetSoundMIRQV
SetUserSoundIRQV	FFSoundDoneStatus

Low-Level Routines

Read Register	Write Register
Read Ram	Write Ram
Read Next	Write Next

For some of you, it may be frustrating not to be able to program sound and super high-resolution graphics on your IIGS with what you now know about programming, but be patient. You will learn the more advanced techniques in time. Books and programming utilities will be available in the future to help you. ©

Label Master

This Commodore 64 program from the February issue (p. 56) and on the COMPUTE! Disk for February–April works properly on very old 64s with version 1 ROMs and on newer 64s with version 3 ROMs—including the 64C and the 128 in 64 mode. However, the graphic-design grid doesn't appear on older 64s with version 2 ROMs. To check which ROM version your 64 has, enter PRINT PEEK(65408). If the value returned is 0, your ROMs are version 2. (A value of 170 indicates version 1, and a value of 3 indicates version 3.)

When the screen is cleared, version 2 fills color memory with the screen background color. Since "Label Master" makes no provision for changing color memory, characters stored into screen memory are the same color as the background, and hence are invisible. The following patches modify this program to work with version 2 ROMs. To make the changes, first load a complete copy of Label Master into "MLX" using the starting and ending addresses given in the article. Next, select the (E)nter data option, specify an address of **0801**, and enter the following lines:

```
0801:0B 08 0A 00 9E 32 30 37 2F
0809:37 00 00 00 A9 01 8D 21 42
0811:D0 A9 93 20 D2 FF A9 00 52
0819:8D 21 D0 60 A9 02 85 FD B6
0821:8D 20 D0 A9 50 85 FB A9 EE
0829:04 85 FC 18 20 0D 08 A0 A3
```

Then change the entry address to **09E9** and enter this line:

```
09E9:00 20 0D 08 A0 00 B9 FD 9C
```

Change the entry address to **0AF1** and enter this line:

```
0AF1:00 20 0D 08 20 7B 0B A9 DE
```

Finally, save a copy of the modified Label Master data before exiting from MLX.

Menu Planner For The Commodore 64

The 64 version of this program from the April 1987 issue (p. 60) has a number of mistakes in the routine which sends menu lists to a printer. To properly print menus, change the following lines:

```
610 PRINT"MENU:":IFA$="P"THENO
    PEN4,4:CMD4:PRINTSPC(45)"ON
     HAND{4 SPACES}NEED TO BUY"
640 IFA$="R"THEN GETA$:IFA$=""
    THEN640
660 IFA$="P"THENPRINT#4:CLOSE4
```

Superplotter

Line 940 of this Commodore 64 graphics program in the March 1987 issue (p. 120) includes the delete ({DEL}) character, which can't be typed in the usual way. To produce this character, use the cursor keys to move the cursor to the spot in the line where the {DEL} is to be inserted, use the SHIFT–INST/DEL key combination to insert a space, and then press INST/DEL again (without SHIFT). The delete character will appear as a reverse-video *T*. ©

A Nation Of Thieves: Responses To Readers

Last October I wrote what, in my view anyway, was a fairly calmly written piece on the illicit copying of software. I mentioned the impact that illicit copying had on software companies and on those of us who choose to obey federal law on this topic. Based on letters I received, one would have thought that I was attacking motherhood!

I knew I was in trouble the day I went to my post office box and all I saw was a pink card instructing me to pick up my mail at the counter. Over the weeks following the publication of that article, my accumulated mail on this topic filled a #3 mailbag. People wrote from as far away as Sweden to challenge my position.

Of the letters received, only two readers wrote to thank me for sensitizing them to a major industry problem—and, no, these people were not mouthpieces for the software companies. I decided to read each letter. Some readers got sidetracked from the issue and attacked me personally, at which point their letters were (generally) discarded.

I placed the remainder of the letters in several piles based on the general themes of the arguments presented. The remainder of this article summarizes the main arguments against my position and includes my responses.

Letter writers are identified only by initials for two reasons. First, some of you admitted to acts that are punishable by incarceration in a federal penetentiary, and I felt it to be in your best interest if I excerpted what I needed from your letter and shredded the rest. Second, there was not a great deal of novelty in the responses, so any point one reader might have made was also made by others.

I also have taken the liberty to change some of the language of the arguments so they can be printed in a family magazine.

The "Service To Society" Argument

"Why is it that software publishers seem to think that they are any different than the guy who writes some worthwhile book? For centuries the world has known that authors are generally underpaid and underappreciated, and yet there is no great outcry in their favor.

Who has more intellectual power invested: The guy who writes a straightforward spreadsheet such as Lotus 1-2-3 *or the guy who writes a book such as* War and Peace? *I would suggest it is the latter."*—DB

Well, DB, everyone I know who writes a worthwhile book expects to get paid for his or her efforts, and usually does. Yes, history is full of examples of authors who didn't get rich from their efforts, but there are plenty that do. I know a science fiction author who gets $1,000,000 advances for his books, and his case is not that uncommon. If you write a book that becomes a best seller, you bring in big bucks, period. Fortunately, books are expensive to copy—but programs aren't, so software gets ripped off.

Your second question has a very simple answer, the "guy" (actually an entire company) that wrote *Lotus 1-2-3* has a lot more energy invested than Tolstoy ever put into *War and Peace.* For example, Tolstoy's hand-written drafts had lots of mistakes, some of which made it into the published version of the book. The reviews of his work never mentioned the misspelled word on page 315 or the awkward phrase on page 74. "Bugs" of this nature cause havoc in a computer program, but are perfectly acceptable in a book. The consequences of typing errors in books is benign—you may wince (depending on your fluency in Russian), but you forgive Tolstoy all the same.

This is not the case with computer software. A typing error may result in a subtle bug that takes weeks to fix. In the meantime, the users keep complaining and requesting a free upgrade once the bugs are fixed. I agree that free upgrades are appropriate for software, and I do not expect publishers to send revised editions of books to their old customers at no cost—even though the parallel that some readers imply would require such service.

No, the point is that the publishing of software is quite different from the publishing of books. I know this because I do both, and books, comparatively, are a snap to publish!

Let's pretend that Tolstoy had to follow the path of a software publisher when he wrote his epic novel. First, he would have had to create several releases of his book to repair his "bugs." Then, he would have to immediately start working on his next book, knowing that if he didn't improve on his product, others would. Most of his profits would have to be placed into this new venture.

Compared to software authors, book authors have it made.

The "Mouthpiece For Big Business" Argument

Now that you former hackers have hit the big time, you've developed vested interests in protecting the vendors to the detriment of the consumers. The real thieves are the companies that can charge outrageous prices for programs that are guaranteed not to work. What other industry do you know that will sell a product and not make full refunds on defective merchandise?

Your blanket statement that copying software is illegal is tanta-

mount to saying that no one has the right to free speech.—WM

You bet we have vested interests, just like the kid who was willing to play sandlot ball for free, but who now gets hundreds of thousands of dollars a year as a pro. I'm sorry, but this argument just doesn't work. I've never seen a software company put out a product guaranteed not to work; nor have I encountered a company that was unwilling to refund a customer's money when the product was returned in a timely manner with documentation to support its return. I say this even though there are some people who will buy a product, make a copy, and then return the product because "it doesn't meet their needs."

As for my blanket statement, I was very careful to level my criticism at those who copy to secure software that they haven't purchased. Backup copies, media conversion, and so forth, are all within the domain of reasonable copying, and do not (to my knowledge) violate any laws.

The "Software Is Lousy, So Why Buy It?" Argument

The list of lousy software that readers came up with was quite large. A typical comment went like this:

You bet I have an illegal copy of XXX; it is a piece of trash. For example, every time I boot the program I have to enter a bunch of preferences all over again. Why didn't the developer let you save your preferences on disk? It wouldn't matter to me if I used the program infrequently, but I use it every day and this is a pain in the tail.

This comment—synthesized from several letters—presents an interesting definition of trash. Every morning I empty the trash so it won't smell up the house. Trash is something you throw away, not something you "use every day." The "trash" argument is just a thinly veiled disguise for the feeling of guilt that these readers must be experiencing.

By the way, "legal" users of your programs have probably received upgrades that fix the problems you mentioned in your letters.

The Real Issue

Behind the anger expressed in many of your letters lies a real issue that needs to be resolved. Many of you feel that software vendors aren't responsive to your legitimate needs. The products don't live up to expectations, the documentation is full of errors, the product's function doesn't justify its cost, and so on. All of these are legitimate complaints that should be addressed to the software publishers. By communicating with these people directly, you stand an excellent chance of seeing your ideas incorporated in new releases of the product. I always send free copies of my programs to people whose suggestions I have incorporated, and this is not an uncommon practice in this industry.

If you feel that the software vendors aren't listening, maybe this is because you aren't speaking to them. By sneaking around behind the vendor's back and stealing software, you are losing any chance you ever had to make sure that the programs you really want will ever make it to the marketplace.

Dr. Thornburg's most recent product is Calliope™, *a "non-linear" idea processor for the Apple IIe, c, GS, and Macintosh computers. He welcomes letters from readers and can be reached in care of this magazine.* ©

COMPUTE!'s Gazette is looking for utilities, games, applications, educational programs, and tutorial articles. If you've created a program that you think other readers might enjoy or find useful, send it, on tape or disk, to:

Submissions Reviewer
COMPUTE! Publications
P.O. Box 5406
Greensboro, NC 27403

Please enclose an SASE if you wish to have the materials returned.
Articles are reviewed within four weeks of submission.

Arlan R. Levitan

Lapping It Up

The face of mobile telecomputing is changing. I've owned a Tandy 100 laptop for several years. When first introduced, the Model 100 significantly expanded the freedom of a generation of journalists. Its portability and built-in 300-bps modem made it possible to write and transmit a story on the spot. A new wave of battery-powered PC-compatible laptop machines—which overcome the small display and memory limitations of the Model 100 with full-sized screens and 640K of memory—is swelling the ranks of mobile telecomputers. If you've been thinking of joining them, here's some friendly advice.

Most PC laptops offer 300-/1200-bps internal modems as an option. They are somewhat pricey, listing for $400 or so, and place an additional drain on the laptop's batteries, shortening the amount of usable time between charges. Compact and attractively priced, battery-powered modems are viable alternatives to internal laptop units. Most notable at this time are Touchbase System's WorldPort 1200 ($199) and Migent's Pocket Modem ($259). Both are compatible with the Hayes command set; about the size of a pack of cigarettes, they plug directly into any computer's serial port. They can be powered from an internal battery when you're on the go, or from an AC adapter when an electrical outlet is handy. A pocket modem is a good choice if you are on a limited budget or want to use your modem with more than one computer. If you're completely immersed in the portability gestalt, an internal modem means one less item to lug around.

When I'm on the run, treating work gear gingerly is usually the last thing on my mind. Since unceremonious dunks and tosses into car trunks are a fact of life for telecomputing on the hoof, a suitable carrying case is a must. Most of the carrying cases offered by the laptop manufacturers themselves offer little real protection and stubbornly refuse to acknowledge the fact that you may want to carry anything other than your computer. More often then not, well-padded cases designed for photographic equipment can be pressed into laptop service with little or no compromise. Fans of "Miami Vice" will opt for the polished aluminum, high-tech look of Halliburton Zero cases. Since my trusty Volvo is decidedly less flashy than a Ferrari, I went for a sedate nylon LowePro Trimtech photographer's case that has enough padding to insure survivability in environments short of a direct nuclear strike; plenty of room for accessories, business papers, and manuals; and a fold-down "office" for storing disks, writing implements, business cards, and other goodies.

You Can't Get There From Here

Then there's the problem of moving data between your laptop and desk machine. Most desktop PCs are equipped with 5¼-inch disks, while the current breed of laptops use 3½-inch microdisks. In order to bridge the gap, many laptops offer external 5¼-inch disk drives as optional accessories, and a 3½-inch disk drive can be added to almost any PC compatible. Adding an internal microdisk to your PC is not too expensive (about $200), but limits your data transfers to wherever your desktop computer sits. An external drive for a laptop is more expensive (about $400) and saddles you with another relatively bulky piece of equipment to lug around. Remember, in the laptop world, as in Bauhaus architecture and California politics, "Less is more."

The humble standard RS-232 serial port found on almost all PCs offers efficient, reasonably priced alternatives for laptop-to-PC file transfers. Using standard public domain or "shareware" telecommunications software and a null modem cable lets you move files between machines at speeds of up to 19,200 bits per second. White Crane System's *Brooklyn Bridge* software ($129.95) makes file movement between laptop and desktop PCs even more of a snap. *Bridge* lets either machine access the other's drives directly, using standard DOS commands. The special drivers supplied allow data to move between systems at 115,000 bps, which is pretty close to the maximum transfer speed rate of laptop microdisk drives.

Padded Cells

Where will this mobile telecomputing madness end? Within the next few weeks, I'll be shopping for a cellular mobile phone that will allow hookup to my NEC MultiSpeed's internal modem. Will life ever be the same? Will online teleconferencing and stock quotations replace such classic highway games as "Geography," "License Plate Lotto," and endless choruses of a "Hundred Bottles of Beer on the Wall?" Stay tuned. ©

Microscope

Sheldon Leemon

In the music industry, compact discs are riding high. When the shiny silver platters were introduced a couple of years ago, players cost over $1,000, and there were only a small number of albums to play on them. Now it's possible to buy CD players for as little as $100, and a vast library of music is available on CDs.

CDs apply digital technology to the recording of music. Sounds are translated into binary numbers composed of 0's and 1's, and stored as a series of high and low spots on a disc that can be read by a laser beam. The same digital principles are used by microcomputers, which translate data such as text characters into a series of binary numbers. So CDs seem an even more natural medium for storing computer data than for storing music.

So far, Microsoft, the software giant, has lead the way in promoting standards for using CDs as computer storage devices. This technology is known as CD-ROM (Compact Disc–Read Only Memory), though CROD (Compact Read Only Disc) would be more accurate. Each tiny removable disc holds about 500 megabytes of data, enough to store over 100,000 pages of text. Since discs are stamped out like records, that vast quantity of information can be duplicated in a fraction of a second, at a cost comparable to that of floppy disks. Even though there's currently no way to erase and write over a CD as you can with a floppy disk, the computer industry sees CD-ROMs as an inexpensive way to provide personal computers with access to vast reference libraries. It's possible to put all of the volumes of an encyclopedia on one CD, along with an index of every word used, so that every occurrence of a word or phrase can be looked up in seconds. In fact, the text of the encyclopedia would only take up a part of the space available, leaving a lot of room for other books as well.

****

In March, Microsoft held its second annual CD-ROM Conference. Though there are still almost no commercial CD-ROM products available (particularly at the consumer level), there were over 1200 attendees and exhibitors at the conference, which indicates the level of interest in this technology. At the conference, Microsoft unveiled *Bookshelf*, a $295 package made up of a memory-resident IBM PC program and a CD containing ten reference works. Included are a dictionary, a thesaurus, a style guide, *Bartlett's Quotations*, a 1987 almanac, and a ZIP code directory. The PC program is designed to work with any word processor. You could, for example, use it to look up a ZIP code while writing a business letter, and have it insert the ZIP code into the text of the letter at the appropriate place. Microsoft also announced at the conference that it would bundle *Bookshelf* with Amdek's Laserdrive I, an external CD-ROM player that can also play audio disks. The $1,099 package includes a PC drive controller and Microsoft's MS-DOS CD-ROM extensions, which make it possible for PC programs to read data from CDs much as they do from ordinary disks.

So far, the applications for CD-ROM technology have been very specialized and expensive. Lotus has been offering a financial database called *One Source*, for example, that contains detailed financial information about publicly held companies, and has announced that it will update that information weekly (for a fat fee). It's also possible to get *Books in Print*, the standard reference for booksellers, on a single CD, at a cost of about $800. Information vendors feel that they must charge such high prices because there's a relatively small number of CD-ROM owners to whom they can sell their wares. But until there's more reasonably priced software available, few CD-ROM players will be sold. By bundling *Bookshelf* with players, Microsoft hopes to get the ball rolling on both the hardware and software side.

****

As exciting as CD-ROM is, it's not the only computer-controlled CD technology. Standards have already been established for an interactive audio disc technology called CD-I (Compact Disc–Interactive). And at the Microsoft conference, RCA put on a demonstration of what it calls DVI (Digital Video Interactive). This new technology packs as much as 72 minutes of high-quality full-motion video and audio on a single disc. The maximum video resolution of this system is an excellent 768 × 512 pixels, with up to 16 million colors per pixel. Such interactive video technology could be used for educational materials, videogames, and a host of applications that have been barely dreamed of as yet.

So far, CD-ROM has not visibly benefited from the success of its audio counterpart. Since CD-ROM requires much higher accuracy than music CD players afford, the plummeting price of music CD players has not brought a corresponding drop in the price of CD-ROMs. Atari has been forced to delay plans for a $500 player for its ST computer because the company has been unable to meet the target price. But the spread of music CD technology, and especially the expansion of disc-production capacity, should work to the advantage of CD-ROM in the long run. ©

Instant Images On Your Apple Computer

As part of our Multi-Media Classrooms Project here in Birmingham, Alabama, I've had the opportunity to play with an exciting new product, the FingerPrint Plus Card from Thirdware Computer Products (4747 N.W. 72nd Ave., Miami, FL 33166). This product has a unique appendage—a little ribbon cable and an activator tab with the picture of a fingerprint on it. The tab projects from the front of your Apple IIe, II+, II, or IIGS computer. When you press it, you have an instant snapshot of your current computer screen.

The FingerPrint Plus card (with installation, slide-show software, and activator tab) normally retails for $149. However, schools get a 35 percent discount. And, if it's the first FingerPrint Plus for a school, it costs only $75.

When I received the card from Thirdware, I installed it in slot #1 in my Apple IIe. The card comes with interfaces for parallel and serial printers, but it's a good idea to make sure you have the right cable (parallel or serial) for your printer. I tested the card on several brands of software, including *Bank Street Writer*, *Print Shop*, *Apple Access*, and *Magic Slate*, and I had no printing problems.

Looking at the documentation, I learned that the card can also double as a serial modem card. All you need is a special FingerPrint Plus-to-modem cable, and you need to make some dip-switch changes. The FingerPrint Card also can turn your Apple into a fancy typewriter. Choosing the Type option on the main menu screen allows you to type directly onto your printer.

Easy As Pie

Capturing a computer screen is easy. You never have to leave your software or issue any special commands. When you see a screen you like—even if it's visible momentarily—you press the fingerprint tab. The screen vanishes, and in its place is a full-screen menu of FingerPrint Plus functions. At this point, if you press the Return key, a copy of the screen is sent to your printer. Or you can choose Display and take another peek at the screen to make sure it's the one you wanted. When you return to the menu (by pressing the fingerprint tab), you can choose Disk on the menu and save your screen image to disk.

After you've saved several screen images to disk, you can boot up the FingerPrint Plus disk and create a custom-made slide show. You can present the slides in any order and select the time each slide remains on the screen during the show (a minimum of ten seconds per slide).

Back in the main menu, there are several other things you can do to manipulate your captured screen image. You can select the number of copies of the image you want printed. Page numbers and headings can be inserted on the pages. You can invert the computer's screen image and print the inverted image. You can enlarge the image to twice its normal size, rotate it 90 degrees, or select just a portion of the image for printing. Finally, you can manipulate the colors in your image when it is printed out.

Some minor caveats: When I got my first FingerPrint Plus card, it caused my Imagewriter II to mysteriously jump columns when printing hi-res screens. A second card suffered from the same problem. I suspect that the problem was caused by the Imagewriter II's 32K memory option, because when I tested the same cards on Imagewriters without the option, the card worked perfectly.

Also, when I created my first slide show, the screen cursor behaved oddly and began scrolling my menu of slides off the screen. However, when I took the COMPUTEREYES/2 video digitizer card out of slot 3 in my Apple IIe and put it into slot 7, the cursor again became well-behaved.

Many Applications

It's a lot of fun to press the fingerprint tab and watch a computer screen instantly print out. But what are some practical uses for a FingerPrint Plus?

- Create custom-made quick-start cards and training manuals for your students or other new software users. You can pick key screens in the software, take FingerPrint snapshots, and use the screens to illustrate your training instructions.
- Create activity worksheets for your students. Lots of educational programs have graphics screens which illustrate math, science, social studies, and language arts concepts. You can capture the screens, duplicate them, and use them as worksheets for your students to do at their desks.
- Capture key screens, organize a slide show, and either present the show on a large-screen monitor or video projector, or send your computer's video output to a VCR and capture it on videotape. You'd then have a video outline illustrating key parts of your software or illustrations for your presentation on a given subject.
- Print out black-and-white screen dumps onto transparencies on your printer (using a carbon plastic ribbon) and show the transparencies with an overhead projector.
- Using a COMPUTEREYES/2 video digitizer ($129 from Digital Vision) and a FingerPrint Plus card, digitize live video images and images from books, magazines, and so on. Then create a slide show of images and send it via modem to other schools. ©

Philip I. Nelson Assistant Editor

Journey From The Center Of The ST: Part 1

Explain, in 25 words or less, each of the following ST acronyms: BIOS, XBIOS, GEMDOS, TOS, VDI, GDOS, AES, GEM. Nearly every ST owner knows that these terms identify various parts of the ST's system software—the fundamental "stuff" that makes it work. But what does each one do, and how do they all fit together? We'll spend the next two months seeking basic answers to these questions.

This knowledge is essential for any serious programming. And even if you're not aspiring to guru status, you may find the survey helpful for those occasions when the air fills with a thick fog of computerspeak. We'll start at the center—the most primitive level of the machine—and work our way out. Waiting for us when we get to the surface will be GEM, the highest level of system software and the element responsible for much of the ST's personality.

More Than A Pile Of Chips

Although you may picture computers as mechanical devices, the most important part of any computer consists of programs. The hardware inside a computer—memory chips, microprocessor, and so on—differs little from one machine to the next and is almost useless by itself. What brings the hardware to life is the system software, a collection of built-in programs which the computer uses to function.

The most basic level of system software is the operating system. Some machines, like the current ST, store this software in a ROM (Read Only Memory) chip. Others, like the original ST models, load the operating system from a floppy disk when you turn on the power. No matter where it comes from, the operating system is necessary in order for the computer to do anything at all.

Layer Upon Layer

The ST's operating system, collectively dubbed TOS, is actually composed of three separate parts: BIOS (Basic Input/Output System), XBIOS (eXtended BIOS), and GEMDOS (GEM Disk Operating System).

If you visualize the ST's system software as a series of layers, then BIOS is the lowest layer. The BIOS makes it possible for the ST to perform simple, fundamental tasks which are necessary and common to every modern computer: printing a character, writing to a disk sector, and so on. Without this base level of functionality, none of the computer's higher capabilities would be of use.

The next layer of the operating system is XBIOS, which provides more sophisticated input/output functions and supports hardware features that the ST doesn't share with other computers. The XBIOS lets you do tasks such as changing the color palette, communicating through the MIDI port, reserving blocks of memory, or formatting a floppy disk. XBIOS is a bit of a grab bag, but its functions generally involve some low-level task tied to an ST-specific architectural or hardware feature.

Next we come to GEMDOS, which provides high-level disk operations such as searching a directory, as well as more sophisticated character input/output, large-scale memory management, and functions for loading and running programs. While BIOS and XBIOS are tied to hardware characteristics, GEMDOS is said to be hardware independent. That is, GEMDOS calls BIOS and XBIOS routines to perform hardware functions, rather than directly accessing the ST's hardware.

This hierarchical arrangement typifies the system software as a whole. As with a ladder or pyramid, the higher levels of the structure depend on lower levels for support. At the bottom are fundamental routines which perform one or two rudimentary tasks and are comparatively simple to use. As you move up the ladder, system routines become more powerful and versatile, and often more complicated to use. It takes a lot more setup to, say, open a GEM window, than it does to output a single character. And you have a lot more decisions to make when calling the open-window routine.

The Face Behind The Mask

Once we install the three software elements of its operating system—BIOS, XBIOS, and GEMDOS—the ST becomes a working device. If the Atari designers had stopped with TOS, the ST would be something like a fast MS-DOS computer with lots of memory. You would have a keyboard, simple character output to the screen, and the ability to access peripherals such as the disk drive.

In fact, there's a simple way to use the PC-like machine hidden inside the ST. When you run a GEM application (a file that ends with .PRG), you have the complete computer at your disposal, including its flashy and convenient visual interface. But when you run a TOS application (a file that ends with .TOS or .TTP), the ST strips off the glamorous mask of GEM and uses only the operating system. The TOS-only machine is powerful, but rather dull and cumbersome compared to the ST we use most of the time.

What's missing from this picture, of course, is GEM—the user interface that provides the windows, menus, and other graphics elements that make the ST much more than just a fast PC. We'll finish our journey in the second part of this column, which examines the various software elements that make up GEM. ©

Bill Wilkinson

The Protection Racket

How many times have you written a program that is supposed to write to a disk file, only to have the drive make an ugly grinding noise and then have BASIC (or your program, if it is doing error trapping) tell you that the disk is write-protected? If you're like me, the answer is *very* often.

I'm usually cautious when writing my programs. I leave the write-protect tab on all my disks until I am 99 percent certain that the program will work. So during the development of a program, I tend to get lots of error number 144 messages. (Error 144 can mean anything from a too-fast disk drive to a 1050 drive's door being open, but most often it means that a disk is write-protected.)

Testing First

Wouldn't it be neat to be able to test if a disk is write-protected *before* you open a file for a write? Actually, you can. The method is a fairly obscure one; and to find it, once again I had to consult the old *Atari Technical Reference Manual.*

A short sidetrack: Those of you who don't mind searching through a couple hundred pages to find a small item buried in a lot of software (and hardware) engineering talk really should own a copy of this manual. It is now a six-year-old document, but it is still useful and accurate (aside from some of the RAM locations used by the newer XL/XE operating systems—for which you can consult COMPUTE!'s revised version of *Mapping the Atari*). This is a tribute to the design capabilities of the engineers from the old Atari and the astuteness of today's Atari: Never has an entire line of personal computers stayed so compatible.

Where The Secret Lies

The secret to the write-protect is buried in the information about the disk drive's status command. (Don't confuse the drive status with a file's status, as tested by BASIC's STATUS command.) If you can recall my September 1985 column regarding SIO (Serial Input/Output) commands, it may be easier for you to understand the code which follows. I am not going into great detail, but, briefly, an SIO call must have certain information placed in page 3 (locations $300 to $30B, specifically) before the program jumps to the subroutine at location $E459. (You can use my handy-dandy number converter program of a few months back to convert those hex numbers to decimal.) Information placed in page 3 includes the drive number, type of request (*S*, for status, in this case), the address of a buffer, and the number of bytes to transfer.

It is this last set of information that is most important to us: The drive status returns four bytes of information that we need to receive somewhere so that we can analyze it. We'll take a closer look at the accompanying listing later, but for now just notice that we dimension a buffer (BUF$) in line 30100. The four bytes of BUF$ will be used to receive the four bytes of drive status. We'll be using only the first byte of that status, because that is where the write-protect flag is located. Some of the bits in that byte are related to various hardware error conditions, which we need not discuss, leaving the following bits as useful to us:

Bit 3	$08	write-protected disk
Bit 5	$20	double density disk
Bit 7	$80	1050 enhanced density disk

Well, well! So not only can we find out if the disk is write-protected, we can also find its density.

A Useful Subroutine

In this month's listing, then, lines 30000 and beyond are a subroutine that you can include in your own programs. To use the subroutine, simply set the variable DRIVE to a valid drive number (1 to 4, usually) and GOSUB 30000. Upon return, the variable CHECK will contain one of the following values:

less than 0	an error occurred (invalid drive number, for example)
0	disk may be written to
1	disk is write-protected

If the value returned is less than zero, then the value will be the negative of the appropriate error code. For example, if you try to check a drive that isn't turned on, the value returned should be −138, indicating that error 138—device timeout—has occurred.

Similarly, a second number is returned in a variable named DENSITY. Its meaning:

−1	density value was invalid
1	drive is single density
2	drive is double density
3	drive is enhanced density

And all of this is demonstrated by the code in lines 100 through 220. These lines are provided just to give you a test bed to try out the subroutine of lines 30000 and up. Try the program, and then incorporate the subroutine in your own programs. And never again will you or your users see that dreaded error number 144 (unless your drive speed is off—but that's another topic).

Just a couple of last comments: Notice line 30170. This demonstrates another programming trick. Worried about making sure that your RESTORE statements refer to the right DATA line numbers? For short DATA lines, why not simply combine RESTORE and DATA on the same line, as in 30170?

Also, notice the sneaky way that BUF$ is dimensioned in line 30100. If we come to line 30100 a second time, we get an error when we try to re-DIMension BUF$. But,

because of the TRAP, the error is effectively ignored—a useful way of making sure a variable is dimensioned only once.

Write-Protect Checker

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
EA 100 REM SIMPLE PROGRAM TO
       DEMONSTRATE
BC 110 REM THE WRITE PROTECT
       CHECKER ROUTINE
HA 120 PRINT :PRINT :PRINT "
       DRIVE NUMBER ";
JD 130 INPUT DRIVE
AJ 140 GOSUB 30100
KG 150 IF CHECK<0 THEN PRINT
       "ERROR # ";-CHECK;"
       ACCESSING DRIVE ";DRI
       VE:GOTO 120
CJ 160 IF CHECK>0 THEN PRINT
       "DISK IS WRITE PROTE
       CTED";
LI 170 IF CHECK=0 THEN PRINT
       "DRIVE IS READY";
PC 180 IF DENSITY=1 THEN PRI
       NT ", SINGLE DENSITY.
       "
ON 190 IF DENSITY=2 THEN PRI
       NT ", DOUBLE DENSITY.
       "
GB 200 IF DENSITY=3 THEN PRI
       NT ", ENHANCED DENSIT
       Y."
FI 210 IF DENSITY<0 THEN PRI
       NT ", UNKNOWN DENSITY
       ."
FO 220 GOTO 100
KG 30000 REM SUBROUTINE TO C
         HECK DISK FOR WRITE
          PROTECT
NI 30010 REM
HM 30020 REM ENTER WITH DRIV
         E TO CHECK IN DRIVE
          (1 TO 8)
MK 30030 REM
JJ 30040 REM ON EXIT, CHECK
         WILL NORMALLY BE 0
LO 30050 REM IF CHECK IS 1,
         DISK IS WRITE PROTE
         CTED
HD 30060 REM IF CHECK IS <0
         THEN CHECK IS NEGAT
         IVE OF SIO ERROR CO
         DE
ON 30070 REM FOR CHECK=1 OR
         0, DENSITY IS RETUR
         NED:
JL 30080 REM DENSITY IS 1,2,
         3 FOR SINGLE, DOUBL
         E, ENHANCED DENSITY
BG 30090 REM DENSITY IS -1 I
         F UNRECOGNIZED
LK 30100 TRAP 30110:DIM BUF$
         (4)
BH 30110 POKE 768,49:POKE 76
         9,DRIVE
PI 30120 POKE 770,83:POKE 77
         1,64
CE 30130 POKE 773,INT(ADR(BU
         F$)/256)
CH 30140 POKE 772,ADR(BUF$)-
         256*PEEK(773)
JA 30150 POKE 774,2:POKE 775
         ,0
JH 30160 POKE 776,4:POKE 777
         ,0
AB 30170 RESTORE 30170:DATA
         104,76,89,228
BA 30180 FOR BYTE=1 TO 4:REA
         D CHECK
BN 30190 BUF$(BYTE)=CHR$(CHE
         CK):NEXT BYTE
NK 30200 BYTE=USR(ADR(BUF$))
BE 30210 CHECK=PEEK(771):IF
         CHECK>127 THEN CHEC
         K=-CHECK:RETURN
DJ 30220 CHECK=0:BYTE=INT(AS
         C(BUF$)/8)
BC 30230 IF BYTE/2<>INT(BYTE
         /2) THEN CHECK=1
DC 30240 BYTE=INT(BYTE/4):DE
         NSITY=-1
DH 30250 IF BYTE=0 THEN DENS
         ITY=1
DK 30260 IF BYTE=1 THEN DENS
         ITY=2
DP 30270 IF BYTE=4 THEN DENS
         ITY=3
NN 30280 RETURN
```

The Beginner's Page

C. Regena

Arithmetic In BASIC

In most of my guest lectures or computer presentations, I emphasize that a mathematics background is *not* necessary to use or to program a computer. However, we do use numbers quite frequently in programming. This month we'll discuss a few arithmetic operations that are important when programming in BASIC.

The computer is a sophisticated calculator, and you can add and subtract numbers by using plus and minus signs:

```
PRINT 4 + 3
PRINT 12 - 7
PRINT 5 + 8 - 3 + 2 - 1
```

Notice that the numbers with the + and − signs are written all on one line, and the computer will add or subtract from left to right.

Of course the computer can multiply and divide also. To multiply, use the * (asterisk) symbol. To divide, use the / (slash) symbol. Remember that everything has to be on one line. To multiply 3 times 4, the equation is represented by 3 * 4. In division, the first number is divided by the second number—12/4 means 12 divided by 4.

Now the order of operations becomes important. The standard rule is to calculate from left to right, first executing the multiplication and division, and then the addition and subtraction. If we have an equation 8 + 12/4, the division 12/4 is performed first. The result, 3, is then added to 8, for a final result of 11.

Let's suppose you really wanted 8 added to 12 first, and then that sum divided by 4. On paper, you could write 8 + 12 as the numerator of a fraction; then under the bar, write the 4 to indicate division. However, with computers, the expression must be written all on one line. Parentheses may be used to group numbers. (Be sure you always have a *matching pair* of parenthesis.) So for our example, we can write (8 + 12)/4. This time the result is 20/4, or 5. You might want to experiment with various expressions using all the operator signs and different placements of the parentheses in PRINT statements.

One more step in the order of operations: Exponents are performed *before* the multiplication and division operations. The exponentiation operator is an up arrow (↑) on Commodore computers and a caret (^) on most others. For example, 5 + 4 ^ 3 * 10 starts with 4 ^ 3, which is 4 raised to the third power (cubed), or 64. The next operation is the multiplication 64 * 10, or 640. The next operation is the addition 5+640, or 645, for the final result.

Expressions With Variables

When you're comfortable using actual numbers (constants), try using numeric variables. A variable is simply a letter or name that represents a number. If you've had any algebra, you'll recall using letters in numeric expressions or formulas. For example, A=L*W is a formula for *area* equals *length* times *width*. When you solve a problem, you'll have a number for L and a number for W, which results in a number for A. Here's a sample program:

```
100 L=10
110 W=5
120 A=L * W
130 PRINT A
```

Line 120 performs the operation of multiplying the numbers represented by the variables L and W; then line 130 prints the number represented by the variable A. If you wish, you can delete line 120 and just use PRINT L * W for line 130. The result is the same.

The arithmetic symbols and order of operations are the same for variables as they are for constants.

In your program, you may make calculations first and then use the result in another statement, or you can indicate calculations within the statement. Assuming all variables have previously been defined, these two program segments will have the same results:

```
500 X=A * B + C
510 Y=D/E
520 PRINT X,Y
```

or

```
500 PRINT A * B + C,D/E
```

So far we've seen arithmetic in the PRINT statement, but this is not the only place you can use numbers. Any statement requiring numeric parameters can use actual numbers (constants), variables, or numeric expressions. You can use these arithmetic operations to make calculations for the numbers required in other statements. For example, you may need to calculate a row and a column to begin printing or drawing. You may want to draw to one point with coordinates calculated in terms of the coordinates of another point. You may want to use SOUND commands with variables and expressions rather than constant numbers. FOR-NEXT statements may have variable index limits. IF-THEN statements may compare constants, or they may compare constants with variables, or they may compare numeric expressions. Functions such as SIN(X) can use numeric expressions in the argument. Here are some examples:

```
200 PRINT INT(A * B * RND(1) + 1)
210 IF BLUE>RED + 1 THEN 500
220 PRINT AT (ROW + 2,COL + 5),
    NAME$
230 SOUND F,D * 4,V + 2
240 CIRCLE (X * 8 + A,Y + B),R + 5
250 LINE (X,Y)-(X + L,Y + W)
260 PRINT SGN(D * E + (F - G)/8)
270 FOR K=A + B TO C - D STEP
    (-1) ^ S
```

Desktop Video

Because IBM PCs and compatibles are so firmly entrenched in the business microcomputer market, any new, non-compatible computer has got to offer something that IBM doesn't. For the Macintosh, desktop publishing has become a Trojan Horse, an application that's allowed it to breach the corporate walls. For the Amiga, desktop video may be the application that secures its niche in the marketplace.

The Turnaround

The Amiga's design makes it particularly well-suited for video applications, since its own video output more closely follows broadcast television standards than other personal computers. Since it was anticipated right from the start that the Amiga would be used to combine computer graphics with live video, provisions were made in its hardware and software design to support such applications. The Amiga is the only personal computer you can buy that has video input lines built right into the video output port. Commodore has been a bit slow to bring to market the hardware peripherals necessary to realize the Amiga's video potential, but things seem to be turning around at last.

One of the most exciting developments has been Commodore's recent release of the Genlock interface. Genlock is widely misunderstood among Amiga owners, because it's more of a video peripheral than a computer add-on. In effect, it turns the Amiga into a video processing accessory for your VCR, by combining a live-action video signal from a video camera or VCR with the Amiga's great computer graphics. The resulting picture can be recorded by another VCR.

Genlock replaces everything on the screen that's colored in the background color (that takes its color from hardware color register 0). On the Workbench screen, for example, every dot that normally appears as blue is replaced by the live video picture. Software doesn't have to do a thing to work with Genlock. As soon as you attach the interface and power up the computer, the operating system sees it's there and allows you to transparently replace the background computer video color with live action video.

If Genlock only offered simple video titling, it would still allow the Amiga to surpass dedicated units costing a lot more. But Genlock lets the Amiga do things like interpose animated "characters" in the middle of live action video. Although the Genlock's signal quality isn't of the broadcast quality needed by television stations, it's certainly good enough for industrial use.

Amiga Live!

If Genlock converts computer graphics into video graphics, a video digitizer does almost exactly the opposite. It takes video graphics and changes them into computer graphics. Commodore has announced a product called Amiga Live!, which will let you freeze a single frame (or series of frames) from live action color video, and then display it on the Amiga in its own computer-graphics format. The resulting picture could be saved to disk, and later loaded into paint programs where they could be edited further.

Commodore-Amiga hasn't delivered on its promised "framegrabber" as of yet. The Digi-View Digitizer from NewTek, however, is an excellent low-cost alternative. Unlike Amiga Live!, Digi-View works too slowly to capture individual frames from live action video. In order to produce a color picture, it must capture three separate "views" of the images, representing the red, green, and blue components of the picture. To screen out other colors, a plastic wheel containing red, green, and blue filters is rotated in front of the camera lens. The whole process may take half a minute, but the quality of the pictures produced is simply astounding. Also, the Digi-View software is extremely versatile, allowing you to capture pictures in virtually any of the Amiga's video modes, including the Hold and Modify mode in which up to 4096 colors can be displayed on screen at once. When capturing pictures in the normal 32-color mode, you can specify the number of colors (1–32) to be used, and even specify the actual color palette. In this way, you can conform the image to the color palette used by an existing paint program image.

One drawback of the Digi-View system is that you normally can't use it to digitize a still frame from a VCR in color, since you can't apply the colored filters to the image. With the Genlock interface, however, you can run the video into the Genlock (set it to video only), and connect the separate red, green, and blue outputs from the Genlock's 23-pin output connector in turn to the Digi-View.The resulting Digi-View picture is quite good, as long as your VCR provides a good still-frame picture.

The applications for the Genlock and video digitizers already exist. Genlock can be used for the kind of video production work that goes on daily in every cable TV company, college video lab, and industrial video shop. It's just that up until now, it took thousands of dollars to put together the equipment needed to perform such tasks. With the Amiga, most of what's required is already sitting on your desktop.C

Donald B. Trivette

A Bit Of BASIC

I'm never completely pleased with the output from off-the-shelf programs—even the good ones. There are many commands I'd like to add to word processing programs, and the screens in my accounting software—one of the most popular programs around—could be redesigned to eliminate clutter. I'd love to make these changes, but without source code and, in many cases, a knowledge of the native language, it's just not possible. Whenever I can do something about the design of the output, though, I usually spend—or waste—a lot of time rearranging things. Thus the focus of this month's column: in particular, *The Personal Ancestral File*, a program from The Church of Jesus Christ of Latter-day Saints; in general, how to use binary data.

Last October, I told you what a terrific program *PAF* is for recording genealogy and family records. And it still is. It's just that there are a few things I'd like to change on the printouts. I want both birth and death dates printed, and the age at death calculated and printed in parentheses where appropriate; I'd like ID numbers not printed; and I want a different page header with certain lines printed in bold. Minor changes, really, if only I could get my hands on the source code, which I've been told is written in C.

I can't change *PAF*, but I can write my own programs to organize and print data from *PAF* files. Basically the program uses four files, and I know what their formats are—which at first appears to be a problem for BASIC.

The main file contains one record indicating each individual's date of birth, date of death, and sex, as well as a pointer to the name, which is recorded in another file. The three things I want from this file—the information I'll use in my own program—are the name pointer and the two dates. The problem is that the pointer and dates are not saved in the file as readable text, such as January 31, 1960 or 1-31-60. To conserve space (and make things difficult for people like me), a date is represented in a three-byte binary field. That is, it's compressed to occupy the same space as the three characters *ABC*.

Information In Bits

Here's how it's done—remember, one byte has 8 bits, so the date occupies 24 bits. The number for the year is contained in the first byte and half of the second byte—a total of 12 bits; the next 5 bits represent the month; the next 5 are the day; and the last 2 are a status code which we don't need. If you've been keeping count, you'll know that the last bit in the month is also the first bit in the third byte. Now how can you work with data that's partly in one character and partly in another? Actually, this is a common way to represent numbers and other data, and many programming languages like PL/1, C, and Pascal have ways to work directly with binary data. BASIC doesn't.

But BASIC has something called logical operators that work at the bit level. Let's see how to grab the day number (in the third byte). First we read a record and assign the three date bytes to X$, Y$, and Z$, respectively. Now we want to convert a part of Z$ to a number. The first bit we ignore because it's really the last bit in the month, and the last two we ignore since we don't need them.

The logical AND operator can be used to "wipe out" unwanted bits. If we AND the Z$ variable with the bit pattern 01111100 we'll be erasing bits 7, 1, and 0 (bits are numbered from right to left); the bits for the day—6, 5, 4, 3, and 2—are unchanged by ANDing with the 1's in the bit pattern. So our BASIC statement might read DAY = (ASC(Z$) AND 01111100), except BASIC doesn't have a binary data type and won't understand the 0's and 1's. We have to convert the binary pattern to hexadecimal—a base 16 numbering system—in order to write it in BASIC. To do this, divide the eight bits into two groups of four and then convert to the appropriate hex value. (Many programming books show how to do this.) The bit pattern 0111 1100 is the same thing as hex 7C, which is written in BASIC as &H7C. So our statement should be DAY. = (ASC(Z$) AND &H7C). Note that we use the ASC function to convert the character value of Z$, which is what we had to use to read data from the file, to a numeric value.

If you print the variable DAY, you still won't get a number between 1 and 31. We aren't quite through. Here's a decimal analogy: Suppose we had the number 432651 and wanted to make a new number out of the middle two digits. After the ANDing process we would have 002600, which is 2,600—and we are after the 26 part. If we were working in decimal, we'd simply divide the ANDed value by 100 (10^2) to lose the rightmost two places and come up with 26. Since we're working in binary, we divide by 4 (2^2) to shift two places to the right. Thus DAY = (ASC(Z$) AND &H7C) / 4 is the correct BASIC statement; printing DAY will print a value between 1 and 31.

To try it yourself, assign Z the initial value of 234. Since we're not reading data from a file and you would have a difficult time typing the character represented by the ASCII value 234, we won't need a character variable or the ASC function. Just make Z equal 234 and decode the day contained therein: Thanksgiving Day, 1987. ©

$15,000.00
Programming Contest!

COMPUTE!'s PC Magazine
For IBM PCs & Compatibles

First Prize $7,500.00 Second Prize $2,500.00
Five Honorable Mentions $1,000.00 each

COMPUTE! Publications, Inc., a longtime leader in personal computer publishing, is launching a new magazine this summer: COMPUTE!'s PC Magazine for IBM PCs & Compatibles. Each bimonthly issue will include a floppy disk filled with programs, source code, and other useful information. We're looking for the very best original software for IBM PCs, XTs, and compatibles, and are sponsoring a programming contest with a total of $15,000.00 in prize money for the top six winners. That's $7,500.00 for First Prize; $2,500.00 for Second Prize; and five Honorable Mentions at $1,000.00 each. In addition, the winners will receive our standard purchase fees for publication in our magazine and royalties if republished in COMPUTE! books.

Even if your contest entry doesn't win a prize, you will still earn purchase fees if we accept your program for publication.

Interested? If so, read these rules:

1. Entries must be your original work, previously unpublished in any form. All those whose programs are accepted will be required to affirm this in writing.

2. You can submit as many entries as you want, but we cannot consider programs which have been either entered in other contests or submitted for publication elsewhere at the same time.

3. The contest deadline is October 31, 1987. All entries must be received at our offices by this date. Programs submitted after this date will still be considered for publication, but will not be entered in the contest. If we purchase an entry for publication before the deadline, the entry is still eligible to win.

4. Entries are allowed (and encouraged) in virtually all software categories: home and business applications, education, recreation, telecommunications, graphics, sound and music, and utilities.

5. Entries may be written in any programming language—including BASIC, C, machine language, Pascal, and Modula-2—as long as they meet two requirements. First, if you're using a compiled language, the compiled object or runtime code must be a self-standing program that can be run by someone who doesn't own a copy of the language. (Interpreted BASIC is an exception. It can be assumed that nearly everyone owns a copy of BASICA or GW-BASIC.) Second, we must be able to legally distribute the program without incurring licensing fees or other obligations to the maker of the language. If you're not sure whether a certain language qualifies, contact its maker for clarification.

6. Entries must be submitted on 5¼-inch floppy disks. If your program is written in a compiled language, you must submit both the runtime code and all of the source code required to compile the program.

7. Entries must be accompanied by an article which explains how to use the program and what it does. If your program employs any new or unusual techniques that you think will be of interest to other programmers, you can also describe how the program works. (If you feel that writing is not your strong point, please do not hesitate to enter; this is a programming contest and the entries will be judged solely on the basis of the programs submitted.)

8. Submissions which do not win a prize and are not accepted for publication will be returned only if accompanied by a self-addressed, stamped mailer.

9. The staff of COMPUTE! Publications, Inc., will judge the contest, and all decisions regarding contest entries and acceptances will be solely at the discretion of COMPUTE! Publications, Inc. All decisions are final. This includes decisions regarding creativity, similarity among entries, and general suitability.

10. Winners will be announced by COMPUTE! Publications, Inc., in early 1988.

11. This contest is void where prohibited by law. Full-time, part-time, and previous employees of COMPUTE! Publications, Inc. and Capital Cities/American Broadcasting Corporation are ineligible for the contest, but may still submit work for publication at standard rates.

Every contest entry must include this signed form:

I warrant that the program presently entitled _____ is my own original work and that the work has not been submitted for consideration elsewhere, nor has it been previously published in any form. If my work is accepted by you, I understand that your decision as to the selection of winners and awarding of prizes is final and without recourse on my part. Should you select my submission, I understand that I will receive no payments until I sign your standard contract, which includes assignment of the copyright of the program to COMPUTE! Publications, Inc., and that you may use my name and image in promotional materials and other forms. *(If you are under age 18, your parent or legal guardian must sign for you.)*

Signature: _____

Address entries to: PC Programming Contest
COMPUTE! Publications, Inc.
P.O. Box 5406
Greensboro, NC 27403

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

64 RAMdisk

Hubert Cross

This Commodore 64 utility creates an electronic disk drive that's much faster than an ordinary disk drive. Since it uses "hidden" memory, the RAMdisk doesn't reduce the amount of programming space available for your use. No machine language knowledge is needed to use the program.

A RAM disk is a familiar device to many personal computer owners. In simple terms, a RAM disk emulates a disk drive entirely in the computer's RAM (Random Access Memory), allowing you to store and retrieve files much faster than you can from a mechanical disk drive. The Amiga, for instance, includes a built-in RAM disk as part of its system software; and RAM disks are popular utility programs for computers such as the Atari ST and IBM PC/PCjr.

"64 RAMdisk" is a RAM disk for the Commodore 64 which doesn't subtract a single byte from the space normally available for programming. You control the RAMdisk with simple BASIC commands, and the program is compatible with the "DOS Wedge" utility (supplied on the 1541/1571 Test/Demo disk), "TurboTape" (COMPUTE!, January 1985), and "TurboDisk" (COMPUTE!, April 1985).

Typing In 64 RAMdisk

Because 64 RAMdisk is written in machine language, the program must be typed with the "MLX" machine language entry program printed elsewhere in this issue. Here are the addresses you need to enter 64 RAMdisk with MLX:

Starting address: 0801
Ending address: 12A8

64 RAMdisk is designed to load and run exactly like a BASIC program. Load the program with LOAD "RAMDISK",8 for disk or LOAD "RAMDISK" (or simply LOAD) for tape. After the program loads, type RUN and press RETURN. The program relocates its code to the safe memory area beginning at location 49152 and prints the message RAM DISK ACTIVATED. 64 RAMdisk is now ready to use.

RAMdisk Commands

Following is a list of 64 RAMdisk commands. All of these commands work in BASIC direct mode (when you aren't running a program).

DIR. This command displays a directory of the files in the RAMdisk. For instance, if you type DIR and press RETURN, 64 RAMdisk prints a directory of the RAMdisk. The number of bytes free for use is printed at the bottom of the directory display.

NAME. The NAME command can be used to change the RAMdisk's name, which appears in reversed characters at the top of the directory display.

RSAVE. The RSAVE command saves a file to the RAMdisk, storing a copy of the BASIC program currently in memory. Here is the correct syntax for the command:

```
RSAVE "PROGRAM"
```

Of course, you should substitute the name of your program for PROGRAM in this example. If you forget to specify a filename, 64 RAMdisk prints the error message MISSING FILENAME and doesn't save anything. The filename can be any combination of 15 or fewer characters.

RLOAD. The RLOAD command copies any program in the RAMdisk back to BASIC. For example, the statement RLOAD "EXAMPLE" loads the program named EXAMPLE from the RAMdisk, storing a copy of it in the usual BASIC program space.

SCRATCH. The SCRATCH command deletes a file from the RAMdisk. For instance, the statement SCRATCH "PROGRAM" removes the file named PROGRAM from the RAMdisk. If you attempt to scratch a file that doesn't exist in the RAMdisk, the program prints the error message FILE NOT FOUND and doesn't scratch anything.

RENAME. This command changes the name of a file in the RAMdisk. Here is the syntax to use:

```
RENAME "OLDNAME","NEWNAME"
```

The RENAME command requires two filenames: the name of the existing file (OLDNAME in this example) and the new name which you want that file to have (NEWNAME in this example). The filenames are separated with a comma. The error message FILE NOT FOUND appears if you attempt to rename a nonexistent file. The error message FILE EXISTS appears if you try to use a new name which already exists in the RAMdisk. (Every file in the RAMdisk must have a unique name.)

REPLACE. This command replaces the designated RAMdisk file with the BASIC program in memory, using the same filename. For instance, REPLACE "TEST" deletes the program TEST from the RAMdisk and saves the BASIC program in memory, using the filename TEST. This is normally done when

you have made changes to a program and wish to save the revised version with the same name.

RNEW. The RNEW command does a NEW of the RAMdisk, erasing every file that it contains. *Be very careful when using this command,* since the program does not ask you to confirm this action, and there is no easy way to recover files after an RNEW. If you include a name after RNEW, this command renames the RAMdisk with the name specified. For instance, this command erases everything in the RAMdisk and renames it as MYDISK:

```
RNEW "MYDISK"
```

EXIT. This command disables the RAMdisk and gives you the option to save the entire contents of the RAMdisk in a single tape or disk file. When you type EXIT and press RETURN, the program prints this prompt:

```
EXIT RAM DISK
ARE YOU SURE? (Y/N)
```

If you type N at this prompt, nothing happens. If you type Y at this prompt, the program copies the 64 RAMdisk machine language program, as well as all the files in the RAMdisk, back into the BASIC program space. This is done so that you can save the entire RAMdisk as a single file, using a normal SAVE command from BASIC. The advantage of this method is that you can reload the RAMdisk program *and all of the saved files* in one operation, at the beginning of your next programming session. You do not need to use any special tricks to save this file. Here is a typical SAVE command:

```
SAVE "CABOODLE",8
```

This example saves 64 RAMdisk and the entire contents of the RAMdisk under the filename CABOODLE. If you replace the 8 with a 1, the file is saved to tape instead of disk. (The resulting file will be considerably longer than 64 RAMdisk itself, since it contains a copy of every file in the RAMdisk, as well as a copy of 64 RAMdisk.)

Once you have saved a master file, you can reload 64 RAMdisk and the individual files with the same load and run commands you would ordinarily use to activate the RAMdisk. In this case, for instance, you would use these commands:

```
LOAD "CABOODLE",8
RUN
```

If you type RUN and press RETURN at this point, the program moves the 64 RAMdisk code to its normal location beginning at location 49152, then transfers each of the saved files to the RAMdisk, too. When the startup message appears, the RAMdisk is ready to use and all of the files are in place.

Notice that the filename used with EXIT has no connection with the name of the RAMdisk itself (see NAME) or the names of individual files contained in the RAMdisk. As a practical matter, however, you will probably want to use a name that reminds you what individual files the master file contains. If you previously used NAME to give the RAMdisk a meaningful name, you can use the same name when saving the entire disk with EXIT.

RAMdisk Notes

64 RAMdisk is designed as a convenience for saving and loading BASIC programs, not as a total replacement for a disk or tape drive. Thus, it supports only one type of file—a BASIC program (PRG) file—and only one form of file access (saving and loading). You cannot use the RAMdisk for other types of files, such as sequential (SEQ) files. And, for instance, you cannot OPEN a file in the RAMdisk for reading or writing, even though those are legitimate operations for program (PRG) files on tape or floppy disk.

Like all RAMdisks, 64 RAMdisk is volatile, meaning that it disappears completely, together with all its contents, when you turn off the computer. For this reason, you should make frequent backup copies of RAMdisk programs on disk or tape. To make a backup copy, load the program into BASIC memory with RLOAD; then save it to disk or tape in the usual way.

This program occupies the memory area beginning at location 49152 ($C000), so you cannot use it with any other machine language program or utility that occupies the same space. Because this program uses the "hidden" RAM underlying the 64's ROM (Read-Only Memory) chips, it is also incompatible with programs which use that area of RAM. 64 RAMdisk *does* work with either TurboTape or TurboDisk, but not with both at the same time, since those programs are incompatible with one another. You must relocate TurboDisk, as explained in the TurboDisk article, before using it with 64 RAMdisk; the best place to put TurboDisk is as close as possible to the top of BASIC RAM.

The 64 has a total of 16K (16,384 bytes) of RAM under its ROM chips. Half of this lies under the BASIC ROM, and the other half lies under the Kernal operating system ROM. Not all of this RAM can be used for file storage with 64 RAMdisk. Every program stored in the RAMdisk requires an extra 18 bytes for a directory entry, 15 bytes for a filename entry, 1 byte as a filename marker, and 2 bytes for a pointer to the beginning of the next program.

Programs are stored beginning at the bottom of the RAM under BASIC and growing upward, toward higher memory locations. The directory begins at the top of the RAM under the Kernal and grows downward, toward lower memory locations. If you fill the 8K space under BASIC with programs, 64 RAMdisk uses as much of the RAM under the Kernal as needed. If you try to save a program that's bigger than the amount of free space left in the RAMdisk, the program prints the error message RAM DISK FULL and doesn't save anything.

It's theoretically possible to create so many individual files that the RAMdisk directory would fill all of the RAM under the Kernal ROM. However, since it would require more than 454 files to overflow the directory, 64 RAMdisk does not check for this unlikely situation, and does not print an error message if it occurs.

64 RAMdisk

Please refer to the "MLX" article elsewhere in this issue before entering the following program.

```
0801:0B 08 00 00 9E 32 30 36 EC
0809:31 00 00 00 4C FD 0F 4C 76
0811:1D C2 4C 30 C2 4C 5F C5 B8
0819:4C 10 C2 4C 5B C2 4C 47 36
0821:C3 4C 07 C4 4C 3E C3 4C 82
0829:55 C5 4C 1C C9 20 55 C0 DA
0831:A9 A9 8D 04 03 A9 C0 8D 40
0839:05 03 A9 69 8D 06 03 A9 8C
0841:C1 8D 07 03 A9 D0 8D 08 5A
```

```
0849:03 A9 C1 8D 09 03 A9 94 92
0851:A0 C0 4C 1E AB A9 00 8D DE
0859:FE FF 8D 78 CA A9 80 8D AD
0861:FF FF 8D 79 CA A9 DC 85 F6
0869:FB 8D 7A CA A9 FF 85 FC 2C
0871:8D 7B CA 60 0F B9 84 C0 AF
0879:99 EE FF 88 10 F7 A9 60 AE
0881:8D 73 C0 60 32 34 4B 20 6C
0889:52 41 4D 20 44 49 53 4B F7
0891:00 00 00 00 0D 52 41 4D 23
0899:20 44 49 53 4B 20 41 43 C9
08A1:54 49 56 41 54 45 44 0D 5A
08A9:00 A6 7A A0 04 84 0F BD CA
08B1:00 02 10 07 C9 FF F0 3E 23
08B9:E8 D0 F4 C9 20 F0 37 85 66
08C1:08 C9 22 F0 56 24 0F 70 6D
08C9:2D C9 3F D0 04 A9 99 D0 A2
08D1:25 C9 30 90 04 C9 3C 90 46
08D9:1D 84 71 A0 00 84 0B 88 82
08E1:86 7A CA C8 E8 BD 00 02 F9
08E9:38 F9 9E A0 F0 F5 C9 80 E5
08F1:D0 30 05 0B A4 71 E8 C8 4D
08F9:99 FB 01 B9 FB 01 F0 38 8F
0901:38 E9 3A F0 04 C9 49 D0 AA
0909:02 85 0F 38 E9 55 D0 9F C8
0911:85 08 BD 00 02 F0 DF C5 F8
0919:08 F0 DB C8 99 FB 01 E8 1B
0921:D0 F0 A6 7A E6 0B C8 B9 03
0929:9D A0 10 FA B9 9E A0 D0 3E
0931:B4 F0 0F BD 00 02 10 BC 7C
0939:99 FD 01 C6 7B A9 FF 85 2C
0941:7A 60 A0 FF CA C8 E8 BD C5
0949:00 02 38 F9 9E C1 F0 F5 56
0951:C9 80 D0 04 05 0B D0 9C 55
0959:A6 7A E6 0B C8 B9 9D C1 15
0961:10 FA B9 9E C1 D0 E0 F0 5F
0969:CA 30 03 4C F3 A6 C9 FF DF
0971:F0 F9 24 0F 30 F5 C9 CC A9
0979:B0 03 4C 24 A7 38 E9 CB 2E
0981:AA 84 49 A0 FF CA F0 08 52
0989:C8 B9 9E C1 10 FA 30 F5 21
0991:C8 B9 9E C1 30 05 20 47 83
0999:AB D0 F5 4C EF A6 4E 41 31
09A1:4D C5 52 45 4E 41 4D C5 42
09A9:53 43 52 41 54 43 C8 52 28
09B1:4E 45 D7 52 53 41 56 C5 6E
09B9:52 4C 4F 41 C4 44 49 D2 A2
09C1:52 52 55 CE 52 45 50 4C BD
09C9:41 43 C5 45 58 49 D4 00 EB
09D1:20 73 00 20 D9 C1 4C AE EF
09D9:A7 C9 CC 90 14 C9 D6 B0 FA
09E1:10 38 E9 CC 0A A8 B9 F8 73
09E9:C1 48 B9 F7 C1 48 4C 73 E0
09F1:00 20 79 00 4C ED A7 0E B2
09F9:C0 11 C0 14 C0 17 C0 1A 08
0A01:C0 1D C0 20 C0 23 C0 26 11
0A09:C0 29 C0 2C C0 2F C0 60 47
0A11:20 79 00 D0 03 4C 55 C0 55
0A19:20 1D C2 30 F8 20 81 C7 F2
0A21:A8 A9 00 99 EE FF 88 B1 C7
0A29:22 99 EE FF 88 10 F8 60 69
0A31:20 81 C7 20 2D C7 90 03 5D
0A39:4C 9E C7 20 FD AE 20 81 82
0A41:C7 48 A6 FB 86 6B A6 FC 0C
0A49:86 6C 20 2D C7 90 1D A5 F2
0A51:6B 85 FB A5 6C 85 FC 68 32
0A59:4C 90 C7 20 79 00 D0 03 23
0A61:4C A1 C7 20 81 C7 48 20 DA
0A69:2D C7 B0 03 4C AD C7 A5 9A
0A71:2D 38 E5 2B 85 FD 85 6D 36
0A79:A5 2E E5 2C 85 FE 85 6E 0D
0A81:AD 7A CA 38 ED 78 CA 85 54
0A89:6B AD 7B CA ED 79 CA 85 4B
0A91:6C AD 78 CA C9 00 AD 79 26
0A99:CA E9 C0 B0 0D A5 6B 38 BE
0AA1:E9 00 85 6B A5 6C E9 20 E4
0AA9:85 6C A5 6B 38 E9 12 85 1A
0AB1:6B A5 6C E9 00 85 6C 90 90
0AB9:0A A5 6B C5 FD A5 6C E5 4B
0AC1:FE B0 03 4C B4 C7 A5 2B E1
0AC9:8D DD C2 A5 2C 8D DE C2 E6
0AD1:AD 78 CA 8D E0 C2 AD 79 F3
0AD9:CA 8D E1 C2 AD FF FF 8D 1A
0AE1:FF FF EE DD C2 D0 03 EE FF
0AE9:DE C2 EE E0 C2 D0 03 EE 58
0AF1:E1 C2 AD E0 C2 C9 00 D0 79
0AF9:11 AD E1 C2 C9 C0 D0 0A 67
0B01:A9 00 8D E0 C2 A9 E0 8D B7
0B09:E1 C2 A5 6D D0 02 C6 6E D6
0B11:C6 6D D0 C8 A6 6E D0 C4 E1
0B19:A0 10 AD E0 C2 91 FB 8D 29
0B21:78 CA C8 AD E1 C2 91 FB 53
0B29:8D 79 CA 68 20 90 C7 AD C4
0B31:7A CA 38 E9 12 8D 7A CA 63
0B39:B0 03 CE 7B CA 60 20 51 63
0B41:C3 20 90 FF 4C 59 A6 24 8C
0B49:9D 10 F3 20 51 C3 4C 6C 51
0B51:CA 20 79 00 D0 03 4C A1 D0
0B59:C7 20 81 C7 20 2D C7 90 DD
0B61:03 4C 9E C7 A0 22 78 A5 80
0B69:01 48 A9 35 85 01 B1 FB 2A
0B71:85 6D C8 B1 FB 85 6E A0 4D
0B79:10 B1 FB 85 2D C8 B1 FB C7
0B81:85 2E 88 85 01 58 A5 2D 2D
0B89:38 E5 6D 85 FD A5 2E E5 04
0B91:6E 85 FE A5 2D C9 00 A5 B0
0B99:2E E9 C0 90 17 A5 6D C9 56
0BA1:00 A5 6E E9 E0 B0 0D A5 17
0BA9:FD 38 E9 00 85 FD A5 FE 78
0BB1:E9 20 85 FE A5 2B 85 FB 46
0BB9:18 65 FD 85 2D A5 2C 85 2B
0BC1:FC 65 FE 85 2E A0 00 78 54
0BC9:A5 01 48 A9 35 85 01 B1 0A
0BD1:6D 91 FB 68 85 01 58 E6 D0
0BD9:6D D0 02 E6 6E A5 6D C9 38
0BE1:00 D0 0E A5 6E C9 C0 D0 35
0BE9:08 A9 00 85 6D A9 E0 85 20
0BF1:6E E6 FB D0 02 E6 FC A5 D0
0BF9:FD 38 E9 01 85 FD B0 02 F1
0C01:C6 FE 05 FE D0 C1 60 A9 C4
0C09:0D 20 D2 FF A9 30 20 D2 2B
0C11:FF A9 20 20 D2 FF A9 12 95
0C19:20 D2 FF A9 22 20 D2 FF C7
0C21:78 A5 01 48 A9 35 85 01 B1
0C29:AD FE FF 85 6D AD FF FF 52
0C31:85 6E A9 EE 85 FB A9 FF 3B
0C39:85 FC A0 00 B1 FB 99 EC 05
0C41:C7 F0 03 C8 D0 F6 68 85 1F
0C49:01 58 A9 EC A0 C7 20 1E 7E
0C51:AB A9 22 20 D2 FF A9 0D E6
0C59:20 D2 FF 20 E1 FF F0 A6 CF
0C61:A5 FB 38 E9 12 85 FB B0 40
0C69:02 C6 FC A5 FB CD 7A CA 05
0C71:D0 0A A5 FC CD 7B CA D0 BB
0C79:03 4C 09 C5 A0 10 78 A5 7F
0C81:01 48 A9 35 85 01 B1 FB 44
0C89:85 69 C8 B1 FB 85 6A 68 26
0C91:85 01 58 A5 69 38 E5 6D 77
0C99:85 FD A5 6A E5 6E 85 FE 42
0CA1:A5 69 38 E9 00 A5 6A E9 E1
0CA9:C0 90 18 A5 6D 38 E9 00 C3
0CB1:A5 6E E9 E0 B0 0D A5 FD 86
0CB9:38 E9 00 85 FD A5 FE E9 2F
0CC1:20 85 FE A5 69 48 A5 6A A7
0CC9:48 A6 FD A5 FE 20 CD BD 9B
0CD1:68 85 6E 68 85 6D A9 05 0E
0CD9:85 D3 20 6C E5 A9 22 20 AE
0CE1:D2 FF 78 A5 01 48 A9 35 7E
0CE9:85 01 A0 00 B1 FB 99 EC B6
0CF1:C7 F0 03 C8 D0 F6 68 85 CF
0CF9:01 58 A9 EC A0 C7 20 1E 2F
0D01:AB A9 22 20 D2 FF 4C 56 27
0D09:C4 AD 7A CA 38 ED 78 CA 22
0D11:85 6D AD 7B CA ED 79 CA 82
0D19:85 6E AD 78 CA C9 00 AD F9
0D21:79 CA E9 C0 B0 0D A5 6D 66
0D29:38 E9 00 85 6D A5 6E E9 FA
0D31:20 85 6E A6 6D A5 6E 20 F3
0D39:CD BD A9 06 85 D3 20 6C 67
0D41:E5 A9 48 A0 C5 4C 1E AB 13
0D49:42 59 54 45 53 20 46 52 B3
0D51:45 45 2E 0D 00 20 81 C7 41
0D59:48 20 62 C5 4C 6F C2 20 0E
0D61:81 C7 20 2D C7 90 03 4C D7
0D69:9E C7 A0 22 78 A5 01 48 9F
0D71:A9 35 85 01 B1 FB 85 66 5D
0D79:C8 B1 FB 85 67 A0 10 B1 CB
0D81:FB 85 69 C8 B1 FB 85 6A A7
0D89:68 85 01 58 A5 69 38 E5 08
0D91:66 85 FD A5 6A E5 67 85 99
0D99:FE A5 69 C9 00 A5 6A E9 BB
0DA1:C0 90 17 A5 66 C9 00 A5 7D
0DA9:67 E9 E0 B0 0D A5 FD 38 4C
0DB1:E9 00 85 FD A5 FE E9 20 6E
0DB9:85 FE AD 7A CA 18 69 12 4F
0DC1:C5 FB D0 28 AD 7A CA 18 5F
0DC9:69 12 AD 7B CA 69 00 C5 4C
0DD1:FC D0 19 A5 66 8D 78 CA 41
0DD9:A5 67 8D 79 CA AD 7A CA B6
0DE1:18 69 12 8D 7A CA 90 03 A0
0DE9:EE 7B CA 60 A0 00 78 A5 55
0DF1:01 48 A9 35 85 01 B1 69 24
0DF9:91 66 68 85 01 58 E6 69 7C
0E01:D0 02 E6 6A A5 69 C9 00 EF
0E09:D0 0E A5 6A C9 C0 D0 08 67
0E11:A9 00 85 69 A9 E0 85 6A 8F
0E19:E6 66 D0 02 E6 67 A5 66 03
0E21:C9 00 D0 0E A5 67 C9 C0 3C
0E29:D0 08 A9 00 85 66 A9 E0 DE
0E31:85 67 A5 69 CD 78 CA D0 EB
0E39:B5 A5 6A CD 79 CA D0 AE 0B
0E41:AD 78 CA 38 E5 FD 85 6B CC
0E49:AD 79 CA E5 FE 85 6C AD E6
0E51:78 CA C9 00 AD 79 CA E9 68
0E59:C0 90 17 A5 6B C9 00 A5 5F
0E61:6C E9 E0 B0 0D A5 6B 38 63
0E69:E9 00 85 6B A5 6C E9 20 B4
0E71:85 6C A5 6B 8D 78 CA A5 60
0E79:6C 8D 79 CA A5 FB 38 E9 82
0E81:01 85 6D A5 FC E9 00 85 9C
0E89:6E A0 00 78 A5 01 48 A9 F7
0E91:35 85 01 B1 6D A0 12 91 88
0E99:6D 68 85 01 58 A6 6D D0 50
0EA1:02 C6 6E C6 6D A5 6D CD 55
0EA9:7A CA D0 DD A5 6E CD 7B AB
0EB1:CA D0 D6 AD 7A CA 18 69 B5
0EB9:12 8D 7A CA 90 03 EE 7B 28
0EC1:CA 78 A5 01 48 A9 35 85 FE
0EC9:01 A0 10 B1 FB 85 6B C8 41
0ED1:B1 FB 85 6C 68 85 01 58 F0
0ED9:A5 6B C9 00 A5 6C E9 C0 50
0EE1:08 A5 6B 38 E5 FD 85 6B F9
0EE9:A5 6C E5 FE 85 6C 28 90 5F
0EF1:17 A5 6B C9 00 A5 6C E9 66
0EF9:E0 B0 0D A5 6B 38 E9 00 BE
0F01:85 6B A5 6C E9 20 85 6C 7F
0F09:A0 10 A5 6B 91 FB C8 A5 9A
0F11:6C 91 FB A5 FB 38 E9 12 4A
0F19:85 FB B0 02 C6 FC A5 FB A0
0F21:CD 7A CA D0 9C A5 FC CD 6E
0F29:7B CA D0 95 60 85 6F A9 CC
0F31:DC 85 FB A9 FF 85 FC A5 EE
0F39:FB CD 7A CA D0 07 A5 FC AF
0F41:CD 7B CA F0 3B A0 00 78 62
0F49:A5 01 48 A9 35 85 01 B1 91
0F51:FB AA 68 85 01 58 8A D1 CD
0F59:22 D0 18 C8 C4 6F D0 E7 B9
0F61:18 78 A5 01 48 A9 35 85 47
0F69:01 B1 FB AA 68 85 01 58 52
0F71:8A F0 0D A5 FB 38 E9 12 B3
0F79:85 FB B0 BB C6 FC D0 B7 AE
0F81:60 20 9E AD 20 A3 B6 C9 4D
0F89:00 F0 16 C9 10 B0 17 60 15
0F91:A8 A9 00 91 FB 88 B1 22 0F
0F99:91 FB 88 10 F9 60 A2 04 2C
0FA1:2C A2 08 4C 37 A4 A2 C2 98
0FA9:A0 C7 4C BB C7 A2 D4 A0 62
0FB1:C7 4C BB C7 A2 DF A0 C7 58
0FB9:4C BB C7 86 22 84 23 4C 04
0FC1:47 A4 46 49 4C 45 20 4E 10
0FC9:41 4D 45 20 54 4F 4F 20 25
0FD1:4C 4F 4E C7 46 49 4C 45 65
0FD9:20 45 58 49 53 54 D3 52 DE
0FE1:41 4D 20 44 49 53 4B 20 8A
0FE9:46 55 4C CC 00 00 FF FF D6
0FF1:00 00 FF FF 00 00 FF FF 10
0FF9:00 00 FF FF A5 33 C9 01 A6
1001:A5 34 E9 80 B0 0A A5 37 76
1009:C9 01 A5 38 E9 80 90 0C 05
```

```
1011:A9 00 85 33 85 37 A9 80 C6
1019:85 34 85 38 A9 00 85 61 F6
1021:A9 C0 85 62 A9 7C 38 E5 B2
1029:61 85 63 A9 CA E5 62 85 9A
1031:64 A5 63 18 69 01 85 65 9A
1039:A5 64 69 08 85 66 A5 65 69
1041:18 69 04 85 69 A5 66 69 B8
1049:00 85 6A A0 00 B1 65 8D 41
1051:78 CA 38 E9 00 85 6B C8 BB
1059:B1 65 8D 79 CA E9 80 85 79
1061:6C AD 78 CA C9 00 AD 79 02
1069:CA E9 E0 90 0D A5 6B 38 9C
1071:E9 00 85 6B A5 6C E9 20 C0
1079:85 6C A9 01 85 02 A9 08 4C
1081:85 03 A0 00 B1 02 91 61 53
1089:E6 02 D0 02 E6 03 E6 61 4A
1091:D0 02 E6 62 A6 63 D0 02 04
1099:C6 64 C6 63 A5 63 05 64 6E
10A1:D0 E2 A0 02 B1 65 8D 7A CF
10A9:CA C8 B1 65 8D 7B CA A9 87
10B1:00 85 02 A9 80 85 03 A5 D3
10B9:6B 05 6C F0 34 A0 00 B1 43
10C1:69 91 02 E6 69 D0 02 E6 23
10C9:6A E6 02 D0 02 E6 03 A5 7D
10D1:02 C9 00 D0 0E A5 03 C9 49
10D9:C0 D0 08 A9 00 85 02 A9 ED
10E1:E0 85 03 A6 6B D0 02 C6 08
10E9:6C C6 6B A5 6B 05 6C D0 D2
10F1:CE AD 7A CA 85 02 AD 7B EB
10F9:CA 85 03 A0 00 B1 69 91 76
1101:02 E6 69 D0 02 E6 6A E6 7F
1109:02 D0 02 E6 03 A5 02 05 C6
1111:03 D0 EA 20 30 C0 20 42 4F
1119:A6 4C 86 E3 A9 31 A0 C9 CD
1121:20 1E AB 20 E4 FF C9 59 66
1129:F0 2B C9 4E D0 F5 4C D2 76
1131:FF 0D 45 58 49 54 20 52 F2
1139:41 4D 20 44 49 53 4B 0D D2
1141:41 52 45 20 59 4F 55 20 16
1149:53 55 52 45 3F 20 28 59 2D
1151:2F 4E 29 20 00 20 D2 FF EB
1159:4C 8F C9 02 A9 C0 85 03 3D
1161:A0 00 B1 02 38 E9 00 91 25
1169:02 C8 B1 02 E9 C0 91 02 8C
1171:A0 00 B1 02 18 69 01 91 34
1179:02 C8 B1 02 69 08 91 02 B5
1181:A9 4C 8D 58 C9 A9 8F 8D 64
1189:59 C9 A9 C9 8D 5A C9 A9 AF
1191:00 85 61 A9 C0 85 62 A9 66
1199:01 85 2B 85 02 A9 08 85 A7
11A1:2C 85 03 A0 00 B1 61 91 C0
11A9:02 E6 61 D0 02 E6 62 E6 17
11B1:02 D0 02 E6 03 A5 61 C9 F2
11B9:7C D0 EA A5 62 C9 CA D0 A6
11C1:E4 AD 78 CA 91 02 C8 AD 51
11C9:79 CA 91 02 C8 AD 7A CA 6A
11D1:91 02 C8 AD 7B CA 91 02 5D
11D9:A5 02 18 69 04 85 02 90 B3
11E1:02 E6 03 A9 00 85 61 A9 3C
11E9:80 85 62 A0 00 A5 61 CD 2B
11F1:78 CA D0 07 A5 62 CD 79 59
11F9:CA F0 32 78 A5 01 48 A9 F6
1201:35 85 01 B1 61 91 02 68 1A
1209:85 01 58 E6 61 D0 02 E6 E2
1211:62 E6 02 D0 02 E6 03 A5 C4
1219:61 C9 00 D0 D0 A5 62 C9 19
1221:C0 D0 CA A9 00 85 61 A9 50
1229:E0 85 62 D0 C0 AD 7A CA F4
1231:85 61 AD 7B CA 85 62 78 87
1239:A5 01 48 A9 35 85 01 B1 87
1241:61 91 02 68 85 01 58 E6 09
1249:61 D0 02 E6 62 E6 02 D0 84
1251:02 E6 03 A5 61 05 62 D0 9F
1259:DE A5 02 85 2D A5 03 85 7A
1261:2E A0 05 B9 72 CA 99 04 F6
1269:03 88 10 F7 20 60 A6 4C CE
1271:86 E3 7C A5 1A A7 E4 A7 9C
1279:00 80 DC FF 00 80 DC FF 15
1281:00 00 FF FF 00 00 FF FF A5
1289:00 00 FF FF 00 00 FF FF AD
1291:00 00 32 34 4B 20 52 41 FF
1299:4D 20 44 49 53 4B 00 00 51
12A1:00 00 00 80 00 00 00 00 CD
                             ©
```

Printing Special Characters

George Vogel

A word processor and printer can do much more than meets the eye. Here are three techniques that explore ways you can produce printed text with special characters for use in foreign-language sets or scientific notation. Although the example program is written for Apple II computers, the general technique can be adapted to any personal computer.

Word processors are generally limited in the kind of text they can print. While most can produce underlining, boldface, and superscripts, they are not usually designed to handle special characters such as foreign language or scientific symbols. In fact, your printer may already be capable of printing such special characters, but the process of using them from a word processor is often difficult. This article explores three possible solutions to the problem.

Easy, But Limited

The first, and most obvious, approach may be suitable for foreign-language text. Many printers have one or more international character sets, which can be selected by setting DIP switches as described in the printer manual. The various special characters are then available in place of certain less-essential characters such as brackets, braces, and so on. One limitation of this method is that it works only if your keyboard has keys corresponding to the characters which are replaced with international characters. The Apple II+ keyboard, for instance, lacks many such keys.

Let's look at a fairly typical example. The Star Micronics SG-10 printer, which is in many respects very similar to Epson printers, offers six international character sets: German, Italian, Danish, Swedish, Spanish, and French. Unfortunately, only the first four sets are complete. (The French and Spanish character sets require more letters with diacritical marks than are pro-

vided in the printer character set.) And there are many languages, of course, which can't be represented with any of these character sets. We'll see how those languages might be accommodated later in this article.

The Top 128

A second method of producing special characters involves the highest 128 characters in the ASCII character set. As you may know, the standard ASCII character set uses character codes 0–127 for the typical English alphabet, numerals, and punctuation. Character codes in the range 128–255 have no standard meaning, so they are often used for block graphics and various other special symbols. These sets vary greatly from one printer to another, but it's possible that yours has the characters you need for a particular application.

Although some computers can produce ASCII codes above 127 from the keyboard (often by pressing two keys at once), the Apple II series does not have this capability. How, then, can you use these special codes in a word processor? One method is to sacrifice a seldom-used pair of special printer commands.

To illustrate, say that you have a Star SG-10 printer and you are using the *Word Handler II* word processor on an Apple II computer. This particular word processor can be set up for many different printers because it allows you to specify the control code sequence (often called an *escape code*) which activates each of the printer's special features. Let's assume that you are willing to do without the ability to print superscript characters. As with many such features, using superscript involves two commands: one to turn superscript on and the other to turn it off.

The normal SG-10 escape codes for superscript are $1B $53 $01 and $1B $54 $1B $48. The first character in each code—$1B, or decimal 27—is usually called the *escape* character (hence the term escape code). This character simply serves as a signal that the printer should interpret the following characters as control characters—commands—rather than as something to print literally.

To adapt the word processor for printing the special characters, enter the printer setup feature and replace the normal escape code for *superscript on* with $1B $3E and the code for *superscript off* with $1B $3D. These are the SG-10 printer's commands for *set highest bit to 1* and *set highest bit to 0*. Since the highest bit of an eight-bit character is equal to 128, setting the highest bit to 1 has the effect of adding 128 to the ASCII code. Thus, when the first condition is activated, the printer automatically adds 128 to every character it receives, before it prints that character. This remains in effect until canceled with the second code. If you type the letter *F* (ASCII 70) in this mode, the printer prints ASCII 198 (70 + 128). On the Star printer, character 198 happens to be a degree (°) symbol, so this method offers an easy means of generating that symbol for math or scientific documents. Whenever you need to use that symbol, you must cause the word processor to issue the command it would normally use to activate superscript. You then type an *F*, followed by the command to turn off superscript. The control code goes to the printer, but the word processor treats the *F* as usual, so the screen display shows an *F* rather than the special symbol.

User-Defined Characters

A third, more versatile, approach is possible if you have a dot-matrix printer with the ability to print user-defined characters. These are simply characters whose shape you design on your own. Typically, you design the special character set in a separate operation and store it in a disk file. Before using user-defined characters, you must invoke a special command to read the new character shapes from the disk file into the computer's memory and download, or send, them to the printer.

Downloading new character definitions is possible only on a printer which contains RAM (Random Access Memory) set aside for this purpose. Under normal circumstances, the printer uses the character shapes permanently recorded in a ROM (Read Only Memory) chip. But if you store new shapes in RAM and send the appropriate command, the printer uses the RAM-based shape definitions, instead.

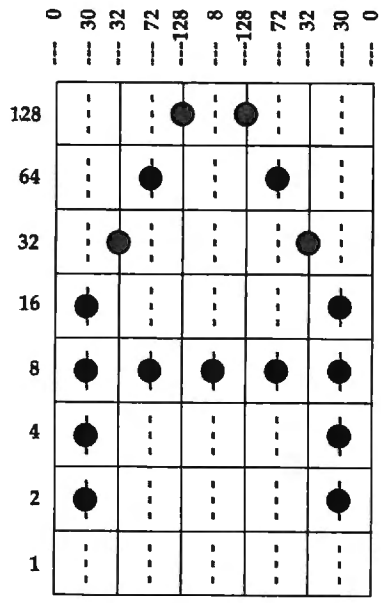
We can use the method described in the previous section to send the command that makes the printer switch to the RAM character set. In the case of the SG-10, for instance, the code to switch to the RAM characters is $1B $24 $01, and the code to switch back to the normal characters in ROM is $1B $20 $00.

The details of this process will vary from one printer to the next, so consult your printer's manual for specific guidance. To give you a general idea of how it's done, we'll describe the procedure for the Star SG-10. On this particular printer, each character is formed out of dots within a matrix that's 8 dots high and 11 dots wide. The two rightmost columns are normally left blank to provide white space between characters.

On Star printers, each vertical line of a character can be described by a byte, or a number in the range 0–255. A character contains 11 vertical lines of dots. Thus, to describe the dot pattern of a character, we must supply a series of 11 numbers, each in the range 0–255.

For each number in the character definition, a dot in the lowest position is equal to 1; the next highest dot, 2; the next, 4; and so on, up to 128 in the topmost position. The figure shows how you might define

Printer Character Pattern



the character *A* for a Star SG-10 printer. The numbers on the left side of the figure indicate the value of a dot in the corresponding column. At the top of each column is the number for that vertical line of the character. This particular character can be defined with the numbers 0, 30, 32, 72, 128, 8, 128, 72, 32, 30, 0.

Three more pieces of information are needed to describe the character completely. The first two define where in the character set this character will appear. For instance, if you wanted the new character to replace the # symbol, ASCII 35, you would supply the number 35. The SG-10 requires a range of positions where the new character is to be installed, so the second value will also be 35. The third value combines information about the width of the character (for proportional spacing, if used) and information to lower the character to produce a descender, or tail. When downloading a user-defined character to this printer, you send these three values first, followed by the 11 numbers which actually define the shape of the character.

The example program follows this procedure to download a number of new characters to the printer. Lines 150–160 are necessary because the Apple printer card cannot transmit eight-bit quantities.

This method allows you to define as many new characters as you need, limited only by the number of characters available from your keyboard. In a few cases, you might want a new character set that completely replaces the normal ROM character set. It's more common to copy most of the ROM characters into RAM and redefine only part of the character set. Once the new characters have been downloaded, you can switch between them and the ROM characters at will.

You may be wondering whether these techniques will work with your particular word processor and printer. The only way to find out is to try. Printers differ very widely in their capabilities, so you will need to read your printer manual carefully before going any further. Most printer manuals include example programs in BASIC which demonstrate that printer's capabilities. Word processors, too, handle special features in different ways. If your word processor allows you to specify the actual control codes for special features, you can implement with it any of the special features that your *printer* can handle. If the word processor is dedicated to specific printers and does not let you change the control codes, your only solution is to use downloaded characters, as long as your printer supports them.

Printer Example

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
A3 10 REM COPYRIGHT 1987 COMPUTE
      ! PUBLICATIONS, INC. ALL R
      IGHTS RESERVED
C1 15 TEXT : HOME : PRINT "COPYR
      IGHT 1987": PRINT "COMPUTE
      ! PUBLICATIONS, INC": PRIN
      T "ALL RIGHTS RESERVED"
7A 17 FOR I = 1 TO 1000: NEXT I
28 20 D$ = CHR$ (4):E$ = CHR$ (2
      7)
9C 30 DIM A(95),M%(14)
7F 40 HOME : INVERSE : PRINT "
         D O W N L O A D . F R E
       N C H      "; NORMAL : PRI
      NT
49 50 PRINT : PRINT "PRINT CHARA
      CTER TABLE? (Y/N) ";: GET
      PR$: PRINT : PRINT
8D 60 PRINT "(E)MPHASIZED?": PRI
      NT "(P)ROPORTIONAL?": PRIN
      T "(D)OUBLE-STRIKE PROPORT
      IONAL? ": PRINT : PRINT "(
      OR ANY OTHER KEY FOR REGUL
      AR) ";: GET PS$: PRINT
1F 70 PRINT : PRINT "DOWNLOADING
       CHARACTERS - WAIT"
95 80 PRINT D$"PR#1": PRINT E$"@
      ";E$"&" CHR$ (0);: REM INI
      T. PRINTER, COPY CHAR. ROM
       TO DL RAM
18 90 ONERR GOTO 190
DB 100 ::: REM : REPLACE REGULAR
       BY CUSTOM CHARS IN DL RA
      M
79 110 FOR K = 1 TO 14: READ M%(
      K): NEXT
5D 120 S = S + 1:A(S) = M%(1)
D9 130 PRINT E$"&" CHR$ (1);
B0 140 FOR K = 1 TO 14
53 150 IF PEEK (49601) > 127 THE
      N 150
C7 160 POKE 49296,M%(K)
EC 170 NEXT K
9B 180 GOTO 110
BF 190 ON PR$ = "Y" GOSUB 260: R
      EM : PRINT CHAR TABLE IF
      REQUESTED
90 200 ::: REM : SET PRINTING ST
      YLE
79 210 IF PS$ = "E" THEN PRINT E
      $"E";: REM EMPHASIZED
0D 220 IF PS$ = "P" THEN PRINT E
      $ CHR$ (112) CHR$ (1);: R
      EM PROPORTIONAL
53 230 IF PS$ = "D" THEN PRINT E
      $"G";E$ CHR$ (112) CHR$ (
      1);: REM DBL.STRIKE/PROPO
      RTIONAL
E0 240 PRINT : PRINT D$"PR# 0":
      END
B2 250 ::: REM : PRINT LIST OF D
      OWNLOAD CHARS WITH THOSE
      ACTIVATING THEM
34 260 PRINT : PRINT D$"PR#1": P
      RINT E$"M" CHR$ (8);:CL =
       32: REM L. MARGIN, CHARS
      /LINE
DD 270 R = S:F = 1:L = CL
C7 280 IF R > CL GOTO 300
69 290 L = F + R - 1
2F 300 FOR J = F TO L: PRINT CHR
      $ (A(J))" ";: NEXT : PRIN
      T : REM CONTROLLING CHARA
      CTERS
32 310 PRINT E$"$" CHR$ (1);: RE
      M DL RAM "ON"
8D 320 FOR J = F TO L: PRINT CHR
      $ (A(J))" ";: NEXT : PRIN
      T : REM DL CHARACTERS
96 330 PRINT E$"$" CHR$ (0): REM
       DL RAM "OFF"
37 340 IF L < > S THEN F = F + C
      L:L = L + CL:R = R - CL:
      GOTO 280
9B 350 PRINT E$"@";: RETURN
4B 1010 DATA 49,49,139,4,10,160,
      10,96,10,32,28,2,0,0: RE
      M a grave
B0 1020 DATA 50,50,139,4,10,96,1
      0,160,10,96,28,2,0,0: RE
      M CIRCUMFLEX
34 1030 DATA 51,51,139,28,34,8,3
      4,72,34,136,34,24,0,0: R
      EM ACCUTE
CB 1040 DATA 52,52,139,28,34,136
      ,34,72,34,8,34,24,0,0: R
      EM GRAVE
17 1050 DATA 53,53,139,28,34,72,
      34,136,34,72,34,24,0,0:
      REM CIRCUMFLEX
0F 1060 DATA 54,54,152,0,98,0,19
      0,0,66,0,0,0,0,0: REM CI
      RCUMFLEX
E3 1070 DATA 55,55,152,0,162,0,6
      2,0,130,0,0,0,0,0: REM D
      IERESIS
7E 1080 DATA 56,56,139,28,34,64,
      34,128,34,64,34,28,0,0:
      REM CIRCUMFLEX
07 1090 DATA 57,57,139,60,2,128,
      2,64,2,0,60,2,0,0: REM G
      RAVE
A8 1100 DATA 48,48,139,60,2,64,2
      ,128,2,64,60,2,0,0: REM
      CIRCUMFLEX
30 1110 DATA 37,37,139,28,34,136
      ,34,8,34,136,34,24,0,0:
      REM DIERESIS
28 1120 DATA 35,35,10,56,68,0,69
      ,0,71,0,68,0,0,0: REM CE
      DILLA
24 1130 DATA 36,36,139,120,132,0
      ,133,0,135,0,132,72,0,0:
       REM CEDILLA
2C 1140 DATA 38,38,152,0,0,64,16
      0,0,160,64,0,0,0,0: REM
      DEGREE
```

Amiga Disk-Based Fonts

Daniel L. Stockton

Expanding on a previous, related article (see "Amiga System Fonts" in the April 1987 issue of COMPUTE!), this article shows you how to load custom text fonts from disk and install them from Amiga BASIC.

In the Utilities drawer of the Amiga Workbench disk is a useful tool called Notepad. One attractive feature of this mini-word processor is its ability to use a variety of text fonts. This article explains how to use those same fonts—or any disk-based text font—in Amiga BASIC. Amiga BASIC does not provide any direct means of loading a custom font from disk. However, this can be accomplished by calling system routines which are used by the computer itself.

Where Are My Fonts?

The fonts used by Notepad are located in the *fonts* directory of the Workbench disk. You can use the FILES command to get a listing of that directory. Type this command in the BASIC output window and press Return:

```
files "workbench:fonts"
```

Each font in the fonts directory has its own subdirectory which contains the various sizes for that font. Font sizes are specified in units called *points*, which are equal to 1/72 inch. Thus, a 9-point font has characters 9/72 inch in size, and so on. The different text styles (italicized, boldface, and so on) are not stored in the font directory; these styles are generated by selectively distorting the shapes found in the basic font file.

The program included with this article can select any of the disk-based fonts, with eight styles for each font. Table 1 lists the fonts, and Table 2 lists the various styles. Version 1.2 of the Amiga operating system adds a few new sizes to existing fonts.

Table 1: Fonts

Fonts	Sizes
Topaz	8, 9, 11
Ruby	8, 12, 15
Diamond	12, 20
Opal	9, 12
Emerald	17, 20
Garnet	9, 16
Sapphire	14, 19

Table 2: Font Styles

0	plain text (Workbench default)
1	underline
2	boldface
3	boldface and underline
4	italics
5	italics and underline
6	italics and bold
7	italics, bold, and underline

Program 1 uses two system libraries named graphics.library and diskfont.library. In order for Amiga BASIC to use these libraries, it must have a file description of the library in a form which it understands. This form is called a *bmap file*. The bmap file is essentially a list of pointers that allow BASIC to access library routines.

Before you can run Program 1, you must make sure that the correct bmap files are present on the same disk as Program 1. The first such file, graphics.bmap, is included in the BASICDemos drawer of the Amiga Extras disk. The second file, named diskfont.bmap, must be created.

This article shows how to access disk-resident character fonts from Amiga BASIC. Two of the Amiga's seven fonts are shown in these photographs.

If you have version 1.2 of the Amiga operating system (available as an inexpensive upgrade from any Amiga dealer), you can create diskfont.bmap quite easily. The BASICDemos disk for 1.2 contains a BASIC program named ConvertFd, as well as a directory named FD1.2. Run the ConvertFd program, using the file named diskfont.lib_fd. When the program is

finished, the disk contains diskfont.bmap. If you haven't obtained the 1.2 upgrade, type in and run Program 2, which reads values from DATA statements, checks them for accuracy, and creates diskfont.bmap on the current disk.

Before you run Program 1, make sure that both graphics.bmap and diskfont.bmap are on the same disk as Program 1. The location of these files is important. They must be either in the current directory *or* in the directory named LIBS on the disk used when you booted the system. The LIBS (LIBrarieS) directory is a good place for bmap files, since their purpose is to give you access to libraries. If you don't have the bmap files in the correct place, BASIC will stop with a *File not found* error when you run Program 1.

When you run the program, it displays the various fonts on the screen in different sizes and waits for you to click the mouse button before proceeding to the next font.

The most important part of Program 1 is contained in the subprogram named FontSelect, which appears at the end of the program. After you have tested Program 1 and have saved a copy, delete everything in the program except the FontSelect subprogram. Then save the subprogram under a new name as an ASCII file, so that it can be MERGEd with other programs. To save a program in ASCII form, add the characters ,a to the end of a normal save command. For instance, to save the subprogram with the name FontSelect, you would type this command in the BASIC output window and press Return:

```
save "FontSelect",a
```

The FontSelect subprogram is invoked with a CALL statement in the main program. Three items of information are passed to FontSelect in the form of variables.

The first variable is a string named font$. This string contains the name of the font you wish to use (Garnet, Ruby, and so on). If the string is a null string (" ", a string containing no characters), only a style change occurs.

The second and third variables passed to FontSelect are numeric variables of the short integer type. The variable height% defines the point size that you wish to use (see Table 1), and style% defines the style to use (Table 2).

Opening And Closing Libraries

A few additional statements are needed to prepare for the CALL to FontSelect, and to clean up afterward. In Program 1, the preparatory statements are grouped together at the beginning, immediately after the first two REMark statements. The DEFLNG statement causes all simple variables to default to the long integer type. (Note that this declaration is overridden by the short integer type specifier attached to height% and style%.)

The LIBRARY and DECLARE FUNCTION statements actually give you access to library routines. These statements should appear in the initialization section of the program, before the first CALL to the FontSelect subprogram.

When the program is about to terminate, you should take some additional steps to close the fonts and the libraries. The CALL to the CloseFont function closes any fonts that were previously opened. (A bug in version 1.1 of the operating system prevents this CALL from working correctly. Version 1.2 corrects the bug. If you have version 1.1, omit the line containing the CALL to CloseFont or put a REM in front of the line.)

The final CALL to FontSelect resets the font to the system font, Topaz. While not absolutely necessary, it's considered good manners for programs which change the computer environment to restore the original environment as closely as possible before terminating.

The LIBRARY CLOSE statement closes libraries that were previously opened. If you omit these final housekeeping chores, the computer may not crash, but the libraries will remain open, wastefully occupying memory which would otherwise be freed for other tasks.

The program module named TestSection uses another system routine named *text* when printing words in boldface and italicized styles. This method prevents the characters in those words from overlapping, as they would if you printed the words with PRINT.

Program 1: Amiga Disk-Based Fonts

For instructions on entering these programs, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
REM Copyright 1987 COMPUTE! Publ
ications, Inc.
REM All Rights Reserved
REM Program  SelectFont
REM Provides for use of Amiga di
sk based fonts from Amiga Basic
PRINT "Copyright 1987 COMPUTE! P
ublications, Inc."
PRINT "All Rights Reserved"
DEFLNG a-z          'all variables
default to long integer
'Include optional CHDIR command
here (CHDIR ":BMAPS")
LIBRARY "diskfont.Library"
LIBRARY "graphics.Library"
DECLARE FUNCTION OpenDiskFont LI
BRARY
DECLARE FUNCTION AskSoftStyLe LI
BRARY
'the above commands must be plac
ed in the main body of your prog
ram

TestSection:
BREAK ON
ON BREAK GOSUB Housekeeping
READ t1$,t2$
FOR i=0 TO 6  'look at 7 fonts
READ font$,height%
CALL FontSeLect(font$,height%,st
yLe%)
FOR styLe%=0 TO 4 STEP 2   'look
at 3 styles each
CALL FontSeLect("",0,styLe%)
IF styLe%=0 THEN
LOCATE 1,1:PRINT  "This is the";
height%;"point ";font$;" font"
PRINT  "Click left mouse button
for the next font"
PRINT t1$:PRINT t2$
ELSEIF styLe%=2 THEN
a$="YOU ARE LOOKING AT BOLD STYL
E"
PRINT
CALL text(WINDOW(8),SADD(a$),LEN
(a$)):PRINT
ELSEIF styLe%=4 THEN
a$="THIS IS ITALICS STYLE"
PRINT
CALL text(WINDOW(8),SADD(a$),LEN
(a$)):PRINT
END IF
NEXT styLe%

WaitForMouse:
IF NOT MOUSE(0) THEN WaitForMous
e
CLS
'include the following CALL stat
ement when using workbench 1.2
'as it closes fonts and frees me
mory
'CALL CloseFont(WINDOW(8),fontpt
r)
NEXT i

Housekeeping:
CALL FontSeLect("topaz",8,0)  'r
eturn to default system font
LIBRARY CLOSE
END

DATA ABCDEFGHIJKLMNOPQRSTUVWXYZ0
1234567891@#$%
DATA abcdefghijklmnopqrstuvwxyz^
&*()=+\|/?<[{
```

```
DATA topaz,9,ruby,12,diamond,12,
opal,11
DATA emerald,20,garnet,16,sapphi
re,19
REM End Of TestSection
SUB FontSeLect(font$,height%,sty
Le%) STATIC
IF font$ <> "" THEN
textAttr(0)=SADD(font$+".font"+C
HR$(0))
textAttr(1)=height%*65536&
fontptr=OpenDiskFont(VARPTR(text
Attr(0)))
IF fontptr THEN SetFont WINDOW(8
),fontptr
END IF
permited%=AskSoftStyLe&(WINDOW(8
))
CALL SetSoftStyle (WINDOW(8),sty
Le%,permited%)
END SUB
```

Program 2: DiskFont.bmap Filemaker

```
REM Copyright 1987 COMPUTE! Publ
ications, Inc.
REM All Rights Reserved
' DiskfontMaker. Creates Diskfon
t.bmap file on disk.
PRINT "Copyright 1987 COMPUTE! P
ublications, Inc.
PRINT "All Rights Reserved"
fiLe$="Diskfont.bmap"
READ fiLesize,checksum
PRINT "Checking data statements.
.."
PRINT
FOR j=1 TO fiLesize
READ a$
a=VAL("&H"+a$)
check=check+a
NEXT j
RESTORE DiskFontData
IF check=checksum THEN GOTO OK
PRINT "Typing error: Check DATA
statements."
END

OK:
PRINT "Creating Diskfont.bmap fi
le..."
ON ERROR GOTO CreationError
OPEN fiLe$ FOR OUTPUT AS #1
FOR j=1 TO fiLesize
READ a$
a=VAL("&H"+a$)
PRINT #1,CHR$(a);
NEXT j
CLOSE
PRINT "Finished"
END

CreationError:
PRINT "Error ";ERR
END

DATA 34, 3196
DiskFontData:
DATA  4f,70,65,6e,44,69,73,6b
DATA  46,6f,6e,74,00,ff,e2,09
DATA  00,41,76,61,69,6c,46,6f
DATA  6e,74,73,00,ff,dc,09,01
DATA  02,00
```

Atari NoDOS

Emmanuel Gendrano

Everyone knows it's impossible to save an Atari BASIC program to disk without DOS—or is it? Although you may not need to use it often, this program can be a real lifesaver on occasions where DOS becomes corrupted, or when you enter BASIC without first booting DOS.

One of the most fearsome messages an Atari BASIC programmer can receive is ERROR 130, the unknown device error. If this message appears when you attempt to save a BASIC program to disk, it usually means that the system doesn't recognize the disk drive, leaving you no way to save your work. This sometimes happens when misdirected POKEs or other accidents corrupt the DOS (Disk Operating System) in memory. It can also occur if you forget to boot DOS when you turn on the computer.

Since the purpose of DOS is to perform disk operations, it seems logical that you can't save a BASIC program if DOS isn't present. But "Atari NoDOS" can perform this seemingly impossible task.

Using Atari NoDOS

Atari NoDOS consists of three short sections: one that formats a disk without DOS, one that saves a program without DOS, and a third that reloads a program saved in NoDOS format. Type in the entire program and save it to disk. If you have a cassette drive, you may want to LIST the program to tape, as well. (If you have a cassette drive, you can always try to CSAVE a program when DOS is absent. However, Atari cassette drives are much less reliable than disk drives, and using NoDOS is also much faster than saving to cassette.)

Of course, if you're in a situation where you need Atari NoDOS, you won't be able to ENTER the program from disk. At this point, you can either ENTER it from cassette, if you have a cassette drive, or simply type the program from the magazine listing. The program is so short that it can be typed in only a few minutes.

If you wish to save the program to an already-formatted disk, you need only type lines 32760–32767 of Atari NoDOS. If you need to format a new disk before performing the save, you must type the formatting section of the program (lines 32755–32757) as well as the saving section (32760–32767). Of course, if you ENTER the program from cassette, all of the lines are present.

To save a program on an already-formatted disk, type GOTO 32760 and press RETURN. To format a disk and then save the program, type GOTO 32755 and press RETURN; the formatting routine jumps to the saving routine after the disk is formatted.

The saving routine prompts you to enter a single-density disk in drive 1 and press RETURN when you are ready. The disk should have enough room to store the program, but it need not be completely empty. Atari NoDOS doesn't save the program in the usual manner. Instead, it writes the program on disk beginning with sector 719 and proceeding to lower-numbered sectors. As you may know, single-density Atari disks contain 720 sectors, and program storage normally begins with lower-numbered sectors and works upward. By writing to the highest sectors on the disk, NoDOS minimizes the likelihood of overwriting information already on the disk. However, it does not check to see whether a given sector already contains information, so you should not attempt to use this program with a disk that is almost full.

The screen blanks out while Atari NoDOS saves the program. This is done to gain a little extra speed. Just as in a normal disk save, you will see the drive's busy light come on. When the screen is restored, the program displays a status number to indicate the success of the operation. If this number is anything other than 1, an error occurred; your disk drive manual explains the meaning of the error number.

If the save was successful, you should turn off the computer, insert a disk containing DOS, and turn the computer back on. Don't expect to see the program saved using NoDOS in the disk directory. It has been saved in a different format which doesn't appear in the directory.

To convert the program to normal form, you must now load the reloader section of Atari NoDOS. Again, if you saved the entire program to disk, simply reload it. Since the reloader is the first section of Atari NoDOS, you can start it by typing either RUN or GOTO 32740. When the program prompts you to do so, insert the disk on which you saved the program without DOS; then hit RETURN. Each line is displayed on the screen as it is entered into memory. If you wish to abort the process, simply press a key. When the process is complete, you should immediately save the program in the usual way.

This program can be modified to work with double-density disks, as well. Change the first two POKEs in line 32767 to POKE 776,0 and POKE 777,1. You may or may not have to change the same POKEs in the reloading routine, depending on your drive.

If you get consistent errors when using the saving routine, your drive may be misaligned—a problem which requires a professional cure. However, you may be able to save the program by changing Atari NoDOS to write to the lowest sectors on the disk rather than the highest. The lower sectors are much easier to read, even for a misaligned drive. Note that this method can be used *only with a completely blank disk*, since it is certain to destroy at least the first file on the disk. To make this modification, change the statement SEC=719 to SEC=1 in lines 32761 and 32742. You must also change the statement SEC=SEC−1 to SEC=SEC+1 in lines 32767 and 32752. These changes cause the program to begin at sector 1 and use higher-numbered sectors as the save proceeds.

Atari NoDOS

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
KP 0 REM COPYRIGHT 1987 COMP
     UTE! PUBLICATIONS, INC.
       ALL RIGHTS RESERVED
MD 1 PRINT CHR$(125);"COPYRI
     GHT 1987":PRINT "COMPUT
     E! PUBLICATIONS, INC."
DL 2 PRINT "ALL RIGHTS RESER
     VED."
BF 3 FOR X=1 TO 1000:NEXT X
IP 32739 REM ---------------
HG 32740 REM  NoDOS  Saves a
          BASIC program from
          memory to disk wit
         hout any DOS presen
         t.  10:24AM/7-18-86
ND 32741 GRAPHICS 0:CLR :DIM
          A$(1):? :? "  NoDOS
         /Reloader ":? "Inse
         rt disk in drive 1
         ";:TRAP 32740
AD 32742 INPUT A$:BUF=1536:S
         EC=719:I=128:POKE 8
         2,0:POKE 83,39
BC 32743 POKE 842,12:L=0:? "
         {CLEAR}{TAB}HIT ANY
          KEY TO ABORT{DOWN}
         ":POKE 766,1
FN 32744 GOSUB 32749:? CHR$(
         D);:L=L+1:IF D=155
         THEN 32747
DC 32745 IF PEEK(764)<>255 T
         HEN POKE 82,2:POKE
         766,0:POKE 764,255:
         END
ED 32746 GOTO 32744
OP 32747 POKE 766,0:IF L=1 T
         HEN POKE 82,2:END
HC 32748 ? "{DOWN}G.32743":P
         OKE 842,13:POSITION
          0,0:STOP
BK 32749 IF I=128 THEN GOSUB
          32751
MI 32750 D=PEEK(BUF+I):I=I+1
         :RETURN
JB 32751 POKE 768,49:POKE 76
         9,1:POKE 770,82:POK
         E 771,64:POKE 772,A
         SC(CHR$(BUF)):POKE
         773,INT(BUF/256):PO
         KE 776,128
MK 32752 POKE 777,0:POKE 778
         ,ASC(CHR$(SEC)):POK
         E 779,INT(SEC/256):
         D=USR(ADR("hLY■")):
         SEC=SEC-1:I=0:RETUR
         N
IM 32754 REM ---------------
NM 32755 ? :? "  NoDOS/Format
         ■ drive #1.":? "Ent
         er 'Y' when ready "
         ;:CLR :DIM A$(1):IN
         PUT A$
ED 32756 IF A$<>"Y" THEN 327
         60
BE 32757 POKE 768,49:POKE 76
         9,1:POKE 770,33:POK
         E 771,0:POKE 774,25
         5:D=USR(ADR("hLY■")
         )
JB 32759 REM ---------------
MB 32760 ? :? "  NoDOS/Saver:
          Hit RETURN to star
         t ";:CLR :DIM B$(25
         5),A$(127):INPUT A$
         :TRAP 40000:CLOSE #
         1:POKE 82,0
CN 32761 POKE 83,39:BUF=ADR(
         B$):I=1:SEC=719:OPE
         N #1,13,0,"E:":POKE
          752,1:POKE 559,0:A
         =PEEK(136)+256*PEEK
         (137)
HB 32762 ? "{CLEAR}";:L=PEEK
         (A)+256*PEEK(A+1):L
         IST L:POSITION 0,1:
         INPUT #1,A$:A$(LEN(
         A$)+1)=CHR$(155)
EN 32763 B$(I)=A$:I=LEN(B$)+
         1:A=A+PEEK(A+2):IF
         L<32739 AND I<=128
         THEN 32762
BP 32764 IF I>128 THEN GOSUB
          32766:B$(I)="#":B$
         =B$(129):I=LEN(B$):
         B$(I)="":IF PEEK(77
         1)=1 AND L<32739 TH
         EN 32762
AP 32765 B$(129)="#":B$(I)=C
         HR$(155):B$(I+1)=B$
         (I):GOSUB 32766:POK
         E 82,2:GRAPHICS 0:?
          "{BELL}STATUS - ";
         PEEK(771):END
PJ 32766 POKE 768,49:POKE 76
         9,1:POKE 770,87:POK
         E 771,128:POKE 772,
         ASC(CHR$(BUF)):POKE
          773,INT(BUF/256)
ME 32767 POKE 776,128:POKE 7
         77,0:POKE 778,ASC(C
         HR$(SEC)):POKE 779,
         INT(SEC/256):D=USR(
         ADR("hLY■")):SEC=SE
         C-1:RETURN
```

Fast Fractal Landscapes For IBM

Paul W. Carlson

It doesn't take an expensive mainframe computer to create realistic-looking fractal landscapes. Using the program included in this article, you can create your own landscapes on any IBM PC or compatible computer with a color graphics adapter.

Some recent magazine articles about fractals have included pictures of computer-generated landscapes so realistic that they appear to be photographs of nature itself. Although "Fast Fractal Landscapes" doesn't produce displays that resemble photographs, it does create realistic landscapes in the sense that all of the scenes resemble nature. And best of all—it creates them in about half a minute.

To get started, type in and save the program listed on the following page. Before you run the program, make sure you have a disk in the active drive with at least 20,000 bytes free. Now run the program. The program will create a file on the disk called FRACLAND.COM. To run FRACLAND.COM, first type SYSTEM to get out of BASIC. At the DOS prompt, type FRACLAND and press Enter. You'll get a message that elevations are being computed. The plot will begin a few seconds later. When a plot is complete, press any key except the Q key to start another plot. Pressing the Q key exits the program.

The program creates the displays so fast that you might think that all the landscapes are stored within the program. The truth is, however, that the program is creating them randomly. The program uses what might be described as "controlled randomness" to create over 30,000 different landscapes. If you don't get a landscape that you like when you first run the program, keep trying—you'll soon get one that you like.

Some of the plots may be almost all land and others may be almost all water. In fact, it's possible that some might be all water. Overall, the amount of land and water should be close to equal. The plots have the vertical scale exaggerated to keep some separation between the lines that form the land. This, along with the colors used, produces landscapes that look like rugged terrain in winter at sunset.

How It Works

The program employs a technique that is often used when dealing with fractals: repeatedly subjecting an object (which can be a line, shape, or almost anything) to a series of operations at an ever-decreasing scale. In the language of fractals, the initial object is called the *initiator* and the series of operations is called the *generator*.

The initiator in this program is a square representing a flat area of land initially at sea level, or zero elevation. Here is a description of the operations performed on this square and on all subsequent squares:

1. A new elevation is computed for the midpoint of each side of the square by averaging the elevations of the corners on either side of the midpoint.

2. Each new midpoint elevation is increased or decreased by a random amount. The random amount is computed so that its maximum absolute value is proportional to the length of the side containing the midpoint. This keeps the height of the landforms in natural proportion to their breadth.

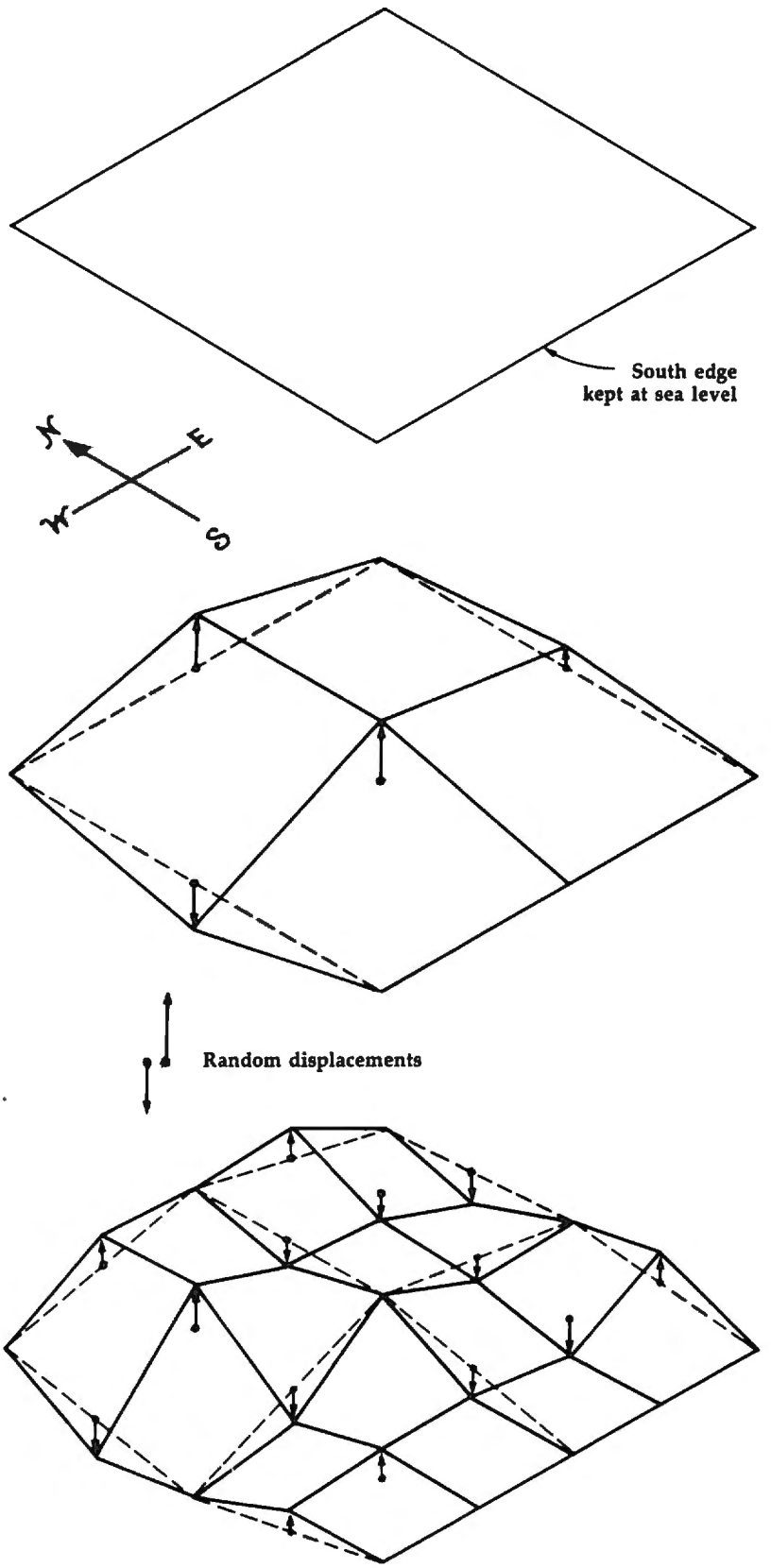
3. The elevation of the point at the center of the square is computed in the same manner using the elevations of a pair of diagonally opposite corners, the pair used being chosen randomly.

4. The square is subdivided into four squares, the midpoints and center of the old square becoming corners of the new squares.

5. The process is then repeated for each of the new squares. Only when all the squares at any level of subdivision have been processed does the computation move on to the next level. When the desired level of subdivision has been accomplished, the process stops. The process of subdivision is illustrated in the accompanying figure.

If the elevations were to be plotted without further processing, the resultant landscape would resemble a scene on the moon with very jagged mountain peaks. To make the landscape more earthlike, the elevations are smoothed by averaging each elevation with the adjacent east and west elevations (north being the top of the screen). This gives the landscapes a more

Landscape Subdivision



rounded, eroded appearance.

The display is created by plotting cross sections from west to east beginning with the southernmost cross section. Each cross section is plotted by connecting the north sides of adjacent squares end to end and removing hidden lines. Before a line segment on a cross section is plotted, the program checks if the segment is going into or coming out of the water. If it is, the segment is adjusted to end at the water level (elevation zero). Any negative elevations are set to elevation zero. Segments with both endpoints at elevation zero are plotted as water. The elevations at the south edge of the plot are kept at sea level to keep the first cross section from slicing through any mountains.

Fast Fractal Landscapes

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue

```
EG 1 ' Program to create FRACLAN
     D.COM
OH 2 '
NC 3 ' Copyright 1987 Compute! P
     ublications, Inc.
DL 4 ' All Rights Reserved
HP 5 CLS:PRINT "Copyright 1987 C
     ompute! Publications, Inc."
PN 6 PRINT"All Rights Reserved"
DO 10 OPEN "FRACLAND.COM" FOR OU
     TPUT AS 1
NA 20 PRINT:PRINT"CREATING FRACL
     AND.COM FILE, PLEASE WAIT.
     ..";
JM 30 PRINT#1,CHR$(&HE9);CHR$(&H
     CA);CHR$(&H44);
BN 40 FOR N=1 TO 17577:PRINT#1,C
     HR$(0);:NEXT
AD 50 T=0:FOR J=1 TO 2049:READ A
     $:N=VAL("&H"+A$)
FP 60 T=T+N:PRINT#1,CHR$(N);:NEX
     T:CLOSE 1:PRINT
HN 70 IF T=202520! THEN PRINT"FR
     ACLAND.COM SUCCESSFULLY CR
     EATED!":END
LP 80 PRINT CHR$(7);"***** ERROR
      IN DATA STATEMENTS *****"
     :END
OA 90 DATA 00,00,00,00,00,00,00,
     00,00,43
PF 100 DATA 6F,6D,70,75,74,69,6E
     ,67,20,65
MD 110 DATA 6C,65,76,61,74,69,6F
     ,6E,73,2E
HJ 120 DATA 2E,2E,24,E8,87,07,A1
     ,2B,01,A3
KF 130 DATA 2D,01,C7,06,2F,01,00
     ,00,E8,54
HI 140 DATA 07,B8,02,00,50,E8,41
     ,07,83,C4
NG 150 DATA 02,8D,16,B5,45,B4,09
     ,CD,21,C7
FD 160 DATA 06,27,01,00,00,8B,1E
     ,27,01,D1
QP 170 DATA E3,C7,87,31,01,C8,00
     ,FF,06,27
GD 180 DATA 01,81,3E,27,01,40,01
     ,7C,E8,C7
```

```
HG 190 DATA 06,0F,01,40,00,C7,06
       ,09,01,00
NC 200 DATA 00,A1,0F,01,A3,0D,01
       ,E9,0B,02
LK 210 DATA C7,06,07,01,00,00,A1
       ,09,01,03
DD 220 DATA 06,0D,01,D1,F8,A3,13
       ,01,A1,0F
EK 230 DATA 01,A3,0B,01,E9,DB,01
       ,A1,07,01
PH 240 DATA 03,06,0B,01,D1,F8,A3
       ,11,01,B8
DP 250 DATA 82,00,F7,2E,09,01,8B
       ,F0,FF,36
MN 260 DATA 0F,01,E8,80,06,83,C4
       ,02,8B,1E
HJ 270 DATA 07,01,D1,E3,8B,C8,8B
       ,80,B1,03
CB 280 DATA 8B,1E,0B,01,D1,E3,03
       ,80,B1,03
HG 290 DATA 99,2B,C2,D1,F8,03,C1
       ,8B,1E,11
GE 300 DATA 01,D1,E3,89,80,B1,03
       ,88,82,00
ID 310 DATA F7,2E,0D,01,8B,F0,FF
       ,36,0F,01
PH 320 DATA E8,46,06,83,C4,02,8B
       ,1E,07,01
DE 330 DATA D1,E3,8B,C8,8B,80,B1
       ,03,8B,1E
PD 340 DATA 0B,01,D1,E3,03,80,B1
       ,03,99,2B
GD 350 DATA C2,D1,F8,03,C1,8B,1E
       ,11,01,D1
PD 360 DATA E3,89,80,B1,03,83,3E
       ,07,01,00
PE 370 DATA 74,44,8B,36,07,01,D1
       ,E6,FF,36
HG 380 DATA 0F,01,E8,0B,06,83,C4
       ,02,8B,C8
NH 390 DATA B8,82,00,F7,2E,0D,01
       ,8B,D8,B8
BD 400 DATA 82,00,F7,2E,09,01,8B
       ,88,B1,03
MA 410 DATA 8B,D8,8B,80,B1,03,03
       ,C7,99,2B
PF 420 DATA C2,D1,F8,03,C1,8B,C8
       ,88,82,00
MA 430 DATA F7,2E,13,01,8B,D8,89
       ,88,B1,03
JE 440 DATA 8B,36,0B,01,D1,E6,FF
       ,36,0F,01
PH 450 DATA E8,C4,05,83,C4,02,8B
       ,C8,B8,82
AC 460 DATA 00,F7,2E,0D,01,8B,D8
       ,88,82,00
CF 470 DATA F7,2E,09,01,8B,B8,B1
       ,03,8B,D8
EI 480 DATA 8B,80,B1,03,03,C7,99
       ,2B,C2,D1
BC 490 DATA F8,03,C1,8B,C8,B8,82
       ,00,F7,2E
FI 500 DATA 13,01,8B,D8,89,88,B1
       ,03,E8,B1
ID 510 DATA 05,A8,01,74,60,FF,36
       ,0F,01,E8
JP 520 DATA 7F,05,83,C4,02,8B,C8
       ,D1,E0,03
JL 530 DATA C1,8B,1E,0B,01,D1,E3
       ,8B,C8,B8
HA 540 DATA 82,00,F7,2E,0D,01,8B
       ,F8,8B,B1
NM 550 DATA B1,03,8B,1E,07,01,D1
       ,E3,8B,D0
FA 560 DATA B8,82,00,8B,FA,F7,2E
       ,09,01,89
BM 570 DATA 7E,FE,8B,F8,8B,81,B1
       ,03,03,46
FJ 580 DATA FE,03,C1,99,2B,C2,D1
       ,F8,8B,1E
LN 590 DATA 11,01,D1,E3,8B,C8,B8
       ,82,00,F7
PG 600 DATA 2E,13,01,8B,F8,89,89
       ,B1,03,EB
KB 610 DATA 5A,FF,36,0F,01,E8,1F
       ,05,83,C4
BM 620 DATA 02,8B,C8,D1,E0,03,C1
       ,8B,1E,0B
BJ 630 DATA 01,D1,E3,8B,C8,B8,82
       ,00,F7,2E
DF 640 DATA 09,01,8B,F0,8B,80,B1
       ,03,8B,1E
MD 650 DATA 07,01,D1,E3,8B,D0,B8
       ,82,00,8B
NF 660 DATA F2,F7,2E,0D,01,8B,F8
       ,8B,81,B1
ML 670 DATA 03,03,C6,03,C1,99,2B
       ,C2,D1,F8
PD 680 DATA 8B,1E,11,01,D1,E3,8B
       ,C8,B8,82
JF 690 DATA 00,F7,2E,13,01,8B,F0
       ,89,88,B1
FA 700 DATA 03,A1,0F,01,01,06,07
       ,01,01,06
AE 710 DATA 0B,01,83,3E,0B,01,40
       ,7F,03,E9
FO 720 DATA 1B,FE,A1,0F,01,01,06
       ,09,01,01
IC 730 DATA 06,0D,01,83,3E,0D,01
       ,40,7F,03
AC 740 DATA E9,EB,FD,D1,3E,0F,01
       ,83,3E,0F
GN 750 DATA 01,02,7C,03,E9,CE,FD
       ,C7,06,15
IB 760 DATA 01,00,00,C7,06,17,01
       ,01,00,B8
NH 770 DATA 82,00,F7,2E,17,01,8B
       ,F0,A1,15
IL 780 DATA 01,D1,E0,03,F0,8B,84
       ,2F,03,03
GL 790 DATA 84,33,04,03,84,B1,03
       ,99,B9,03
IF 800 DATA 00,F7,F9,89,84,B1,03
       ,FF,06,17
DO 810 DATA 01,83,3E,17,01,40,7C
       ,CF,FF,06
AB 820 DATA 15,01,83,3E,15,01,40
       ,7E,BE,E8
HA 830 DATA A1,04,B8,04,00,50,E8
       ,8E,04,83
OB 840 DATA C4,02,C7,06,05,01,00
       ,00,B9,03
NP 850 DATA 00,89,0E,03,01,B8,C7
       ,00,50,2B
NN 860 DATA C0,50,B8,01,00,50,2B
       ,C0,50,E8
KI 870 DATA 15,03,83,C4,08,B8,C7
       ,00,50,B8
GP 880 DATA 3C,01,50,B8,01,00,50
       ,B8,3C,01
FK 890 DATA 50,E8,FF,02,83,C4,08
       ,B8,01,00
OH 900 DATA 50,B8,3C,01,50,B8,01
       ,00,50,2B
OE 910 DATA C0,50,E8,EA,02,83,C4
       ,08,B8,C7
BD 920 DATA 00,50,B8,3C,01,50,B8
       ,C7,00,50
LB 930 DATA 2B,C0,50,E8,D5,02,83
       ,C4,08,C7
FC 940 DATA 06,27,01,00,00,8B,1E
       ,27,01,D1
AD 950 DATA E3,C7,87,31,01,C8,00
       ,FF,06,27
GH 960 DATA 01,81,3E,27,01,40,01
       ,7C,E8,C7
BM 970 DATA 06,15,01,00,00,83,3E
       ,15,01,40
KK 980 DATA 7E,03,E9,5E,02,C7,06
       ,05,01,00
KC 990 DATA 00,C7,06,27,01,00,00
       ,83,3E,27
OH 1000 DATA 01,02,7D,1A,C7,06,2
        5,01,00,00
EN 1010 DATA C7,06,29,01,01,00,8
        3,3E,15,01
PM 1020 DATA 40,75,0D,83,3E,05,0
        1,00,74,06
DJ 1030 DATA FF,06,15,01,EB,C3,C
        7,06,17,01
OP 1040 DATA 00,00,A1,17,01,8B,C
        8,D1,E0,D1
KE 1050 DATA E0,03,C1,A3,21,01,0
        B,C9,75,03
BO 1060 DATA E9,10,01,83,F9,3F,7
        C,03,E9,08
OP 1070 DATA 01,83,3E,15,01,40,7
        C,03,E9,FE
AG 1080 DATA 00,83,3E,05,01,00,7
        5,03,E9,8A
JL 1090 DATA 00,B8,82,00,F7,E9,8
        B,F0,A1,15
KE 1100 DATA 01,D1,E0,03,F0,8B,8
        C,31,03,03
GI 1110 DATA BC,2F,03,0B,FF,7E,2
        7,8B,84,B3
CN 1120 DATA 03,03,84,B1,03,89,4
        6,FE,0B,C0
KA 1130 DATA 7D,18,8B,C8,D1,E0,D
        1,E0,03,C1
OH 1140 DATA 99,8B,8C,B3,03,03,8
        C,B1,03,2B
FL 1150 DATA CF,F7,D9,E9,AD,00,B
        8,82,00,F7
OP 1160 DATA 2E,17,01,8B,F0,A1,1
        5,01,D1,E0
CM 1170 DATA 03,F0,8B,84,35,04,0
        3,84,33,04
GJ 1180 DATA 0B,C0,7F,03,E9,94,0
        0,8B,BC,B3
PF 1190 DATA 03,03,BC,B1,03,78,0
        3,E9,87,00
IJ 1200 DATA 8B,C7,8B,C8,D1,E0,D
        1,E0,03,C1
HB 1210 DATA 99,8B,8C,B3,03,2B,8
        C,35,04,2B
KK 1220 DATA 8C,33,04,03,8C,B1,0
        3,EB,64,B8
MF 1230 DATA 82,00,F7,2E,17,01,8
        B,F0,A1,15
EN 1240 DATA 01,D1,E0,03,F0,83,B
        C,2F,03,00
FH 1250 DATA 7E,1E,83,BC,B1,03,0
        0,7D,17,8B
GM 1260 DATA 84,B1,03,8B,C8,D1,E
        0,D1,E0,03
NC 1270 DATA C1,99,8B,8C,2F,03,2
        B,8C,B1,03
AE 1280 DATA EB,2F,B8,82,00,F7,2
        E,17,01,8B
EI 1290 DATA F0,A1,15,01,D1,E0,0
        3,F0,83,BC
IC 1300 DATA 33,04,00,7E,1E,83,B
        C,B1,03,00
SD 1310 DATA 7D,17,8B,84,B1,03,8
        B,C8,D1,E0
OB 1320 DATA D1,E0,03,C1,99,2B,8
        C,33,04,F7
HJ 1330 DATA F9,01,06,21,01,83,3
        E,05,01,00
QJ 1340 DATA 74,31,8B,1E,15,01,D
        1,E3,B8,82
PD 1350 DATA 00,F7,2E,17,01,8B,F
        0,8B,80,B1
CM 1360 DATA 03,8B,1E,15,01,D1,E
        3,8B,C8,B8
ED 1370 DATA 82,00,F7,2E,17,01,8
        B,F0,8B,80
DE 1380 DATA B3,03,03,C1,99,2B,C
        2,D1,F8,EB
PJ 1390 DATA 13,8B,1E,15,01,D1,E
        3,B8,82,00
HF 1400 DATA F7,2E,17,01,8B,F0,8
```

```
        B,80,B1,03
FP 1410 DATA A3,23,01,0B,C0,7D,0
        6,C7,06,23
OM 1420 DATA 01,00,00,83,3E,23,0
        1,00,75,0C
OH 1430 DATA 83,3E,25,01,00,75,0
        5,B8,01,00
DE 1440 DATA EB,03,B8,03,00,A3,0
        3,01,A1,23
PN 1450 DATA 01,A3,25,01,A1,21,0
        1,A3,1D,01
JM 1460 DATA A1,15,01,D1,E0,03,0
        6,05,01,03
FJ 1470 DATA 06,23,01,40,A3,1F,0
        1,83,3E,29
PD 1480 DATA 01,00,74,15,B8,01,0
        0,A3,1D,01
JJ 1490 DATA A3,19,01,A1,1F,01,A
        3,1B,01,C7
FD 1500 DATA 06,29,01,00,00,83,3
        E,15,01,00
LH 1510 DATA 75,07,83,3E,05,01,0
        0,74,16,FF
MC 1520 DATA 36,1F,01,FF,36,1D,0
        1,FF,36,1B
HN 1530 DATA 01,FF,36,19,01,E8,7
        B,00,83,C4
BD 1540 DATA 08,A1,1D,01,A3,19,0
        1,A1,1F,01
KN 1550 DATA A3,1B,01,FF,06,17,0
        1,83,3E,17
JE 1560 DATA 01,40,7D,03,E9,F3,F
        D,83,3E,05
JH 1570 DATA 01,00,75,05,B8,01,0
        0,EB,02,2B
KH 1580 DATA C0,A3,05,01,FF,06,2
        7,01,E9,AE
PL 1590 DATA FD,C7,06,03,01,02,0
        0,C7,06,1B
OF 1600 DATA 01,81,00,FF,36,1B,0
        1,B8,3B,01
FN 1610 DATA 50,FF,36,1B,01,B8,0
        1,00,50,E8
MO 1620 DATA 27,00,83,C4,08,FF,0
        6,1B,01,81
AD 1630 DATA 3E,1B,01,C7,00,7C,D
        E,E8,E0,01
EG 1640 DATA 0B,C0,75,03,E9,1B,F
        A,B8,02,00
E8 1650 DATA 50,E8,5F,01,83,C4,0
        2,E8,65,01
I8 1660 DATA C3,55,8B,EC,83,EC,1
        2,57,56,B8
LJ 1670 DATA C7,00,2B,46,06,89,4
        6,06,B8,C7
MH 1680 DATA 00,2B,46,0A,89,46,0
        A,8B,76,08
GO 1690 DATA 2B,76,04,39,46,06,7
        D,28,89,46
IP 1700 DATA F4,8B,46,06,89,46,0
        A,8B,46,F4
OC 1710 DATA 89,46,06,8B,46,08,8
        9,46,F4,8B
PB 1720 DATA 46,04,89,46,08,8B,4
        6,F4,89,46
O8 1730 DATA 04,C7,46,F6,FF,FF,E
        B,05,C7,46
BP 1740 DATA F6,01,00,8B,46,06,2
        B,46,0A,89
DG 1750 DATA 46,FC,C7,46,F2,FF,F
        F,C7,46,FA
IF 1760 DATA 00,00,3B,C6,7E,10,C
        7,46,FA,01
JM 1770 DATA 00,89,76,F4,8B,F0,8
        B,46,F4,89
CO 1780 DATA 46,FC,8B,46,FC,D1,E
        0,89,46,EE
KA 1790 DATA 8B,46,FC,2B,C6,D1,E
        0,89,46,F8
QI 1800 DATA 8B,46,EE,2B,C6,89,4
        6,F0,46,EB
DF 1810 DATA 6C,8B,7E,04,D1,E7,8
        1,C7,31,01
GM 1820 DATA 8B,05,39,46,06,7D,1
        F,8B,46,06
DD 1830 DATA 89,05,83,3E,03,01,0
        3,75,07,83
BL 1840 DATA 3E,05,01,00,75,0C,F
        F,76,04,FF
FK 1850 DATA 76,06,E8,F1,00,83,C
        4,04,83,7E
FH 1860 DATA FA,00,74,08,8B,46,F
        2,01,46,06
GP 1870 DATA EB,06,8B,46,F6,01,4
        6,04,83,7E
DI 1880 DATA F0,00,7D,08,8B,46,E
        E,01,46,F0
IE 1890 DATA EB,1A,8B,46,F8,01,4
        6,F0,83,7E
HB 1900 DATA FA,00,74,08,8B,46,F
        6,01,46,04
LO 1910 DATA EB,06,8B,46,F2,01,4
        6,06,4E,0B
BE 1920 DATA F6,75,90,5E,5F,8B,E
        5,5D,C3,55
CM 1930 DATA 8B,EC,83,EC,02,E8,1
        E,00,8B,4E
DE 1940 DATA 04,49,23,C8,89,4E,F
        E,E8,12,00
FM 1950 DATA 3C,7F,73,07,8B,46,F
        E,F7,D8,EB
BL 1960 DATA 03,8B,46,FE,8B,E5,5
        D,C3,B9,08
OG 1970 DATA 00,A1,2D,01,33,D2,A
        9,02,00,74
FO 1980 DATA 02,B2,01,A9,04,00,7
        4,02,B6,01
NK 1990 DATA 32,D6,D0,EA,D1,D8,E
        2,E8,A3,2D
KP 2000 DATA 01,A3,2F,01,C3,55,8
        B,EC,8B,46
EO 2010 DATA 04,CD,10,8B,E5,5D,C
        3,B4,0F,CD
JD 2020 DATA 10,B7,07,3C,03,7E,0
        6,3C,07,74
KJ 2030 DATA 02,32,FF,B8,00,06,3
        3,C9,BA,4F
II 2040 DATA 18,CD,10,B8,00,02,3
        3,D2,33,DB
EJ 2050 DATA CD,10,C3,FB,B4,00,C
        D,1A,89,16
MI 2060 DATA 2B,01,B9,08,00,A1,2
        B,01,33,D2
OB 2070 DATA A9,02,00,74,02,B2,0
        1,A9,04,00
CA 2080 DATA 74,02,B6,01,32,D6,D
        0,EA,D1,D8
FB 2090 DATA E2,E8,A3,2B,01,C3,5
        5,8B,EC,57
EN 2100 DATA 56,A1,03,01,B4,0C,8
        B,4E,06,8B
ED 2110 DATA 56,04,CD,10,5E,5F,8
        B,E5,5D,C3
OP 2120 DATA B4,07,CD,21,3C,51,7
        4,07,3C,71
BA 2130 DATA 74,03,33,C0,C3,B8,0
        1,00,C3
```


Atari XL/XE Super Editor

Rhett Anderson

This clever utility adds a variety of new features that improve the screen editor on Atari XL and XE computers. BASIC memory is unaffected.

Atari eight-bit computers are equipped with an exceptional screen editor. As you edit your BASIC programs, you can clear the screen, insert characters and lines, delete characters and lines, and move the cursor up, down, left, and right.

When Atari engineers developed the 1200XL computer, they used its new function keys to add even more editing features: one-key cursor control, cursor home, cursor to the bottom right corner, cursor to the left margin, cursor to the right margin, and a quick way to turn off the keyclick. Unfortunately, when the function keys were omitted in the newer Atari computers—the 600XL, 800XL, and the XE models—so were the new features. Fortunately, the code for these keys remains in the operating system. "Atari XL/XE Super Editor" resurrects this code, adding new editing power to the Atari 600XL, 800XL, 65XE, and 130XE.

Super Editor is written in machine language but is listed in the form of a BASIC loader. Since the DATA statements include machine language and key definitions which must be typed accurately, be sure to use the "Automatic Proofreader" found elsewhere in this issue to enter the program. Be sure to save a copy of the program before running it, since the BASIC part of the program is erased from memory after it is run.

Normal And Super Modes

After you have saved a copy of the program to tape or disk, type RUN. In a few seconds, you will see a message indicating that Super Editor is enabled. At this point, you have no new editing features. Super Editor has two modes of operation: normal and super. Whenever you run Super Editor, you'll be in the normal mode of operation, where all keystrokes act as they normally would. Change the mode with the HELP key, which acts as a toggle. Press it once to enter super mode, and again to return to normal mode. To help you tell the difference between the two modes, the screen color is changed to a darker blue in super mode. Here is a list of the new key assignments available in super mode:

CTRL-5	toggle keyclick
CTRL-6	cursor home
CTRL-7	cursor bottom left
CTRL-8	cursor to left margin
CTRL-9	cursor to right margin

In addition, you no longer need to press CTRL to cursor up, down, left, or right. To obtain the standard assignments of these keys (−, =, +, and *), press CTRL and the appropriate key.

None of these new assignments are available in normal mode, but normal mode does retain the keyclick value that you last used in super mode. If you turn the keyclick off in super mode, it remains off when you return to normal mode.

If you wish to disable Super Editor completely, press SYSTEM RESET. Type ?USR(1664) or DUMMY=USR(1664) in direct or program mode to reenable it.

Super Editor requires the top half of page 6 for its program and the bottom three quarters of page 1 (the stack) for the Super Editor keyboard table. These memory areas are not used by BASIC, but some utilities use them. A conflict could result in a locked-up computer, so be sure to test Super Editor with any other utilities that you may use before beginning any important work.

Atari XL/XE Super Editor

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
LA 1 REM COPYRIGHT 1987 COMP
     UTE! PUBLICATIONS, INC.
     {4 SPACES}ALL RIGHTS RE
     SERVED
EG 2 PRINT CHR$(125):PRINT "
     COPYRIGHT 1987"
AD 3 PRINT "COMPUTE! PUBLICA
     TIONS, INC."
AP 4 PRINT "ALL RIGHTS RESER
     VED"
BJ 5 FOR X=1 TO 1200:NEXT X
KF 10 FOR MEMORY=1664 TO 177
     4
GH 20 READ ML:POKE MEMORY,ML
     :CHECK=CHECK+ML:NEXT M
     EMORY
CF 30 IF 12623<>CHECK THEN P
     RINT "ERROR IN TYPING
     DATA STATEMENTS.":END
MC 40 DUMMY=USR(1664):? "
     {CLEAR}Atari XL/XE Sup
     er Editor Enabled.
     {5 SPACES}HELP key tog
     gles Super Editor.":NE
     W
PF 1010 DATA 104,162,0,189,8
     1
HO 1020 DATA 251,157,0,1,232
     ,224
CJ 1030 DATA 192,208,245,162
     ,12,189
JJ 1040 DATA 209,6,168,189,2
     22,6
EK 1050 DATA 153,0,1,202,16,
     243
JH 1060 DATA 162,6,160,166,1
     69,7
JD 1070 DATA 76,92,228,173,2
     20,2
IJ 1080 DATA 201,17,208,32,1
     69,0
IH 1090 DATA 141,220,2,173,2
     08,6
FB 1100 DATA 73,1,141,208,6,
     170
PI 1110 DATA 189,235,6,133,1
     21,189
IM 1120 DATA 237,6,133,122,1
     69,4
GH 1130 DATA 77,198,2,141,19
     8,2
NE 1140 DATA 76,98,228,1,6,7
OI 1150 DATA 14,15,134,135,1
     42,143
DC 1160 DATA 157,155,179,181
     ,176,30
CO 1170 DATA 31,28,29,43,42,
     45
CE 1180 DATA 61,137,142,143,
     144,145
ML 1190 DATA 0,81,1,251
```


Attention Programmers

COMPUTE! magazine is currently looking for quality articles on Commodore, Atari, Apple, and IBM computers (including the Commodore Amiga and Atari ST). If you have an interesting home application, educational program, programming utility, or game, submit it to COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Or write for a copy of our "Writer's Guidelines."

Resave

Amy Galtman

This convenient utility for the Commodore 64 lets you resave a program without fear of the infamous save-with-replace bug.

You have loaded a BASIC program from a disk and have decided that it needs a change or two. After editing, it must be saved to disk again, with a different, unique filename. One solution is to use a new name every time you save the program, perhaps using numbers to keep the versions straight. Under this scheme, you might have programs named SPRITE1, SPRITE2, and so on. This wastes disk space and also requires that you remember which number to use next. Later, you must scratch all the old versions and rename the most recent version with the final name.

Another option is to use a save-with-replace command which, unless you are very careful, may damage an entire disk. To avoid this danger completely, you can scratch the original program before saving a new copy. This may be time consuming, particularly if you are the careful type who likes to validate the disk or verify the saved program afterward.

"Resave" is a short, convenient utility designed to simplify the process of resaving programs. Type in the program and save it. To install Resave in memory, simply load the program and run it; then type NEW and press RETURN. Now load the program you wish to edit.

When you are ready to resave the program, type SYS 49152 and press RETURN to activate Resave. You will be prompted for a filename. Enter the name of the program to resave. Do not include a 0: before the filename, since this utility automatically adds the drive prefix to the name. Resave scratches the original copy of the program, validates the disk to recover unused disk space, saves the new edition of the program, and verifies the save.

Note that Resave will not work with machine language programs unless they are designed to load and run like BASIC programs. Since Resave occupies memory beginning at location 49152, it cannot be used with any other program or utility that uses the same space.

You might prefer to eliminate the disk validation from Resave, since validation can take a long time for a disk that contains many files. To make this alteration, change the value 86 in line 100 to 73.

Resave

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
FC 1 REM *COPYRIGHT 1987*
GF 2 REM *COMPUTE! PUBLICATION
     S, INC.**
SH 3 REM *ALL RIGHTS RESERVED*
CA 4 PRINT"{CLR}{12 SPACES}COP
     YRIGHT 1987":PRINT"
     {6 SPACES}COMPUTE! PUBLIC
     ATIONS, INC."
GS 5 PRINT"{9 SPACES}ALL RIGHT
     S RESERVED"
RP 6 FOR X=1TO1500:NEXT
QQ 10 FOR T=49152 TO 49360:REA
      D S:POKET,S:NEXT:PRINT"
      {CLR}{DOWN}SYS 49152 TO
      {SPACE}UTILIZE"
CM 20 DATA76,55,192,70,73,76
JE 30 DATA69,78,65,77,69,32
BD 40 DATA40,60,61,49,54,32
QG 50 DATA67,72,65,82,65,67
AX 60 DATA84,69,82,83,41,63
DJ 70 DATA32,0,32,32,32,32
DH 80 DATA32,32,32,32,32,32
GG 90 DATA32,32,32,32,32,32
PK 100 DATA32,32,32,0,86,48
JM 110 DATA0,169,3,160,192,32
GB 120 DATA30,171,162,0,32,207
HH 130 DATA255,201,13,240,7,15
     7
SD 140 DATA36,192,232,76,64,19
     2
FE 150 DATA232,134,2,169,1,162
MK 160 DATA8,160,15,32,186,255
JH 170 DATA169,83,141,33,192,1
     69
QF 180 DATA48,141,34,192,169,5
     8
BQ 190 DATA141,35,192,166,2,23
     2
XG 200 DATA232,138,133,2,162,3
     3
CK 210 DATA160,192,32,189,255,
     32
HD 220 DATA192,255,32,195,255,
     169
CB 230 DATA1,162,8,160,15,32
KB 240 DATA186,255,169,2,162,5
     2
CM 250 DATA160,192,32,189,255,
     32
HQ 260 DATA192,255,32,195,255,
     169
DE 270 DATA1,162,8,160,1,32
XX 280 DATA186,255,166,2,202,1
     38
RB 290 DATA162,34,160,192,32,1
     89
JQ 300 DATA255,169,43,166,45,1
     64
RH 310 DATA46,32,216,255,169,1
DM 320 DATA162,8,160,1,32,186
MQ 330 DATA255,166,2,202,138,1
     62
DS 340 DATA34,160,192,32,189,2
     55
HP 350 DATA169,1,32,213,255,16
     9
EM 360 DATA1,32,195,255,96,0
```

Full-Screen Editor For Applesoft

Alex Wong

This convenient, full-featured editor makes it much easier to edit Applesoft BASIC programs. The program runs only in DOS 3.3.

One of the least popular features of Applesoft BASIC is its primitive means of editing a program line. "Full-Screen Editor" for the Apple II series gives you the benefit of a full-screen editor within Applesoft BASIC. With this program, you can list the part of the program you wish to change, move the cursor to the desired spot, change the line as needed, and enter the new line in memory.

This program works on any Apple II-series computer using DOS 3.3. Enter the program with the "MLX" machine language entry program printed elsewhere in this issue. When you run MLX, you'll be asked for a starting address and an ending address for the data you'll be entering. Here are the values to use for Full-Screen Editor:

```
STARTING ADDRESS?  8F00
ENDING ADDRESS?    95FF
```

Be sure to save a copy of the program before trying to run it. You can start Full-Screen Editor with a BRUN command. For instance, if you saved the program with the name FSE, type BRUN FSE and press Return to load and activate the editing features. The program lowers HIMEM to protect itself.

Listing The Lines To Edit

Once the editor has been installed, you can load the BASIC program you wish to edit. To prepare for editing, type a slash (/) followed by a range of line numbers that indicate which lines you want to display on the screen. This range can have the following forms:

100	list line 100
100–	list lines from 100 to the end of the program, or until the Esc key is pressed
100–200	list lines 100 to 200
–100	list from the beginning of the program to line 100

The Full-Screen Editor displays BASIC lines somewhat differently from the way Applesoft BASIC displays them. All line numbers start in column 1 with the first statement of the line starting in column 5. Lines too long to fit across one screen line will start in column 2 of the next line. There is only one space between keywords, and all control characters are displayed in inverse mode. The bottom of the screen will show CURSOR MODE or EDIT MODE, depending on the mode you are currently in.

When you invoke the editor, it always clears the bottom of the screen with a listing of the lines you specified. While lines are being listed, the space bar can be used to pause and restart the listing. If you invoke the editor without specifying a range of line numbers, it simply clears the bottom of the screen.

Cursor Mode

Once you have listed the desired lines on the screen, the Full-Screen Editor goes into cursor mode. In this mode you can move the cursor anywhere on the screen or enter edit mode to change a line. Use the I, J, K, M, and the left and right arrow keys to position the cursor on the spot where you want to make a change. Once you have the cursor positioned where you want it, press Control-E (hold down Control and press E) to enter edit mode. The Esc key exits the editor and returns you to BASIC.

Edit Mode

Edit mode allows you to change any line displayed on the screen. Most editing commands are performed by pressing Control together with a letter key. Following is a

list of the editing commands.

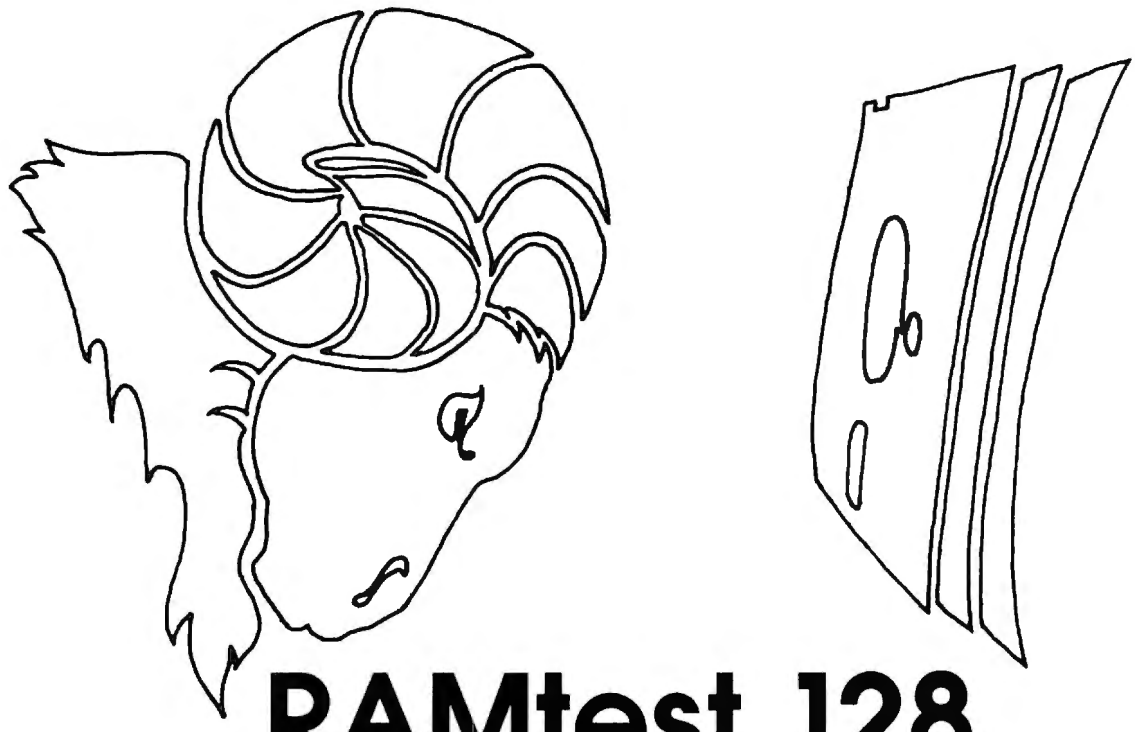
Control-J	Cursor down
Control-K	Cursor up
left arrow	Cursor left
right arrow	Cursor right
Control-I	Insert a blank space under the cursor and move following characters right
Control-D	Delete the character under the cursor and move following characters left
Return	Enter the line into memory
Esc	Exit edit mode, returning to cursor mode

You can press Return anywhere on a line to enter it into memory. The beginning of a line is marked by its line number, so be sure that all line numbers begin in the first column of the screen.

Full-Screen Editor

Please refer to the "MLX" article elsewhere in this issue before entering the following program.

```
8F00: 20 2F FB 20 AE 94 20 58 DC
8F08: FC A0 8F A9 16 20 48 95 B1
8F10: 20 3F 94 4C 8C 95 06 13 40
8F18: 05 20 16 31 2E 30 13 20 10
8F20: 20 01 17 05 20 31 39 38 33
8F28: 36 8D 00 20 1B FD C9 8D B9
8F30: D0 15 48 E6 76 F0 06 A9 A1
8F38: DD C5 33 D0 07 A9 AF CD 37
8F40: 00 02 F0 04 68 C6 76 60 E9
8F48: 68 9D 00 02 20 39 D5 86 3B
8F50: 76 86 B8 84 B9 A2 13 9D C7
8F58: E8 95 CA 10 FA 85 24 20 02
8F60: 3F 94 A5 00 8D F6 95 A5 12
8F68: 01 8D F7 95 A9 0F 8D E5 4E
8F70: 93 20 2F FB A2 17 86 23 A8
8F78: A9 11 85 25 20 42 FC 20 D7
8F80: AE 94 A2 01 20 34 95 20 9D
8F88: B4 94 20 B1 00 20 B1 00 2A
8F90: 20 6E 92 4E E8 95 A2 0F 84
8F98: 8E E5 93 A2 01 20 34 95 9B
8FA0: 20 0C FD C9 9B F0 2D C9 F3
8FA8: 85 F0 0D C9 E0 90 02 29 7B
8FB0: DF 20 0D 90 B0 E0 90 E8 85
8FB8: A2 02 20 34 95 20 0C FD 34
8FC0: C9 A0 90 06 20 ED FD 4C 60
8FC8: BD 8F C9 9B F0 C5 20 03 7F
8FD0: 90 4C B8 8F AD F6 95 85 55
8FD8: 00 AD F7 95 85 01 A2 18 49
8FE0: 86 23 CA 86 25 20 11 92 2C
8FE8: 20 42 FC A9 2B 85 38 A9 6C
8FF0: 8F 85 39 20 93 FE 20 EA 26
8FF8: 03 A9 FF 85 76 A2 EF 9A 15
9000: 4C D0 03 A2 95 A0 AB 84 11
9008: 9D A0 0C D0 08 A2 95 A0 45
9010: 9C 84 9D A0 0C 20 26 90 1C
9018: A2 95 A0 BA 84 9D A0 06 91
9020: 20 26 90 4C 6B 94 86 9E 0B
9028: D1 9D F0 06 88 88 88 10 9F
9030: F7 60 68 68 C8 B1 9D 85 C6
9038: 3C C8 B1 9D 85 3D 6C 3C EF
9040: 00 20 C1 93 A4 24 D0 07 39
9048: B1 28 20 44 92 90 3E C8 B0
9050: C4 21 F0 0B B1 28 88 91 BB
9058: 28 C8 C8 C4 21 D0 F5 88 E5
9060: A9 A0 91 28 E6 25 A5 25 6F
9068: C5 23 B0 1F A5 28 85 2A 40
9070: A5 29 85 2B 20 22 FC A0 36
9078: 00 B1 28 20 44 92 B0 0D E7
9080: C8 B1 28 A0 27 91 2A A0 F5
9088: 02 90 C9 C6 25 4C 9B 93 99
9090: 20 C1 93 A0 27 B1 28 8D 8C
9098: EE 95 88 B1 28 C8 91 28 72
90A0: 88 88 C4 24 10 F5 A4 24 C8
90A8: A9 A0 91 28 A0 00 84 24 AD
90B0: E6 25 A5 25 C5 23 D0 2B 1D
90B8: C6 25 AD EE 95 C9 A0 F0 31
90C0: 1F 20 E6 93 A5 25 48 20 02
90C8: 9B 93 A5 25 C5 22 F0 02 3E
90D0: C6 25 20 C1 93 68 20 C1 FE
90D8: FB A0 01 AD EE 95 91 28 34
90E0: 4C 9B 93 20 C1 FB A0 00 C2
90E8: B1 28 C9 A0 F0 0C AD EE 32
90F0: 95 C9 A0 F0 EB A6 25 20 D6
90F8: 00 94 A0 00 AD EE 95 91 39
9100: 28 4C 93 90 60 A6 25 8E 3C
9108: FB 95 20 4F 92 90 F5 8E D8
9110: EC 95 86 25 20 22 FC A2 58
9118: 00 8E F0 95 B1 28 C9 A0 B8
9120: B0 02 29 3F 29 7F 9D 00 B7
9128: 02 E8 C8 C4 21 90 ED EE 02
9130: F0 95 E6 25 A5 25 C5 23 D0
9138: D0 02 B0 18 20 22 FC A0 FF
9140: 00 B1 28 C9 A0 D0 0D AD 81
9148: F0 95 C9 06 F0 03 C8 D0 D8
9150: CB EE F0 95 2C EC 95 30 FC
9158: 3E C6 25 20 22 FC CA A0 2E
9160: 27 B1 28 C9 A0 D0 10 CA 58
9168: 88 10 F6 E8 CE F0 95 C6 6D
9170: 25 20 22 FC 4C 5F 91 8E D3
9178: F5 95 20 11 92 AD F0 95 D3
9180: C9 07 D0 14 84 24 A2 03 A2
9188: 20 34 95 20 0C FD C9 8D F6
9190: D0 05 4E E8 95 10 11 60 25
9198: E6 25 A5 25 C5 23 D0 08 E3
91A0: C6 25 8E F5 95 20 E6 93 30
91A8: AE F5 95 BD 00 02 C9 0D D7
91B0: F0 01 E8 A9 00 9D 00 02 BC
91B8: 20 BD 94 20 C1 93 A2 FF 91
91C0: A0 01 4C 44 D4 20 1B 92 32
91C8: 4E EA 95 A5 24 D0 10 20 7F
91D0: 16 92 A5 25 C5 22 F0 5C 9F
91D8: C6 25 20 22 FC 18 60 C6 9E
91E0: 24 4E EF 95 18 60 38 6E 22
91E8: EF 95 20 1B 92 E6 24 A5 3D
91F0: 24 C5 21 90 EC 20 11 92 61
91F8: 2C EC 95 30 E4 38 6E F8 01
9200: 95 E6 25 A5 25 C5 23 B0 DF
9208: 03 4C C1 FB C6 25 4C E6 04
9210: 93 A9 00 85 24 60 A9 27 DE
9218: 85 24 60 2C FB 95 10 14 42
9220: A4 24 B1 28 29 DF C9 C1 77
9228: 90 0A C9 DB B0 06 B1 28 38
9230: 49 20 91 28 60 20 84 94 D7
9238: 20 BD 94 20 B7 00 20 0C 7B
9240: DA 4C 99 94 C9 BA B0 05 02
9248: C9 B0 90 01 60 18 60 A5 6A
9250: 25 85 35 A0 00 8A 20 C1 46
9258: FB B1 28 20 44 92 B0 04 C0
9260: CA 10 F2 18 08 A5 35 20 30
9268: C1 FB 28 60 18 60 20 B7 B2
9270: 00 F0 F9 20 35 92 20 1A 61
9278: D6 BA 8E FA 95 20 B7 00 D5
9280: F0 14 C9 FF D0 08 A9 FF 56
9288: 85 50 85 51 D0 08 20 B1 E2
9290: 00 F0 F3 20 35 92 A0 01 A8
9298: B1 9B F0 78 C8 B1 9B AA 12
92A0: C8 B1 9B C5 51 D0 04 E4 21
92A8: 50 F0 02 B0 67 84 85 2C 02
92B0: F8 95 10 05 2C F9 95 30 AE
92B8: 5B 2C EB 95 30 03 20 52 8D
92C0: 94 A0 00 8C EB 95 84 24 04
92C8: 20 24 ED A9 A0 A6 24 E0 28
92D0: 05 F0 06 20 72 93 E8 D0 FB
92D8: F6 A4 85 C8 B1 9B 30 0E 4A
92E0: A9 A0 20 72 93 D0 03 20 34
92E8: 72 93 E6 85 A4 85 B1 9B 9B
92F0: D0 26 A8 A9 0D 20 72 93 19
92F8: AD EA 95 0D E9 95 30 08 41
9300: A9 A0 20 72 93 4C F8 92 A1
9308: B1 9B AA C8 B1 9B 86 9B 75
9310: 85 9C D0 82 4E EA 95 60 0D
9318: 10 CD AE 00 C0 E0 A0 D0 2C
9320: 0B 2C 10 C0 AE 00 C0 10 EC
9328: FB 2C 10 C0 E0 9B F0 52 10
9330: 38 E9 7F AA 84 85 A0 D0 D4
9338: 84 9D A0 CF 84 9E 20 4E 47
9340: 93 A4 85 C8 B1 9B F0 A6 1C
9348: 30 A4 A9 20 D0 99 A0 FF 16
9350: CA F0 07 20 2C D7 10 FB D8
9358: 30 F6 A5 9E C9 CF 90 05 A7
9360: A9 20 20 72 93 20 2C D7 DC
9368: 30 05 20 72 93 D0 F6 4C 2E
9370: 72 93 48 09 80 20 ED FD AD
9378: 2C F8 95 10 0D 2C F9 95 4A
9380: 10 08 AE FA 95 9A 2C 10 B6
9388: C0 60 68 60 48 A5 25 38 96
9390: 69 00 C5 23 D0 03 6E F9 C0
9398: 95 68 60 08 8E FF 02 BA 64
93A0: 8E FE 02 AE E5 93 9A 68 15
93A8: 85 25 68 85 24 68 8D F9 19
93B0: 95 BA 8E E5 93 AE FE 02 D8
93B8: 9A 20 22 FC AE FF 02 28 EA
93C0: 60 8E FF 02 BA 8E FE 02 EB
93C8: AE E5 93 9A AD F9 95 48 A5
93D0: A5 24 48 A5 25 48 BA 8E 85
93D8: E5 93 AE FE 02 9A AE FF 75
93E0: 02 4E F9 95 60 0F 48 98 9D
93E8: 48 2C EC 95 30 08 A0 00 19
93F0: 8C EA 95 8C F9 95 20 70 6B
93F8: FC 20 22 FC 68 A8 68 60 D1
9400: A5 22 48 86 22 A5 23 38 1C
9408: E9 01 48 20 24 FC A5 28 F9
9410: 85 2A A5 29 85 2B A4 21 11
9418: 88 68 C5 22 F0 0F E9 01 13
9420: 48 20 24 FC B1 28 91 2A 45
9428: 88 10 F9 30 E1 A5 22 20 E5
9430: C1 FB A0 27 A9 A0 91 28 DA
9438: 88 10 FB 68 85 22 60 A9 CE
9440: 8F A2 2B C5 74 90 06 D0 5E
9448: 08 E4 73 B0 04 85 74 86 CD
9450: 73 60 48 20 11 92 8D EA 2F
9458: 95 E6 25 A5 25 C5 23 D0 5C
9460: 05 C6 25 20 E6 93 20 22 4C
9468: FC 68 60 A0 20 8C FF 02 75
9470: AD 30 C0 CC FF 02 D0 03 0E
9478: AD 30 C0 88 D0 F5 CE FF 21
9480: 02 10 ED 60 A5 36 8D F1 85
9488: 95 A5 37 8D F2 95 A5 38 17
9490: 8D F3 95 A5 39 8D F4 95 0A
9498: 60 AD F1 95 85 36 AD F2 48
94A0: 95 85 37 AD F3 95 85 38 F0
94A8: AD F4 95 85 39 60 20 89 06
94B0: FE 4C 93 FE A9 C6 85 36 78
94B8: A9 94 85 37 60 A9 FE 85 2D
94C0: 36 A9 94 85 37 60 48 C9 EF
94C8: A0 B0 02 29 3F 48 84 35 9A
94D0: 2C EA 95 10 0B 20 52 94 90
94D8: A9 A0 A4 24 91 28 E6 24 F4
94E0: A4 24 68 91 28 E6 24 A5 56
94E8: 24 C5 21 D0 08 6E EA 95 2C
94F0: C6 24 20 8C 93 A5 F1 20 8A
94F8: A8 FC 68 A4 35 60 C9 A0 6C
9500: 90 FB C9 DD F0 14 20 4A EB
9508: FF AD ED 95 D0 03 20 78 01
9510: 95 20 3F FF EE ED 95 4C 9C
9518: F6 FD AD ED 95 F0 09 20 75
9520: 6B 94 20 0C FD 4E E8 95 7B
9528: 20 9B 93 20 B4 94 A2 EF EB
9530: 9A 4C 96 8F 20 3A 95 4C E8
9538: 9B 93 A9 C3 85 9D A9 94 12
9540: 85 9E 20 78 95 4C 4E 93 6F
9548: 85 9D 84 9E 20 58 FC A0 15
9550: 00 B1 9D F0 20 C9 81 90 66
9558: 11 C9 8D B0 0D 29 0F AA 11
9560: A9 A0 20 F6 FD CA D0 FA B3
9568: F0 03 20 F6 FD E6 9D D0 D7
9570: E0 E6 9E D0 DC 4C 0C FD D4
9578: 20 C1 93 A9 17 20 C1 FB E9
9580: 20 11 92 8D EA 95 8D ED E1
9588: 95 4C 9C FC A9 EB 8D F2 FF
9590: 03 A9 8F 8D F3 03 20 6F CD
9598: FB 4C EB 8F 95 EA 91 C9 90
95A0: D2 91 CA C8 91 CB ED 91 A8
95A8: CD FD 91 8D 05 91 FF 41 F4
95B0: 90 84 41 90 89 90 90 95 BB
95B8: E6 91 8A FD 91 8B D2 91 DE
95C0: 88 C5 91 43 55 52 53 4F F1
95C8: 52 20 4D 4F 44 C5 45 44 CB
95D0: 49 54 20 4D 4F 44 C5 56 FB
95D8: 45 52 49 46 59 3F 20 28 F8
95E0: 45 53 43 2F 47 45 54 A9 80
95E8: 00 00 00 00 00 38 46 39 BA
95F0: 35 46 30 39 35 46 30 35 3A
95F8: 46 38 46 38 38 33 41 00 B2
```



RAMtest 128

Jim Butterfield, Associate Editor

Have you ever suspected that your Commodore 128 might have a bad memory chip? This program can provide an accurate answer.

The 128 has a lot of RAM (Random Access Memory). Just one bad byte could make your programs misbehave. How can you be sure that your 128's memory is OK? "RAMtest 128" performs a thorough memory test.

Programmers often suspect hardware problems when their newest programs fail to behave correctly. They are usually wrong; it's almost always a programming error, not a problem with the machine. But on rare occasions memory does fail, and it's useful to have a program that can confirm or deny such a suspected problem.

Using The Program

RAMtest 128 is written in machine language (ML). The program can be entered using the "MLX" machine language entry utility published regularly in COMPUTE!. If you don't have a copy of 128 MLX (found in the April issue), you can use the 64 version (in 64 mode) of MLX, found elsewhere in this issue. MLX will request a starting address and an ending address for the data you'll be entering. For RAMtest 128, use these values:

Starting address: 1C01
Ending address: 1CD1

Be sure you save a copy of the RAMtest 128 data before leaving MLX.

You can also bypass MLX and use the 128's built-in ML monitor to enter the data. (If you're using the 80-column display, you should first switch to the 40-column screen.) To enter the monitor, type MONITOR and press RETURN, or simply press the F8 function key. You must remember several things when entering the data using the monitor: Precede each line with a greater-than (>) symbol, substitute a space for the colon (:) following the four-digit address value, and omit the final (rightmost) two-digit value in each line. Each line you enter should consist of a > character, a four-digit address, and eight two-digit data values. After you have entered and checked all of the lines, save the program with a command of the form:

```
S "filename",08,1C01,1CD0
```

(For tape, substitute 01 for 08 in the command above.) After saving the program, type X and press RETURN to exit the monitor and return to BASIC.

When you're ready to conduct the test, load the RAMtest 128 program as you would any BASIC program. Use RUN to start the test. Every 20 seconds the program reports OK and does the test again. If you're using the 80-column display, type FAST before you run the program to double its speed. Press the reset button (near the power switch).to stop the program.

About Memory Tests

It seems simple to devise a memory test. Just store something in every memory cell you can conveniently reach, and then read it back to insure that the data was stored correctly. But that's not good enough. A good test should check for correct

addressing, as well.

Let's outline the kind of problem that simple tests won't catch. Whenever you do something that requires access to a memory address, the computer's processor signals the address it wants to use by sending it over the *address bus*—a group of 16 wires that together define the address. Now suppose there's something wrong with the address bus or its connections, so that a memory chip doesn't read all the wires correctly. The chip still sees an address, but it's the wrong one. If this happens, the data is stored in the wrong address. Worse, when the processor reads the data back to check it, it reads from the same wrong address. The data seems to be right, but it's not.

Here's a specific example. Suppose that you try to store data to address 5000, but an error on the address bus causes the address to be seen as 4992. (Hardware hackers will spot this as an error in line 3 of the address bus.) The processor stores something there, thinking that the data is going to 5000. But it really goes to 4992. Now the processor tries to test memory by recalling the contents of 5000. It sends out the address again. The same error causes the address to be seen as 4992, and the contents of 4992 are delivered to the processor. The processor sees that it's the same value that was stored and does not report a memory error.

A Better Algorithm

Several good methods for testing memory have been developed. RAMtest 128 uses a shortened version of a method published by Knaizuk and Hartmann in 1977. In this scheme, individual memory cells are tested in the usual way, by storing a value and reading it back. A special wrinkle is used to detect addressing faults.

A single fault in the addressing system causes an address error that is a power of 2. In our example above, the error was 8, which is 2 raised to the third power. Depending on the address line that is at fault, the error might be 2, 4, 8, 16, 32, 64, 128, or other power of 2 up to 32768. Notice that none of these numbers may be divided evenly by 3.

To detect addressing faults, we store to memory in a pattern of three bytes: the first byte with one value, and the next two with a different value. Any power-of-2 error will disrupt the three-byte pattern. When the program sweeps through memory checking for correct values, it will find a misplaced value and know there's a problem.

Inside The Test

The testing portion of this program stores a value (hex $55, which has the binary pattern %01010101) in every location it can reach in both bank 0 ($0B00–$FEFF) and bank 1 ($0400–$FEFF). Then it goes back and puts a different value (hex $AA, which has the binary pattern %10101010) into every third location. It's important, by the way, that this be done as a separate program step. Finally, the program sweeps through both banks, checking that all locations contain the expected values.

After the first sweep, the program reverses the test values and does it all over again. If everything checks out, RAMtest 128 prints OK on the screen and repeats the test. If an error occurs, the program transfers control to the 128's built-in machine language monitor.

Why does the test repeat? Some memory failures are intermittent; they might be brought on by heat, power supply variations, or even aging components. If that's your suspicion, let it run; you'll see a lot of OK messages printed.

If the program stops with an error, you'll end up in the machine language monitor. If you're familiar with machine language, the contents of memory locations $FA–$FC will give you a hint as to the location of the error. If you're not, type M FA FA and copy down the computer's response in order to show it to someone who understands the hardware side.

Program Setup

This program is constructed to take into account some oddities of the 128's configuration. If you have executed a GRAPHIC command in the current session, BASIC might start at an address that's higher than normal. Thus, the program begins with the command BANK 0. Without this command, we might SYS to ROM addresses instead of to our program in RAM. The SYS command doesn't go to an absolute address, but to an offset relative to the beginning of BASIC. Before it begins the test, the program relocates the testing section to the memory area at location $1000. The test cleans away the relocating code as well as the original copy of the test code.

Scope

The program tests over 123K of memory (to be exact, 126,720 bytes) in 20 seconds—even less if you're in FAST mode. That's over 6000 bytes per second, which is very fast indeed. Each byte is written at least twice and then checked twice in a mere 160 microseconds or so.

A few areas are not tested. The program doesn't go below address 2816 ($0B00) in bank 0, since that would garble the operating system information and the screen display. And it doesn't go above 65279 ($FEFF) to avoid disturbing the MMU (Memory Management Unit) registers and the sensitive interrupt routines which reside there.

It will be extremely rare to find memory problems; there are many more bugs in software than in hardware. If you think you've found a problem, check the program on a friend's computer before you take your machine in for service.

RAMtest 128

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
1C01:1D 1C 64 00 FE 02 30 3A F5
1C09:9E C2 28 34 35 29 AA C2 EF
1C11:28 34 36 29 AC 32 35 36 92
1C19:AA 33 30 00 00 00 A0 2E E8
1C21:B1 2D 99 D2 00 C8 C0 CE 51
1C29:D0 F6 78 4C 00 01 20 6F 0F
1C31:01 AD 9E 01 91 FA 20 7F 71
1C39:01 90 F9 20 6F 01 AD 9F D1
1C41:01 91 FA 20 7F 01 B0 0A 2B
1C49:20 7F 01 B0 05 20 7F 01 45
1C51:90 EF 20 6F 01 B1 FA CD 5B
1C59:9F 01 D0 2C 20 7F 01 B0 30
1C61:18 B1 FA CD 9E 01 D0 20 09
1C69:20 7F 01 B0 0C B1 FA CD A7
1C71:9E 01 D0 14 20 7F 01 90 26
1C79:DC AD 9E 01 AE 9F 01 8D F2
1C81:9F 01 8E 9E 01 10 A7 A9 C6
1C89:00 8D 00 FF A9 4F 20 D2 C2
1C91:FF A9 4B 20 D2 FF A9 20 A9
1C99:20 D2 FF D0 91 A0 3F 8C BD
1CA1:00 FF A0 0B 84 FB A0 00 F3
1CA9:84 FA 84 FC 60 18 C8 D0 09
1CB1:FB E6 FB A6 FB E0 FF D0 BF
1CB9:F3 A2 04 86 FB AE 00 FF 18
1CC1:E0 7F B0 E8 E6 FC A2 7F DE
1CC9:8E 00 FF 60 AA 55 00 00 F9
```

ML Runner

Paul Lindner

This Commodore 64 utility makes machine language programs easier to handle by turning them into files that load and run like BASIC. No machine language expertise is needed to use the program.

Some of the best programs available for the Commodore 64 are written in machine language—the computer's native tongue. But machine language files aren't as easy to use as BASIC programs. Most require special loading procedures, and it's easy to forget the SYS address that starts up a program. But it doesn't have to be that way. COMPUTE!'s *SpeedScript* word processor, for instance, loads and runs exactly like BASIC, even though it's written entirely in machine language. Wouldn't it be convenient if every machine language program worked that way?

"ML Runner" does exactly that, converting a machine language program into a program file that saves, loads, and runs as if it were a BASIC program. Type in the program and save a copy on disk or tape. Then select the program you wish to convert. In order to do this, you must know four facts about the machine language program: its name, starting address, ending address, and the SYS address normally used to run the program.

After you run the program, respond to the prompts as indicated. Be sure to use a different name for the new file you are about to create. When typing the starting and ending addresses, you can use either decimal or hexadecimal numbers. Hexadecimal numbers must be preceded with a dollar sign ($). These addresses are the same ones printed in the article for any COMPUTE! program which is typed in with the "MLX" machine language entry program.

After you answer all the questions, the program goes to work, displaying the number of bytes it has already processed. When it is done, the file is ready to use. Load it and type RUN, as you would any BASIC program. The converted program is also much easier to copy than the original. To make a new copy of the program, simply load it into memory and then save a new copy as if this were a normal BASIC program.

This method can be used even if you don't know all the required address information. This short program will display the starting address of any program on disk:

```
QG 10 OPEN 2,8,2,"0:FILENAME,P
      ,R"
PA 20 GET#2,A$,B$:CLOSE 2
BH 30 PRINT ASC(A$+CHR$(0))+25
      6*ASC(B$+CHR$(0))
```

Finding the ending address of a program is even easier. Enter these two commands in direct mode (without line numbers):

```
LOAD"FILENAME",8,1
PRINT PEEK(46)*256+PEEK(45)
```

In both of these examples, you should replace FILENAME with the name of the program you wish to investigate. The SYS address of a machine language program is usually the same as the starting address.

There are a few programs for which this technique is unsuitable. Obviously, if a program already loads and runs like BASIC, there is no point in converting it with this program. Some programs, including much commercial software, take over the computer when you load them with ,8,1 (they start without requiring that you enter a SYS command). Programs of this type rely on special features of the computer and may not work at all if you convert them with this program.

ML Runner

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
RH 10 REM *COPYRIGHT 1987*
FQ 20 REM *COMPUTE! PUBLICATIO
      NS, INC.*
KP 30 REM *ALL RIGHTS RESERVED
      *
QK 40 DEF FN HI(Q)=INT(Q/256)
GE 50 DEF FN LO(Q)=Q-256*FN HI
      (Q)
FJ 60 PRINT"{CLR}COPYRIGHT 198
      7 COMPUTE! PUBLICATIONS"
BD 70 PRINT"{8 SPACES}ALL RIGH
      TS RESERVED"
KX 80 FOR X=1TO1500:NEXT
```

```
HA 90 PRINT"{CLR}{2 DOWN}":PRI
      NTTAB(15)"ML RUNNER"
SK 100 INPUT "{3 DOWN}NAME OF
       {SPACE}PROGRAM";NA$
ME 110 INPUT"STARTING ADDRESS
       {SPACE}OF PROGRAM";A$:G
       OSUB390:SA=A
CH 120 INPUT "ENDING ADDRESS'O
       F PROGRAM";A$:GOSUB390:
       EA=A
PK 130 EE=EA-SA+2093:EH=FN HI(
       EE):EL=FN LO(EE)
DD 140 INPUT "SYS ADDRESS";A$:
       GOSUB390:SY=A
CC 150 INPUT "OBJECT FILE NAME
       ";N$:PRINT:PRINT
CR 160 PRINT"{CLR}{HOME}
       {3 DOWN}BYTES TO BE CON
       VERTED.."
ME 170 OPEN 15,8,15
PE 180 OPEN 1,8,3,"0:"+N$+",P,
       W"
GB 190 OPEN 2,8,4,"0:"+NA$+",P
       ,R"
JX 200 GOSUB280
MQ 210 GET#2,A$,A$
PM 220 FOR X=SA TO EA
KH 230 GET#2,A$:AA=ASC(A$+CHR$
       (0))
DR 240 PRINT#1,CHR$(AA);
FJ 250 PRINT"{UP}"TAB(24)EA-X"
       {LEFT}{3 SPACES}":NEXT
CK 260 CLOSE 1:CLOSE 2:CLOSE 1
       5:END
HG 270 REM PUT FUSE AND ML TOG
       ETHER{7 SPACES}*
FD 280 READ A:IFA<0 THEN ON AB
       S(A) GOSUB320,330,340,3
       50,360,370
FA 290 PRINT#1,CHR$(A);
KH 300 IF E=0 THEN280
PK 310 RETURN
HM 320 A=EL:RETURN
HP 330 A=EH:RETURN
GX 340 A=FN LO(EA+1):RETURN
QC 350 A=FN HI(EA+1):RETURN
QB 360 A=FN LO(SY):RETURN
ER 370 A=FN HI(SY):E=1:RETURN
EK 380 :
DK 390 IF LEFT$(A$,1)="$"ANDLE
       N(A$)=5THEN A$=RIGHT$(A
       $,4):GOSUB410:RETURN
SF 400 A=VAL(A$):RETURN
HB 410 A=0: P=4096:FORX=1TO4:L
       $=MID$(A$,X,1):GOSUB450
       :A=A+P*N
XR 420 P=P/16
XC 430 NEXTX:RETURN
XQ 440 :
XH 450 N=0
KC 460 N=-15*(L$="F")-14*(L$="
       E")-13*(L$="D")-12*(L$=
       "C")-11*(L$="B")-10*(L$
       ="A")
FQ 470 IF N=0 THEN N=VAL(L$)
BE 480 RETURN
QA 490 :
SQ 500 REM STARTING ADDRESS
DH 510 DATA 1,8
SG 520 REM BASIC FUSE
KX 530 DATA 12,8,10,0,158,32,5
       0,48,54,50,0,0,0
MB 540 REM ML PART
HX 550 DATA 169,44,133,95,169,
        8,133,96
JX 560 DATA 169,-1,133,90,169,
       -2,133,91
SB 570 DATA 169,-3,133,88,169,
       -4,133,89
HR 580 DATA 32,191,163,76,-5,-
       6
```


Car Payments

Brian Flynn

Planning on borrowing money to buy a car? If so, you'll find this short program for the IBM PC/PCjr and compatibles helpful in estimating your payments. Excerpted from Easy BASIC Programs for the IBM PC and PCjr *by Brian Flynn (COMPUTE! Books). Cartridge BASIC is required on the PCjr.*

After many months, you've finally succumbed to your dream of owning that new car. You haggle with the dealer and finally get the price within reason. But can you afford the payments?

It's easy to have that information *before* you walk into the dealership, or into your bank or credit union. All you have to do is run this short program on your IBM PC or PCjr or compatible. By inserting the amount of your loan, its length, and its interest rate, you can quickly see how much your monthly payments will be, as well as the total amount of interest you'll pay over the life of the loan.

Loan Officer

You can have this information at your fingertips, just as your loan officer does, by typing in and saving "Car Payments." Run it, and you'll cycle through a number of screen displays, each of which asks for a different piece of data.

Amount borrowed. You can enter any amount up to $999,999. Enter whole numbers, not fractions or decimals. The program automatically places dollar signs, commas, and decimal points.

Length of loan. Type in the length of the loan, in years and months. Three years, for instance, can be entered as 3,0 (3 years, 0 months) or as 2,12.

Interest rate. Enter the interest rate you expect to pay for your car loan. You'll probably have to call your bank or credit union, or the dealership, to find out some possible interest rates. You can enter decimals in this category. Twelve and a half percent interest would be typed in as 12.5.

The program takes only a moment to compute your loan summary. It summarizes your entries and then tells you the amount of your monthly payment, the total you'll pay, the principal (which should be identical to the amount borrowed), and the total interest paid.

Pressing any key stops the program. If you want to go through it again, to see the results of a different interest rate, for example, type RUN and the program starts over.

Buy A New Car

How about an example?

We'll borrow $5,000 for three years. Assume our interest rate will be 12.5 percent. After entering those numbers, the computer pauses a moment and then displays the loan summary.

The monthly payments will be $167.27. The total amount you'll pay is $6,021.65, with $1,021.65 being interest on the loan. What would be the monthly payments if you stretched out the loan to four years? Or if you found a loan for 12 percent instead? All you have to do is run the program again, using the new data. It's that easy.

Two-Toned

Assuming you have the hardware to display this program in color, it will take only one line to create a two-toned display. Create a new line 90 in this format:

90 COLOR *f,b*

where *f* is the foreground color and *b* is the background color. For instance, COLOR 14,1 will create a blue background with yellow text.

To enliven the screen display even more, you could place COLOR statements before each LOCATE statement in the routine that starts at line 810. Put COLOR 4,1 at the beginning of line 830, for example, and the text starting with LOAN VALUES changes to red on blue. Experiment until you have combinations you like, then save the modified version of the program to disk.

If you have a PCjr and you're using a color TV or a non-RGB monitor for your display, you'll have to make another change. In line 190, change the SCREEN 0 statement to SCREEN 0,1. Include the colon. If you don't make this change, you'll see the screen in shades of gray, not color.

Car Payments

For instructions on entering this program, please refer to "COMPUTE!'s Guide To Typing In Programs" elsewhere in this issue.

```
PH 10 'Copyright 1987 Compute! P
      ublications, Inc.
FA 20 'All Rights Reserved
EC 30 CLS:PRINT TAB(20)"Copyrigh
      t 1987 Compute! Publicatio
      ns, Inc.":PRINT TAB(30)"Al
      l Rights Reserved"
KO 40 FOR X=1 TO 900:NEXT X
FN 100 REM CAR PAYMENTS
GD 110 REM ENTER DATA
GF 120  GOSUB 180
HD 130 REM COMPUTE
GE 140  GOSUB 630
DA 150 REM DISPLAY RESULTS
JO 160  GOSUB 750
MI 170 END
MB 180 REM ENTER DATA
JM 190  KEY OFF: SCREEN 0: WIDTH
       80: LOCATE ,,0
PJ 200  DEFINT M,N,Y: DEFDBL L,P
       ,R,T
IC 210  H$ = STRING$(80,CHR$(205
       ))
IE 220  CLS
IC 230  PRINT H$;
CJ 240  PRINT TAB(34)"CAR PAYMEN
       TS
FM 250  PRINT H$
EB 260  REM AMOUNT OF LOAN
EN 270   GOSUB 330
DJ 280  REM PERIOD OF LOAN
DC 290   GOSUB 410
MB 300  REM INTEREST RATE
BK 310   GOSUB 540
MB 320 RETURN
DB 330 REM LOAN
DB 340  LOCATE 5,42: PRINT SPC(3
       0): BEEP
DK 350  LOCATE 5,1: INPUT "How m
       uch money would you like
       to borrow ";L$
OE 360  LOAN = VAL(L$)
ML 370  LOCATE 23,27: PRINT SPC(
       30)
CA 380  IF LOAN <= 0 THEN LOCATE
        23,28:PRINT "Please borr
       ow something !": GOTO 340
HD 390  IF LOAN > 999999! THEN L
       OCATE 23,27:PRINT "Please
        scale down figure !": GO
       TO 340
MD 400 RETURN
LH 410 REM PERIOD OF LOAN
IG 420  CLS
EP 430  PRINT "Please enter the
       length of your loan in ye
       ars and months.
OL 440  LOCATE 3,11: PRINT SPC(3
       0): BEEP
MM 450  LOCATE 3,2: INPUT "Years
        = ";Y$
OE 460  Y = INT(VAL(Y$))
FI 470  IF Y < 0 THEN 440
QN 480  LOCATE 5,11: PRINT SPC(3
       0): BEEP
CA 490  LOCATE 5,1: INPUT "Month
       s = ";M$
MJ 500  M = INT(VAL(M$))
NK 510  IF M < 0 OR M > 12 THEN
       480
IF 520  N = Y*12 + M
MF 530 RETURN
MC 540 REM INTEREST RATE
JN 550  CLS
OP 560  PRINT "Please enter the
       interest rate on your loa
       n.  For example, enter 8
       for 8%,
KE 570  PRINT "11 for 11%, and s
       o on.
HO 580  LOCATE 5,18: PRINT SPC(3
       0): BEEP
KA 590  LOCATE 5,1: INPUT "Inter
       est Rate = ";R$
FL 600  RATE = VAL(R$)
KG 610  IF RATE <= 0 THEN LOCATE
        23,28:PRINT "There's no
       free lunch !":GOTO 580
ME 620 RETURN
MD 630 REM COMPUTE
FK 640  REM INTEREST RATE PER PE
       RIOD
BH 650   R = (RATE/100)/12
ED 660  REM PAYMENT PER PERIOD
JK 670   REM NUMERATOR
ED 680    P1 = LOAN*R$(1+R)^N
OI 690   REM DENOMINATOR
GK 700    P2 = (1+R)^N - 1
EI 710    PPP = P1/P2
GG 720   REM TOTAL PAYMENT
AG 730    TPAYMENT = N*PPP
NJ 740 RETURN
DE 750 REM DISPLAY
JE 760  CLS
PK 770  F1$ = "=$$#,##########.##
OY 780  F2$ = "=   #,##########
GK 790  F3$ = "=    ##########.##
       %
IC 800  PRINT H$;
LN 810  PRINT TAB(30)"SUMMARY OF
       THE LOAN
FM 820  PRINT H$
CB 830  LOCATE 5,15:PRINT "LOAN
       VALUES:
BK 840  LOCATE 7,18:PRINT "Amoun
       t";TAB(40) USING F1$;LOAN
KH 850  LOCATE 8,18:PRINT "Numbe
       r of years";TAB(40) USING
        F2$;Y
PF 860  LOCATE 9,18:PRINT "Numbe
       r of months";TAB(40) USIN
       G F2$;M
LN 870  LOCATE 10,18:PRINT "Inte
       rest rate";TAB(40) USING
       F3$;RATE
NH 880  LOCATE 14,15:PRINT "LOAN
        PAYMENTS:
AG 890  LOCATE 16,18:PRINT "Mont
       hly";TAB(40) USING F1$;PP
       P
BE 900  LOCATE 17,18:PRINT "Tota
       l";TAB(40) USING F1$;TPAY
       MENT
MJ 910  LOCATE 18,18:PRINT "Prin
       cipal";TAB(40) USING F1$;
       LOAN
HP 920  LOCATE 19,18:PRINT "Inte
       rest";TAB(40) USING F1$;T
       PAYMENT - LOAN
FA 930  LOCATE 22,1:PRINT H$
OE 940  LOCATE 23,34: PRINT "Pre
       ss any key
IN 950  S$ = INKEY$
KC 960  IF S$ = "" THEN 950
NB 970 RETURN
```

College Planner

Brian Flynn

College costs are spiraling ever upward. Tuition with room and board at many state-supported schools often runs a couple of thousand dollars a year, and double or triple that at private schools. With costs like these, a long-range savings plan is certainly useful. That's where "College Planner," written for the IBM PC/PCjr and compatibles, can help. Excerpted from Easy BASIC Programs for the IBM PC and PCjr *by Brian Flynn (COMPUTE! Books). Cartridge BASIC is required for the PCjr.*

If you have children, you probably assume they'll go on to some form of higher education. Perhaps they'll take classes at a community college. Or maybe at the local technical school. Perhaps even at a four-year university. All cost money. How much are you going to have to save for those years?

"College Planner" gives you an idea of what it will cost to educate a child beyond high school. As with "Car Payments," found elsewhere in this issue, you can run the program as many times as you want, changing the parameters to reflect different situations. This will give you a better idea of what it *might* cost to send your child to college.

The Paper Chase

After typing in and saving a copy of College Planner, run it. You have to select the appropriate menu if you want to change the default parameters (the values that are set when the program first runs). It's not difficult.

The easiest way to show how College Planner works is to go through an example.

Let's assume your child is now 12 years old. The first display screen asks for the number of years until the paper chase begins. Respond with 6 (18 − 12 = 6).

The computer then displays the main menu. Here, you decide if you want to review savings, expenses, or economic assumptions; to compute totals; or to exit the program. Pressing a single key selects another menu.

Review savings. By pressing the 1 key, you'll see this menu on the screen:

```
Annual Savings

A. Parents  = $1,000.00
B. Kids     =    500.00

Change Value (Y/N)?
```

Notice the default settings of $1,000 for parents' and $500 for child's savings. Just press the Y key to change either of these. You'll be asked for the item to change (A or B), and then for the new amount. Let's change them to $500 for the parents' contribution and to $250 for the child's contribution. After you've entered the figures, press the N key; you'll return to the main menu.

Review expenses. Now press the 2 key. You should see something similar to this:

```
First-Year Expenses (Today's Prices)

A. Tuition        = $4,000.00
B. Room & Board   =  3,500.00
C. Books          =    300.00
D. Travel         =    150.00
E. Laundry        =     75.00
F. Entertainment  =    250.00

Change Value (Y/N)?
```

Again, to change the amount in each category, press the Y key, select the item, and enter the new figure. For example, let's say that your child is going to a school several states away, with higher travel expenses to and from school. Change that category to $500. Press the N key to return to the main menu.

Review economic assumptions. Press the 3 key and you'll see the display change:

```
Economic Assumptions

A. Expected interest rate  = 10.00%
B. Expected inflation rate =  5.00%

Change Value (Y/N)?
```

The default values of 10 and 5 percent can be changed. The interest rate is what your yearly savings will be compounded by, while the inflation rate signifies how much costs increase each year. College Planner assumes that college costs increase at the same rate as inflation. This might not be true in a single year, but over a longer period of time it's accurate enough for this forecaster. Let's change the interest rate to 12 percent. (You were able to make better-than-average investments.) Hit the N key to return to the main menu.

Compute totals. Hit the 4 key and the program will figure out what it's going to cost you to send your child to college. The first screen displays the cost of a college education (based on what you entered in the Review Expenses menu) in terms of *today's dollars,* discounting inflation. All four years should have the same total. Press any key and another display appears. This shows what the cost of the education will be when your child reaches college age. The numbers are adjusted for inflation, and in fact increase each year by 5 percent, or by the rate of inflation you earlier specified. Note that these figures are in *future dollars.* Hit any key and the Bottom Line screen displays. It shows how much you saved, the interest you received over those six years, the total money available, the cost of college in today's dollars, the total in inflated dollars, and the final balance. Hitting any key will return you to the main menu again.

Now you can run the program again, this time entering a higher amount of savings. By trial and error, you'll find how much you'll have to save in order to pay for your child's education. Of course, when you use this program yourself, you'll be entering different parameters to reflect your child's age and your own estimates of what college will cost.

School Colors

Adding color is relatively simple. If you want to change the display, insert a new line 90, in the format

90 COLOR *f,b*

where *f* is the foreground color value and *b* is the background color value. A line 90 that includes COLOR 15,4, for example, will show the text in high-intensity white on a red background. If you have a PCjr and you're using a color TV or a non-RGB monitor for your display, you'll have to make one additional change. In line 290, change the SCREEN 0 statement to SCREEN 0,1. Make sure you include the colon that follows. If you don't make this change, you'll see the screen in shades of gray, not color.

College Planner

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
FI 10 ' COPYRIGHT 1987 COMPUTE!
      PUBLICATIONS, INC.
PK 20 ' ALL RIGHTS RESERVED
KL 30 REM COLLEGE PAYMENTS
CM 40 REM INITIALIZE
CA 50  GOSUB 130
NN 60 REM ENTER VALUES
HD 70  GOSUB 470
II 80 REM CHOOSE FROM MAIN MENU
GL 90  GOSUB 630
HN 100  ON PICK GOSUB 760,760,76
       0,1170
GC 110  IF PICK <> 5 THEN 90
LO 120 END
MF 130 REM INITIALIZE
KN 140  REM TITLE
BP 150   GOSUB 210
PL 160  REM HEADING
GH 170   GOSUB 270
EE 180  REM KEY VALUES
EA 190   GOSUB 330
MM 200 RETURN
HC 210 REM TITLE
OB 220  KEY OFF: SCREEN 0: WIDTH
       80: LOCATE ,,0: CLS
DD 230 PRINT TAB(20)"Copyright 1
       987 Compute! Publications
       , Inc.":PRINT TAB(30)"All
        Rights Reserved"
KA 240  LOCATE 13,32: PRINT "Col
       lege Planner
CD 250  FOR DELAY=1 TO 2500: NEX
       T
NI 260 RETURN
GK 270 REM HEADING
JA 280  CLS
IF 290  PRINT "College Planner i
       s designed to help you de
       velop a plan to pay for y
       our child's
NA 300  PRINT "education.  Colle
       ge Planner assumes that y
       our annual savings will e
       arn interest
BI 310  PRINT "but that inflatio
       n will force expenses upw
       ard.
MB 320 RETURN
MP 330 REM KEY VALUES
EN 340  DEFINT I-Q,T,Z: DEFDBL C
       ,E,R,S,V,X
MB 350  REM NUMBER OF CATEGORIES
LC 360   K = 3
KH 370  REM MAX NUMBER OF ITEMS
       PER CATEGORY
JB 380   DATA 10
PM 390   READ NX
KK 400   DIM C$(3), EK(NX+1,4),
       EF(NX+1,4), X$(NX,3), X(N
       X,3)
DM 410  REM ACTUAL NUMBER OF ITE
       MS
AG 420   DATA 2,6,2
BM 430   FOR I=1 TO K
OM 440    READ N(I)
MI 450   NEXT
MK 460 RETURN
QL 470 REM ENTER VALUES
QM 480  REM YEARS TO COLLEGE
CJ 490   LOCATE 5,39: PRINT SPC(
       30): BEEP
IF 500   LOCATE 5,1: INPUT"How m
       any years until college b
       egins ";S$
DD 510   N = VAL(S$)
BE 520   IF N < 1 THEN LOCATE 23
       ,23: PRINT "It's a bit to
       o late for planning !": G
       OTO 490
QL 530  REM CATEGORIES
BP 540   FOR I=1 TO K
OC 550    READ C$(I)
ML 560   NEXT
NP 570  REM ITEMS
BH 580   FOR I=1 TO K
NN 590    FOR J=1 TO N(I)
AH 600     READ X$(J,I), X(J,I)
ME 610    NEXT J,I
ME 620 RETURN
HP 630 REM MAIN MENU
JM 640  CLS
PN 650  LOCATE 10,31: PRINT "Wou
       ld you like to
KH 660  PRINT : PRINT TAB(32)"1.
       Review savings
IF 670  PRINT TAB(32)"2. Review
       expenses
KK 680  PRINT TAB(32)"3. Review
       economic assumptions
DB 690  PRINT TAB(32)"4. Compute
       totals
FE 700  PRINT TAB(32)"5. Exit
ED 710  PRINT: PRINT TAB(31)"Cho
       ice = ?_": BEEP
HF 720  S$ = INKEY$
CF 730  PICK = INT(VAL(S$))
DA 740  IF PICK < 1 OR PICK > 5
       THEN 720
NL 750 RETURN
CF 760 REM UPDATE
JD 770  H$ = STRING$(80,CHR$(205
       ))
FF 780  P = PICK
PG 790  Z = N(P)
QF 800  REM DISPLAY
JE 810   GOSUB 850
IF 820  REM SELECT
MB 830   GOSUB 970
MK 840 RETURN
DF 850 REM DISPLAY
LF 860  IF PICK <> 3 THEN F$ = "
       = $$#,####.##" ELSE F$ =
       "= ###.## %
JE 870  CLS
JC 880  PRINT H$;
AD 890  LN = LEN(C$(P))
EA 900  PRINT TAB(40-LN/2)C$(P)
FL 910  PRINT H$
HM 920  FOR J=1 TO Z
MB 930   PRINT TAB(13)CHR$(J+64)
       ;".";TAB(16)X$(J,P);TAB(4
       5) USING F$;X(J,P)
DK 940  NEXT J
OI 950  LOCATE 16,1: PRINT H$
MP 960 RETURN
CG 970 REM SELECT
BA 980  LOCATE 18,1: PRINT "Chan
       ge value (Y/N) ?": BEEP
IF 990  S$ = INKEY$
DM 1000  IF S$ = "N" OR S$ = "n"
        THEN 1160
II 1010  IF S$ <> "Y" AND S$ <>
       "y" THEN 990
NE 1020  LOCATE 20,1: PRINT "Ite
       m = ?": BEEP
BM 1030  S$ = INKEY$
JJ 1040  IF S$ = "" THEN 1030
BI 1050  Q = ASC(S$)
DF 1060  REM CAPS
ID 1070   IF Q > 96 THEN Q = Q -
       32
JD 1080  REM ELEMENT IN VECTOR
ML 1090   Q = Q-64
AG 1100   IF Q < 1 OR Q > Z THEN
       1030
KB 1110  REM NEW VALUE
MK 1120   LOCATE 22,1: BEEP: INP
       UT "New value = ";S$
DM 1130   X(Q,P) = VAL(S$)
PG 1140   GOSUB 850
CD 1150   GOTO 980
JP 1160 RETURN
EM 1170 REM COMPUTE TOTALS
```

```
JP 1180  CLS: LOCATE 13,35: PRIN
         T "Computing ..."
GH 1190  REM COMPUTE
CN 1200    REM SAVINGS
CF 1210      GOSUB 1320
BP 1220    REM EXPENSES
EK 1230      GOSUB 1510
NB 1240    REM TOTAL EXPENSES
FA 1250      GOSUB 1610
MA 1260  REM DISPLAY EXPENSES
QP 1270    TITLE$ = "TODAY'S":  G
         OSUB 1750
ID 1280    TITLE$ = "INFLATED": G
         OSUB 1750
NF 1290  REM DISPLAY TOTALS
PC 1300      GOSUB 2010
IE 1310  RETURN
EH 1320  REM SAVINGS
AD 1330  T = N+4
BI 1340  REM INTEREST RATE IN IN
         DEX FORM
GE 1350    RATE = 1 + X(1,3)/100
GE 1360  REM SAVINGS PER YEAR
KK 1370    SPY = 0
CN 1380    FOR J=1 TO N(1)
JB 1390      SPY = SPY + X(J,1)
BC 1400    NEXT
PL 1410  REM PRINCIPAL
OH 1420    SAVPRN = SPY*T
GF 1430  REM TOTAL (INCLUDES INT
         EREST)
JJ 1440    SAVTOT = 0
NB 1450    FOR I=1 TO T
BP 1460      SAVTOT = SAVTOT + SPY
         *RATE^(T-I+1)
CH 1470    NEXT
CK 1480  REM INTEREST
KE 1490    SAVINT = SAVTOT - SAVP
         RN
IF 1500  RETURN
JE 1510  REM EXPENSES
GI 1520  REM INFLATION RATE IN I
         NDEX FORM
HH 1530    RATE = 1 + X(2,3)/100
HE 1540    Q = N(2)
NI 1550  FOR I=1 TO Q
IC 1560    FOR J=1 TO 4
ID 1570      EK(I,J) = X(I,2)
DP 1580      EF(I,J) = X(I,2)*RATE
         ^(N+J)
CC 1590  NEXT J,I
CH 1600  RETURN
NG 1610  REM TOTALS
LH 1620  FOR I=1 TO 4
FH 1630    EK(Q+1,I)=0: EF(Q+1,I)
         =0
KF 1640    FOR J=1 TO Q
QK 1650      EK(Q+1,I) = EK(Q+1,I)
          + EK(J,I)
LD 1660      EF(Q+1,I) = EF(Q+1,I)
          + EF(J,I)
CO 1670    NEXT J,I
AC 1680  REM GRAND TOTALS
LK 1690    COSTK=0: COSTF=0
NI 1700    FOR I=1 TO 4
JN 1710      COSTK = COSTK + EK(Q+
         1,I)
BE 1720      COSTF = COSTF + EF(Q+
         1,I)
CB 1730    NEXT
JF 1740  RETURN
EL 1750  REM DISPLAY
JJ 1760  F$ = "$$#,####.##
FE 1770  H$ = STRING$(80,CHR$(20
         5))
FC 1780  X$(Q+1,2) = "Total
JK 1790  CLS
QP 1800  PRINT H$;
KE 1810  T$ = "COLLEGE EXPENSES:
          " + TITLE$ + " DOLLARS"
GL 1820  L = LEN(T$)
BI 1830  PRINT TAB(40-L/2)T$
HG 1840  PRINT H$
GO 1850  PRINT TAB(2)"Category";
         TAB(31)"1st Year";TAB(44
         )"2nd Year";TAB(57)"3rd
         Year";TAB(70)"4th Year
GM 1860  PRINT
QJ 1870  FOR I=1 TO Q+1
JD 1880    PRINT TAB(3)X$(I,2);TA
         B(28);
JB 1890    FOR J=1 TO 4
CI 1900      IF TITLE$ = "TODAY'S"
          THEN VL = EK(I,J) ELSE
         VL = EF(I,J)
LP 1910      PRINT USING F$;VL;: P
         RINT "  ";
BE 1920    NEXT J
MC 1930    PRINT
DP 1940    IF I = Q THEN PRINT
OI 1950  NEXT I
LN 1960  LOCATE 20,1: PRINT H$
MC 1970  LOCATE 23,34: PRINT "Pr
         ess any key
DN 1980  S$ = INKEY$
QP 1990  IF S$ = "" THEN 1980
IM 2000  RETURN
ML 2010  REM TOTALS
LL 2020  F$ = "= $$#,#######.##
HL 2030  CLS
OM 2040  PRINT H$;
QO 2050  PRINT TAB(33)"THE BOTTO
         M LINE
HM 2060  PRINT H$
CF 2070  PRINT TAB(16)"SAVINGS F
         OR COLLEGE
FD 2080  PRINT
JG 2090  PRINT TAB(18)"Principal
         ";TAB(40) USING F$;SAVPR
         N
NK 2100  PRINT TAB(18)"Interest"
         ;TAB(40) USING F$;SAVINT
IH 2110  PRINT TAB(18)"Total";TA
         B(40) USING F$;SAVTOT
NL 2120  LOCATE 12,16:PRINT "COS
         T OF COLLEGE
EG 2130  PRINT
JM 2140  PRINT TAB(18)"In today'
         s dollars";TAB(40) USING
          F$;COSTK
BK 2150  PRINT TAB(18)"In inflat
         ed dollars";TAB(40) USIN
         G F$;COSTF
ID 2160  LOCATE 18,16: PRINT "SA
         VINGS - INFLATED COST";T
         AB(40) USING F$;SAVTOT-C
         OSTF
KB 2170  LOCATE 20,1: PRINT H$
MG 2180  LOCATE 23,34: PRINT "Pr
         ess any key
CB 2190  S$ = INKEY$
EH 2200  IF S$ = "" THEN 2190
ID 2210  RETURN
JD 2220  REM DATA
PJ 2230  REM CATEGORIES
HP 2240    DATA Annual Savings, F
         irst-Year Expenses (Esti
         mate), Economic Assumpti
         ons
DD 2250  REM SAVINGS
FF 2260    DATA Parents, 1000
BB 2270    DATA Kids - summer job
         s, 500
ND 2280  REM EXPENSES
OD 2290    DATA Tuition,4000
NF 2300    DATA Room & Board,3500
QI 2310    DATA Books,300
OB 2320    DATA Travel To & From
         School,150
DG 2330    DATA Laundry,75
AH 2340    DATA Entertainment,250
JD 2350  REM ECONOMIC ASSUMPTION
         S
FN 2360    DATA Expected interest
          rate,10
JC 2370    DATA Expected inflatio
         n rate,5
```

©

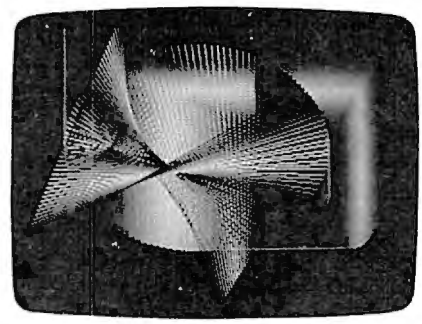
GraphiDemo For Amiga

Stefan Lindahl

This intriguing graphics program, written by a COMPUTE! reader in Sweden, demonstrates the the Amiga's tremendous graphics processing power as well as the speed of Amiga BASIC. 512K of memory is required

"GraphiDemo" demonstrates just how easy it is to create impressive graphics in Amiga BASIC. Type it in and save a copy of the program, then run it. GraphiDemo begins by displaying a help screen that explains all of the program's options. You can recall this screen at any time by pressing the Help key. Take a moment to look at all the different options—you'll want to try them all.

GraphiDemo's options can be invoked in two different ways. If you press the right mouse button and examine the menus at the top of the screen, you will see that every option can be selected from a menu, using the mouse pointer. However, GraphiDemo uses all of the Amiga's colors, which can make the menus unreadable at times. Thus, you can also select any option by pressing the key indicated in the help screen. If you forget which key is assigned to which option, simply press Help. When you exit the help screen, the main screen is restored to its original condition.



"GraphiDemo" can produce thousands of interesting designs similar to the one shown here.

Since the program is entirely self-prompting, no elaborate explanations are necessary. Just run it, follow the prompts, and enjoy the show. If you are interested in graphics programming, the program provides examples of how to draw different shapes and control the color palette for various effects.

GraphiDemo for Amiga

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
REM ** Copyright 1987 Compute! P
ublications, Inc.  **
REM ** All Rights Reserved **
'
CLEAR ,13000          :REM * Relea
se basic memory to system *
DEFINT b-y            :REM * Integ
er definition *
'
RANDOMIZE TIMER       :REM * New r
andom seed *
'
b=5                   :REM * Maxim
um step length *
cm=0                  :REM * Circl
emode off *
depth=4               :REM * No of
bitlayers *
'
SCREEN 2,640,200,depth,2
WINDOW 2,,,16,2
maxcoLor=2^depth-1
'
GOSUB SetcoL
'
GOSUB CLrmenu
'
ch=2:ch2temp=3:GOSUB 10:GOSUB 20
:REM * Set menus & check marks *
'
'
ON MENU GOSUB Mnuche
MENU ON
'
ON MOUSE GOSUB Chkmus
```

```
MOUSE ON
'
GOSUB Info        :REM * Display
info-window *
'
VarvaL:
x1=50+RND*540:y1=50+RND*120:x2=5
0+RND*540:y2=50+RND*120
IF x2<x1 OR y2<y1 THEN VarvaL
xs1=(1+RND*b):xs2=(1+RND*b):ys1=
(1+RND*b):ys2=(1+RND*b)
minx=0:maxx=629:miny=0:maxy=195
'
Main:
FOR doit=-1 TO 1 STEP 2
'
FOR cc=maxcoLor*-(doit=-1) TO ma
xcoLor*-(doit=1) STEP doit
oLdx1=x1:oLdy1=y1:x1=x1+xs1:y1=y
1+ys1
oLdx2=x2:oLdy2=y2:x2=x2+xs2:y2=y
2+ys2
'
IF x1<minx OR x1>maxx THEN x1=x1
-xs1:xs1=(1+RND*b)*-SGN(xs1)
IF x2<minx OR x2>maxx THEN x2=x2
-xs2:xs2=(1+RND*b)*-SGN(xs2)
IF y1<miny OR y1>maxy THEN y1=y1
-ys1:ys1=(1+RND*b)*-SGN(ys1)
IF y2<miny OR y2>maxy THEN y2=y2
-ys2:ys2=(1+RND*b)*-SGN(ys2)
key=ASC(INKEY$+CHR$(0)):IF key T
HEN GOSUB Keyche
ON choice GOSUB Serpent,Lines,Bo
xes,Filledboxes
NEXT cc
NEXT doit
'
GOTO Main
'
Serpent:
COLOR cc:IF cm THEN
GOSUB CircLepos:AREA STEP (0,0):
AREA (x1,y1):AREA (x2,y2)
AREAFILL
ELSE
AREA (x1,y1):AREA (oLdx1,oLdy1):
AREA (x2,y2):AREA (oLdx2,oLdy2)
AREAFILL
AREA (x1,y1):AREA (x2,y2):AREA (
oLdx1,oLdy1):AREA (oLdx2,oLdy2)
AREAFILL
END IF
RETURN
'
Lines:
IF cm THEN GOSUB CircLepos ELSE
PSET (x1,y1),cc
LINE -(x2,y2),cc
RETURN
'
Boxes:
IF cm THEN GOSUB CircLepos ELSE
PSET (x1,y1),cc
LINE -(x2,y2),cc,b
RETURN
'
Filledboxes:
IF cm THEN GOSUB CircLepos ELSE
PSET (x1,y1),cc
LINE -(x2,y2),cc,bf
RETURN
'
CircLemode:
CLS:cm=ABS(cm-1)
RETURN
'
CircLepos:
api=api+.05:PSET (314+200*SIN(ap
i),90+70*COS(api)),cc
RETURN
'
Mnuche:
IF MENU(0)=2 THEN menu2      :REM
* Menu 1 or 2 ?
menu1:
ch=MENU(1)
IF ch>4 THEN ON ch-4 GOTO SetcoL
,Info,Resetprog
10  GOSUB Setmenu
choice=ch:MENU 1,choice,2
RETURN
menu2:
ch2temp=MENU(1)
20  IF ch2=ch2temp THEN RETURN E
LSE ch2=ch2temp
21  GOSUB menuNo2:MENU 2,ch2,2
GOTO SetcoL
'
Keyche:                 '
IF key=27 THEN Resetprog
:REM * Esc  key *
IF key=139 THEN Info
:REM * Help key *
IF key=133 THEN SetcoL
:REM * F5   key *
IF key=127 THEN CLS
:REM * Del  key *
IF (key AND 223)=67 THEN CircLem
ode  :REM * 'C'  key *
IF key>133 AND key<137 THEN coLo
rmode
IF key<129 OR  key>132 THEN RETU
RN
ch=key-128:GOTO 10          :REM
* F1 - F4 *
coLormode:
ch2temp=key-133:GOTO 20     :REM
* F6 - F8 *
RETURN
'
Setmenu:
MENU 1,0,1,"   Main:"
MENU 1,1,1,"   Serpent     "
MENU 1,2,1,"   Lines       "
MENU 1,3,1,"   Boxes       "
MENU 1,4,1,"   Filledboxes "
MENU 1,5,1,"   New Colors  "
MENU 1,6,1,"   Help        "
MENU 1,7,1,"   Stop program"
RETURN
menuNo2:
MENU 2,0,1,"   ColorOptions:"
MENU 2,1,1,"   RGB <-> BLACK sha
ding"
MENU 2,2,1,"   Random color shad
ing "
MENU 2,3,1,"   Random colors
"
RETURN
'
CLrmenu:
MENU 1,0,0,""
MENU 2,0,0,""
MENU 3,0,0,""       '} Clear sta
ndard menus
MENU 4,0,0,""
RETURN
'
SetcoL:
ON ch2 GOTO SetcoL1,SetcoL2,Setc
oL3
'
SetcoL1:
CLS
c=RND*7+.5
c1=SGN(c AND 1)
c2=SGN(c AND 2)
c3=SGN(c AND 4)
FOR c=0 TO maxcoLor
PALETTE c,(c/16)*c1,(c/16)*c2,(c
/16)*c3
NEXT c
RETURN
'
SetcoL2:
CLS
FOR w=1 TO 3:a1(w)=RND:a2(w)=RND
:NEXT w
FOR w=1 TO 3:adeL(w)=(a2(w)-a1(w
))/(maxcoLor+1):NEXT w
'
FOR w=0 TO maxcoLor
PALETTE w,a1(1),a1(2),a1(3)
FOR wl=1 TO 3:a1(wl)=a1(wl)+adeL
(wl):NEXT wl
NEXT w
RETURN
'
SetcoL3:
FOR c=0 TO maxcoLor
PALETTE c,RND,RND,RND
NEXT c
RETURN
'
'
Resetprog:
MENU RESET
PALETTE 0,.4375,.125,.1875
PALETTE 1,1,.56,0
PALETTE 2,1,.1,.6
PALETTE 3,.44,.6,.94
WINDOW CLOSE 2
SCREEN CLOSE 2
CLS
END
'
Chkmus:
IF inf THEN inf=0 ELSE CLS
RETURN
'
Info:
MENU STOP:inf=1:REM* To tell mou
se-trapping routine
WINDOW 3,,(100,10)-(517,175),0,2
:REM that we're in Info
CLS:COLOR maxcoLor-2
PRINT SPACE$(5);"Copyright 1987
Compute! Publications, Inc."
PRINT SPACE$(16);"All Rights Res
erved":PRINT
PRINT SPACE$(20);"GRAPHIDEMO"
PRINT SPACE$(8);"F1 or Menu ....
.............. Serpent"
PRINT SPACE$(8);"F2 or Menu ....
.............. Lines"
PRINT SPACE$(8);"F3 or Menu ....
.............. Boxes"
PRINT SPACE$(8);"F4 or Menu ....
......... Filledboxes"
PRINT SPACE$(8);"F5 or Menu ....
.......... New Colors"
PRINT :PRINT SPACE$(8);"F6 or Me
nu ... RGB <-> BLACK shading"
PRINT SPACE$(8);"F7 or Menu ....
Random color shading"
PRINT SPACE$(8);"F8 or Menu ....
....... Random colors"
PRINT SPACE$(8);"'C' key .......
. toggle 'Circlemode'"
PRINT:PRINT "Clear Screen with L
eft Mouse button or the DEL key.
"
PRINT "  Stop the Program with t
he ESC key or from Menu."
PRINT:PRINT "Get this window bac
k with the HELP key or from Menu
."
PRINT:PRINT SPACE$(14);"PRESS AN
Y KEY TO CONTINUE";
Waithere:
IF INKEY$="" AND inf=1 THEN Wait
here    :REM * Check for key or
'
WINDOW CLOSE 3
:REM * mousebutton
MENU ON
RETURN
'
'
'
```

Font Loader
For Apple ImageWriter

Ed Thompson

This convenient utility allows you to preview a custom printer font on the screen before downloading it to an Apple ImageWriter printer. An ImageWriter I or ImageWriter II printer is required. The program runs on any Apple II-series computer, but only under ProDOS.

One powerful special feature of Apple's ImageWriter printers is the ability to print user-defined character sets, or *fonts*. A wide variety of font designs are available commercially and through Apple user groups. "Font Loader" is a utility for loading a standard printer font to an Apple ImageWriter I or ImageWriter II printer. You must have one or more ImageWriter font files to use this program. Font Loader doesn't have any provision for creating new fonts; it simply makes existing fonts easier to use.

Type in and save a copy of Font Loader. Before you run the program for the first time, create a disk subdirectory named FONTS and copy all of your font files into that subdirectory. When you run Font Loader, it loads the first font from the FONTS subdirectory and displays it on the screen.

At this point, you have several options as indicated by the on-screen prompts. Use the right- and left-arrow keys to cycle forward or backward through different fonts. Press Return to load and view a font, and press L to download a font to the printer. The P key prints the entire font on the printer. The C key clears a downloaded font from the printer, and R resets the printer.

Font Loader

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
BB 10 REM * COPYRIGHT 1987
      *
46 20 REM * COMPUTE! PUBLICATION
      S, INC.*
BC 30 REM * ALL RIGHTS RESERVED
      *
37 100 HIMEM: 136 * 256
CF 110 TEXT : PRINT : PRINT CHR$
       (21): GOSUB 1120: GOTO 5
       60
21 120 REM PRINT FONT
46 130 PRINT CHR$ (4)"PR#1"
E8 140 PRINT
42 150 PRINT CHR$ (4)"PR#0"
A8 160 E$ = CHR$ (27) + CHR$ (43
       ) + CHR$ (27) + CHR$ (73)
53 170 P = 768 + 12:S = 768 + 16
       + 6
A4 180 POKE 48688,12: POKE 48689
       ,3
F8 190 PRINT E$;
B5 200 ADD = 34816
48 210 FOR C = 32 TO 126
66 220 PRINT CHR$ (C)"B";
D8 230 POKE 1, INT (ADD / 256)
FC 240 POKE 0, INT (ADD - PEEK (
       1) * 256)
45 250 POKE 3,7
D3 260 CALL S
EA 270 ADD = ADD + 8
EB 280 NEXT C
45 290 PRINT CHR$ (4)
18 300 PRINT CHR$ (4)"PR#1
E4 310 PRINT
42 320 PRINT CHR$ (27)"'
78 330 IF FLAG < > 1 THEN 350
AE 340 FLAG = 0: GOTO 430
39 350 PRINT A$(X) + ".SET"
44 360 FOR I = 32 TO 63: PRINT C
       HR$ (I);: NEXT
F8 370 PRINT
BF 380 FOR I = 64 TO 95: PRINT C
       HR$ (I);: NEXT
F4 390 PRINT
39 400 FOR I = 96 TO 126: PRINT
       CHR$ (I);: NEXT
3B 410 PRINT CHR$ (27)"$
34 420 PRINT A$(X) + ".SET"
E9 430 PRINT
19 440 PRINT CHR$ (4)"PR#0
DD 450 POKE 48688,240: POKE 4868
       9,253
F0 460 POKE 0,76: POKE 1,60: POK
       E 76,1
24 470 GOTO 660
F8 480 REM LOAD FONT ONLY
A8 490 FLAG = 1
90 500 GOTO 120
49 510 REM RESET PRINTER
48 520 PRINT CHR$ (4)"PR#1"
95 530 PRINT CHR$ (27) + CHR$ (9
       9)
44 540 PRINT CHR$ (4)"PR#0"
21 550 GOTO 660
75 560 REM DEMO ROUTINE
A2 570 DATA  76,48,3,0,0,0,0,0,1
       41,144,192,96,41,127,16,2
       40,169,0,133,0,169,64,160
       ,0,177,0,106,145,0,102,2,
       200
BB 580 DATA 192,8,208,244,165,2,
       32,0,3,198,3,208,233,96,0
       ,0,72,173,153,192,41,48,2
       01,16,208,247,104,141,152
       ,192,96,240
F6 590 DATA 216,120,133,69,134,7
       0,132,71,166,7,10,10,176,
       4,16,62,48,4,16,1,232,232
       ,10,134,27,24,101,6,133,2
       6,144,2
0C 600 DATA 230,27,165,40,133,8,
       165,41,41,3,5,230,133,9,1
       62,8,160,0,177,26,36,50,4
       8,2,73,127,164,36,145,8,2
       30,26
77 610 DATA 208,2,230,27,165,9,2
       4,105,4,133,9,202,208,226
       ,165,69,166,70,164,71,88,
       76,240,253
5B 620 D$ = CHR$ (4)
FF 630 FOR I = 768 TO 919: READ
       J: POKE I,J: NEXT
1C 640 HIMEM: 150 * 256
04 650 PRINT CHR$ (4);"BLOAD/FON
       TLOADER/FONTS/ASCII.SET,A
       $8B00"
48 660 HIMEM: 136 * 256
```

```
A2 670 POKE 6,0: POKE 7,139
B4 680 HGR2
C1 690 PRINT CHR$ (4);"PR# A$340
   "
DF 700 GOSUB 1070
B6 710 X = 1
97 720 VTAB 16: PRINT "NEXT SET
   IS ";A$(X);".SET"
42 730 A$(1) = "APL"
F6 740 HIMEM: 150 * 256:D$ = CHR
   $ (4)
E6 750 PRINT D$;"BLOAD/FONTLOADE
   R/FONTS/"A$(X)".SET,A$880
   0"
49 760 HIMEM: 136 * 256
42 770 VTAB 5: PRINT A$(X) + ".S
   ET"
8D 780 POKE 6,0: POKE 7,136
0A 790 VTAB 8: FOR I = 32 TO 63:
    PRINT CHR$ (I);: NEXT
E7 800 PRINT
B6 810 VTAB 10: FOR I = 64 TO 95
   : PRINT CHR$ (I);: NEXT
EB 820 PRINT
21 830 VTAB 12: FOR I = 96 TO 12
   7: PRINT CHR$ (I);: NEXT
EF 840 PRINT
A0 850 POKE 6,0: POKE 7,139
A2 860 GET KY$
0A 870 IF KY$ = CHR$ (13) THEN 1
   010
D1 880 IF KY$ = CHR$ (69) THEN 1
   040
3D 890 IF KY$ = CHR$ (80) THEN G
   OTO 120
0D 900 IF KY$ = CHR$ (76) THEN G
   OTO 480
F0 910 IF KY$ = CHR$ (8) THEN 97
   0
D3 920 IF KY$ = CHR$ (67) THEN 5
   10
53 930 IF KY$ = CHR$ (21) THEN 9
   50
24 940 GOTO 860
A7 950 X = X + 1: IF X > NU THEN
    X = 1
AA 960 GOTO 980
16 970 X = X - 1: IF X < 1 THEN
   X = NU
59 980 VTAB 16: PRINT "
                "
A7 990 VTAB 16: PRINT "NEXT SET
   IS ";A$(X);".SET"
D7 1000 GOTO 860
C1 1010 HOME : CALL - 3092: GOSU
   B 1070
D2 1020 VTAB 16: PRINT "NEXT SET
   IS ";A$(X);".SET"
DE 1030 GOTO 740
FF 1040 REM END
34 1050 PRINT CHR$ (4)"PR#0"
C1 1060 TEXT : END
B7 1070 VTAB 2: PRINT "COPYRIGHT
   1987 COMPUTE! PUBLICATI
   ONS"
59 1075 VTAB 3: PRINT "
   ALL RIGHTS RESERVED"
0E 1080 VTAB 20: PRINT "USE <- -
   > TO MOVE FORWARD OR BAC
   KWARD"
5D 1090 VTAB 22: PRINT "'Return'
   -Select Font 'L'-Downloa
   d Font"
23 1100 VTAB 23: PRINT "'P'-Prin
   t Font 'C'-Clear Font  '
   E'-End"
D7 1110 RETURN
41 1120 REM READ FONT NAMES
49 1130 DIM A$(60):NU = 0
66 1140 D$ = CHR$ (4)
4E 1150 PR$ = "/FONTLOADER/FONTS
   "
A7 1160 PRINT D$"OPEN ";PR$;",TD
   IR"
B7 1170 PRINT D$"READ ";PR$
87 1180 INPUT N$: REM READ DIREC
   TORY NAME
EE 1190 INPUT T$: REM READ COLUM
   N TITLES
69 1200 INPUT L$: REM READ BLANK
    LINE
DA 1210 INPUT AA$: REM  READ FIL
   E NAME
D9 1220 NU = NU + 1
8C 1230 A$(NU) = AA$
EB 1240 IF AA$ < > "" THEN GOTO
   1210
E9 1250 NU = NU - 1
21 1260 INPUT B$: REM READ BLOCK
    COUNT
A1 1270 PRINT D$;"CLOSE ";PR$
A0 1275 TEXT : HOME
01 1280 VTAB 12: HTAB 13: INVERS
   E : PRINT "ANALYZING DAT
   A": NORMAL : VTAB 1: HTA
   B 1
1E 1290 FOR I = 1 TO NU
49 1300 A$(I) = LEFT$ (A$(I),15)
25 1310 IF RIGHT$ (A$(I),1) = "
   " THEN GOTO 1330
72 1320 GOTO 1350
81 1330 A$(I) = LEFT$ (A$(I), LE
   N (A$(I)) - 1)
6A 1340 GOTO 1310
8F 1350 A$(I) = LEFT$ (A$(I), LE
   N (A$(I)) - 4)
A4 1360 A$(I) = MID$ (A$(I),2, L
   EN (A$(I)))
4E 1370 PRINT D$"FRE"
91 1380 NEXT I
FB 1390 RETURN                    ©
```

News & Products

Commodore Peripherals

Two peripheral systems for the Commodore 64 have been announced by Computer Specialties (CSI).

The ST10C is a ten-megabyte hard drive system that is compatible with the Commodore 8050, 8250, 4040, and 1541 drives. It features unlimited directory space, 154 tracks with 256 sectors per track, IEEE and serial interface, built-in diagnostics for sector errors, built-in backup, an external format disable switch to prevent accidental erasure, and an external device switch. The internal drive read/write transfer rate is five million bps, and the track-to-track access time is three milliseconds. The ST10C has 10,092,544 formatted bytes and a 16,000-word DOS.

The C-64 Power Plus combines surge protection with power supply for your Commodore 64. It has one on/off control to turn on your computer and three peripherals at once. There is a single AC-switched power supply, and the short circuit current is limited to .75 amps for DC power. There's also overcurrent protection, over-temperature protection that starts at 56 degrees C, surge protection up to 470 volts AC, AC fuse protection externally mounted, and over-voltage protection to prevent burnout.

Suggested retail price for the ST10C hard drive is $895, and the Power Plus costs $59.95. CSI also offers a variety of other peripherals for the Commodore 64.

Computer Specialties, P.O. Box 1718, Melbourne, FL 32902-1718
Circle Reader Service Number 200.

Interactive Comics On The Apple II

Accolade has released *Accolade's Comics*, an interactive comic book for Apple II computers. Comprised of three disks and retailing for $44.95, the program features two distinct themes and dozens of major and minor story lines incorporating eight arcadelike games. Its main plot follows a spy named Steve Keene, whose mission is to thwart evil plots for the chief of Spystuff, Inc.

Unlike text adventures, *Accolade's Comics* offers the experience of reading a comic book on a computer screen, combining detailed graphics and humorous animation. The player determines the direction of the story by continually selecting from a series of possible answers to questions asked of Keene; some are dead ends, and others lead the player in more fruitful directions.

Accolade's Comics *is an interactive computer comic book available in Commodore and Apple formats.*

The first theme revolves around the kidnapping of Professor Zoron Farad, a Nobel Prize winner. The second focuses on a scheming underworld character named Zardo, who has developed a system by which fire hydrants reproduce, thereby getting people to park in garages instead of on the streets.

Accolade has also announced a Commodore 64 version of the program, set to retail for $39.95.

Accolade, 20813 Stevens Creek Blvd., Cupertino, CA 95014
Circle Reader Service Number 201.

Professional Keyboards For Macintosh

DataDesk International has announced two high-performance, full-function keyboards for the Apple Macintosh: the MAC 101/ADB and the MAC-101. Both keyboards use the industry standard 101-key layout, have a full complement of function keys, and are designed to meet the high-speed data input needs of corporate word processing, spreadsheet, and desktop publishing applications.

The MAC 101/ADB takes advantage of the features of the new Macintosh II and Macintosh SE computers. It has built-in Apple DeskTop Bus connectors, which allow daisy-chaining of up to 16 input devices, including a mouse, graphics tablets, and joysticks. The MAC-101 is designed for use with the installed base of Macintosh computers. Both keyboards are bundled with a macro-creating accessory program. They also take particular advantage of new Macintosh software, like *Microsoft Word 3.0*.

In addition to the 15 function keys, both MAC-101 keyboards offer a separate numeric pad, extra-large RETURN and SHIFT keys, two command and option keys for ease of operation with either right or left separate cursor cluster in the industry-standard inverted *T* arrangement, six separate file/edit keys, a separate dedicated cancel key, keyboard status indicator lights, and multikey rollover.

Each keyboard is available for $169.95.

DataDesk International, 7650 Haskell Ave., Van Nuys, CA 91406
Circle Reader Service Number 202.

Disk Utilities For Commodore 1571 Drive

Free Spirit Software has released a new utilities disk for the Commodore 128 computer and 1571 disk drive.

Super Disk Utilities includes two-drive and single-drive backup; File Unscratch, Create Autoboot, Lock and Unlock Files, and Write-Protect utilities; disk editor; CP/M Plus disk backup; and the ability to print in either hexadecimal or ASCII to any sector on disk. Many other utilities are included, many of which also work on the 1541 drive.

Super Disk Utilities is available for $39.95.

Free Spirit offers a varied line of programs for the Commodore 64, including *Super 64 Librarian*, a disk cataloging and library system for the 64 and 1541 drive ($29.95); *The Weapon of Choice*, a text adventure game ($29.95); *Wheel & Deal*, a fast-paced real estate game (19.95); *Fun Biorhythms*, a program that lets you print personalized biorhythm charts using Commodore

and Commodore-compatible printers ($9.95); and Strider's Computer Classics, a new series of classical music disks, each of which contains about one hour of classical music and 40 screens of commentary ($9.95 each).

Free Spirit Software, 538 S. Edgewood, LaGrange, IL 60525
Circle Reader Service Number 203.

Desktop Organizer For IBM PC And Compatibles

First Avenue, a desktop organizer, helps save time as well as simplify and organize your day better through its accessible menu, organizer, and communication features.

The Desktop Organizer features include an autodialer, memo pad/letter writer, to-do lists, directory, index card catalog, and calculator. The Software Librarian lets you easily load a program from your software library when cataloged within the library function. And the Micro Networker allows you to connect up to five computers together to perform four functions: electronic mail, file transfer, phone-message handling, and schedule updating.

First Avenue, available for MS-DOS machines, retails for $69 (copy-protected) or $99 (unprotected).

Times Square is an add-on organizer-to-go that makes work done within *First Avenue* portable; it includes software, a binder, and insert tabs for $29.95.

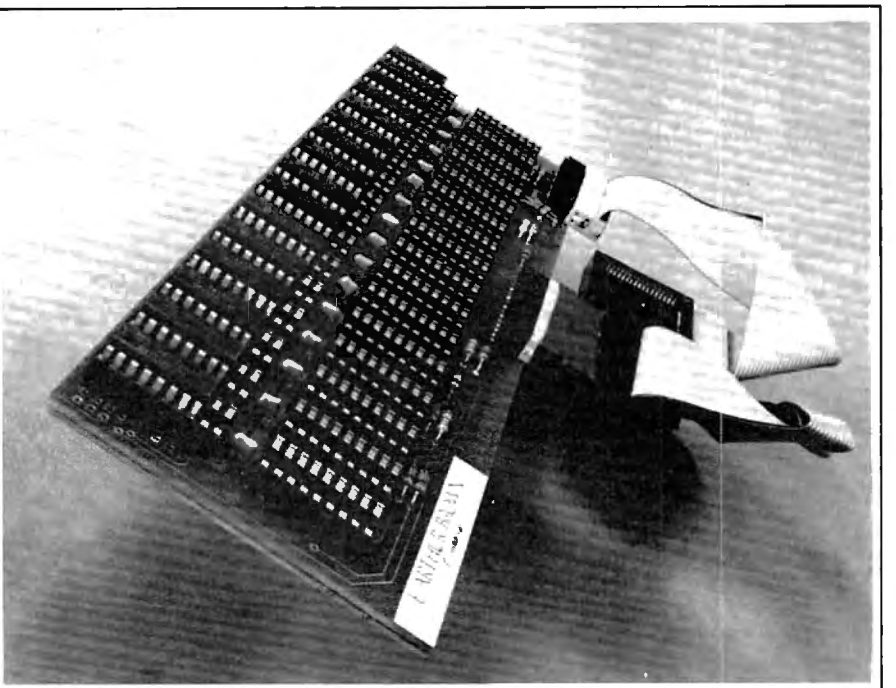
CANAL Systems, 5230 Clark Ave., Lakewood, CA 90712
Circle Reader Service Number 204.

New Graphics Disk From Epyx

Epyx has introduced a new collection of graphics compatible with the popular packages *Print Shop, PrintMaster, IBM PrintMaster Plus*, and all other print packages that accept fonts and borders from other disks. Geared to educational applications, *Graphics Scrapbook Chapter III: School* offers art for newsletters, banners, flyers, cards, and posters. It covers a wide variety of subjects, including geometry, drivers' education, student-body elections, band practice, school plays, cheerleading tryouts, dances, fund-raisers, rallies, graduation, and more.

The program is available for Commodore 64, Apple II series, and IBM PC and compatible computers. (Epyx does not set suggested retail prices for products.)

Epyx, 600 Galveston Dr., P.O. Box 8020, Redwood City, CA 94063
Circle Reader Service Number 205.



The ST Solderless RAM adds up to 4 megabytes to the Atari ST.

RAM Expansion For The Atari ST

A new 1–4 megabyte upgradable RAM add-on has been announced by the E. Arthur Brown Company for the Atari ST. The unit features solderless plug-in installation, and comes with enough RAM to upgrade a 520ST to 1 megabyte. Upgrading to 2.5 and 4 megabytes is simply a matter of plugging 1-megabyte RAM into the empty sockets.

Suggested retail price for the ST Solderless RAM is $199.95.

E. Arthur Brown, 3404 Pawnee Dr., Alexandria, MN 56308
Circle Reader Service Number 206..

DLM Educational Software

DLM has released Commodore 64 and Apple II versions of the new deluxe package for its *Create with Garfield*, a program for designing and printing cartoons based on the popular Garfield cartoon character. This version contains two disks (program and graphics library) that offer special features, including more than 200 pieces of artwork; a wide variety of typefaces for writing captions and stories; color printing capabilities; and an electronic comic strip, in which cartoons move continuously across the screen. The program, which was designed for both the home and school markets, offers users the chance to create their own posters, cartoons, labels, invitations, and other similar items. The original version is available for $29.95, and the deluxe version, for $39.95.

DLM has also announced *Teddy Bear*rels of Fun*, a new two-disk program for the Commodore 64 and Apple II family that makes it easy for both youngsters and adults to design and produce charming teddy bear artwork. It contains more than 200 graphics, including teddy bear characters, backgrounds, scenes and props, as well as a variety of borders and typefaces for writing stories, messages, and captions. Suggested retail price is $39.95.

Decimal Discovery and *Fast-track Fractions* introduce DLM's new Intermediate Math Series, modeled after similar DLM programs designed to develop other skill areas. The programs, available for the Apple II family, retail for $46 each.

DLM Teaching Resources, One DLM Park, Allen, TX 75002
Circle Reader Service Number 207.

Clip Art For Atari ST

More! Graphics ST is a collection of 128 icons, symbols, letters, and other clip art that can supplement many other ST graphics programs, including *Printmaster, Printmaster Plus, DEGAS, DEGAS Elite, Typesetter Elite, PM Interface*, and *Publishing Partner*. Suggested retail price for the Atari ST version is $14.95.

An eight-bit Atari version that offers 128 icons and 11 screen-magic files compatible with *Print Shop, PS Interface*, and compatible programs, is available for $12.95.

(When ordering direct, add $3 shipping and handling charge. Add $2 more for C.O.D. orders. California residents add 6% tax.)

Extended Hours M-F 9a.m.-9p.m. CST

PC COMPATIBLE

Extended Hours SAT 11a.m.-5p.m. CST

COMMODORE PC-10
INCLUDES

512K RAM
360K D/S DRIVE
CGA BOARD
PARALLEL & SERIAL I/O
GW BASIC AND MS DOS
SIDEKICK BY BORLAND
1 YEAR WARRANTY

$679

COMMODORE PC10-2
INCLUDES

640K RAM
2-360K D/S DRIVE
CGA BOARD
PARALLEL & SERIAL I/O
GW BASIC AND MS DOS
SIDEKICK BY BORLAND
1 YEAR WARRANTY

$819

COMMODORE TTL GREEN MONITOR $99.95

ZUCKERBOARDS (Made In USA)

CLOCK CALENDAR 49.95
256K EXPANSION BOARD 79.95
MONO GRAPHICS BD (HERCULES COMP.) 99.95
COLOR GRAPHICS BD (CGA) 99.95

PRINTER EXTRAVAGANZA

PANASONIC

Panasonic 1080I 199
Panasonic 1091I 259
Panasonic 1092I 349
Panasonic 3131 279
Panasonic 3151 409
Panasonic 1592 419
Panasonic 1595 539

OKIDATA

Okidata 120 215
Okidata 182 249
Okidata 192+ 369
Okidata 193+ 559
Okidata 292E 449
Okidata 293E 599

EPSON

LX 86 219
FX 86E 399
EX 800 479
EX 1000 629
FX 286E 568
LQ 800 519
LQ 1000 729
LQ 2500 1149

STAR MICRONICS

NX-10 . . . . . Call
NL-10 . . . . . Call
NP-10 . . . . . Call
NX-15 339
ND-15 449
NR-15 539
Powertype 189
NB 24/15 (24 Wire) . . . . Call

CANON

CANON A-50 279
CANON A-55 389
CANON A-60 349

SEIKOSHA

SP-180 169
SP-180VC 189
SP-1000VC 185
SP-1000A 199
SP-1200 209
(MINIMUM $8.00 SHIPPING IN CONTINENTAL USA)

BROTHER

BROTHER 1009 139
BROTHER 1509 379
BROTHER 1709 489
BROTHER HR 35 749
BROTHER 2024 LD 819
TWINWRITER 5 839

COMMODORE 64/128 SUPER PRINTER PACKAGES

NX-10 PRINTER & XETEC SUPER GRAPHIC INTERFACE
$259

PANASONIC 1080I PRINTER & XETEC SUPER GRAPHIC INTERFACE
$249

PANASONIC 1091I PRINTER & XETEC SUPER GRAPHIC INTERFACE
$309

SEIKOSHA SP1000VC PRINTER WITH BUILT-IN GRAPHICS INTERFACE
$165
MINIMUM $8.00 SHIPPING

ATARI XE/XL SUPER PRINTER PACKAGES

NX-10 PRINTER & SUPRA 1150 INTERFACE
$259

PANASONIC 1080I PRINTER & SUPRA 1150 INTERFACE
$249

PANASONIC 1091I PRINTER & SUPRA 1150 INTERFACE
$309

PANASONIC 1092I PRINTER & SUPRA 1150 INTERFACE
$379

SUPER COMPUTER PACKAGES

ATARI 130XE COMPUTER
$134
(LIMITED TO FIRST 50 ORDERS)

COMMODORE 128 COMPUTER
$239

1571 DISK DRIVE
$239

COMMODORE 64C COMPUTER WITH GEOS
$159

COMMODORE 128 PACKAGE
- COMMODORE 128 COMPUTER
- 1571 DISK DRIVE
- 1902 RGB MONITOR

$749
(LIMITED TO FIRST 100 ORDERS)

COMMODORE 64C PACKAGE
- COMMODORE 64C COMPUTER
- 1541C DISK DRIVE
- 1802C MONITOR
- GEOS

PRICE TOO LOW TO ADVERTISE!!

ATARI 1040ST SYSTEM PACKAGE
- INCLUDING RGB OR MONOCHROME MONITOR, MOUSE, DOUBLE-SIDED DISK DRIVE, BASIC, TOS ON ROM, AND BUILT-IN POWER SUPPLY
- FULL MANUFACTURER'S WARRANTY APPLIES

CALL FOR LATEST PRICE

ATARI 520ST SYSTEM PACKAGE
- INCLUDING RGB OR MONOCHROME MONITOR, MOUSE, DISK DRIVE, BASIC, TOS ON ROM, AND RF MODULATOR
- FULL MANUFACTURER'S WARRANTY APPLIES

CALL FOR LATEST PRICE

AMIGA SYSTEM PACKAGE
- AMIGA 1000 COMPUTER
- AMIGA 1080 MONITOR

$849
FACTORY RENEWED

OKIMATE 20 & PLUG 'N PRINT
INTERFACES AVAILABLE FOR APPLE*ATARI ST AMIGA*IBM*C-64/128
$199

MISCELLANEOUS HARDWARE

COMMODORE 64/128

1541C Disk Drive . . . . . Call
1670 Modem 119
1351 Mouse 44.95
1750 Ram Expander 159
1902 RGB Monitor 279
Xetec Super Graphic 69.95
Avatex 1200 HC Modem 125

ATARI XE•XL•400•800

1050 Disk Drive 135
P.R. Connection 62.95
Animation Station 49.95
Atari XM-301 . . . . . 32.95
Supra 1150 Interface . . . . . Call
850 Interface . . . . . Call
US Doubler 49.95
Graphic Al Interface 39.95
Avatex 1200 HC Modem 125.00

ATARI 520ST•1040ST

SH 204 20 MEG Hard Drive . . . . . Call
SF314 DS/DD Disk Drive . . . . . Call
Supra 20 MEG Hard Drive 559.00
1040 Dust Cover . . . . . Call
520 Dust Cover . . . . . Call
Z-Time 520 . . . . . 39.95
Z-Time 1040 . . . . . 49.95
Avatex 1200 HC Modem 125.00
Computereyes 189.00

APPLE

Grappler+ 74.95
Grappler+/W16K 119.95
Serial Grappler 74.95
Mach II Joystick 27.95
Mach III Joystick 34.95

CALL FOR OTHER APPLE HARDWARE

Extended Hours M-F 9a.m.-9p.m. CST

COMMODORE 64 SOFTWARE

Extended Hours SAT 11a.m.-5p.m.

BERKLEY SOFTWORKS

GEOS 37.95
Writer's Workshop 32.95
Geodex 25.95
Desk Pak I 22.95
Font Pak I 19.95
Geopublish 34.95
Geofile 34.95
Geocalc 34.95
Geochart 27.95
Desk Pak 2 . . . . . 32.95

ACCESS

Mach V-Cart 21.95
Leader Board 24.95
Tournament Disk . . 14.95
Executive Disk 14.95
WLD Class Lb. 25.95
Tenth Frame 24.95
Famous Courses . . 14.95

MICROPROSE

See Atari 130XE section for items and prices.

MISCELLANEOUS COMMODORE 64

Final Cartridge . . . 39.95
Bureacracy 22.95
Ultima 2 . . . . . 32.95
Phantse 1,2 or 3 . . 27.95
Shanghai 22.95
Ghost/Goblins 19.95
Rebel Charge 32.95
Superstr Hockey . . 22.95
Falcon 18.95
PT Boat 18.95
Carmen-USA 29.95
Sentry 25.95
XE Alpha Mision . . 19.95
Blue Powder Gry . . 32.95
Gunship 22.95
Leather Goddesses 22.95
Moonmist 22.95
Hollywood Hijinks . 22.95
Fontmaster 2 . . . . . 29.95
Disney Comic Strip 23.95
Disney Card&Party 23.95
WWF Wresting 25.95
Thunder Chopper . 19.95
UP Periscope 19.95
The Pawn 25.95
Starglider 25.95
Sublogic Football . 27.95
Flight Simulator II . 33.95
Sublogic Baseball . 33.95
Printmaster 21.95
Art Gallery 1 Or 2 . 16.95
Superscript 64 . . . . 32.95
Superbase 64 . . . . 46.95
Hardball 18.95
Print Shop 28.95
P.S. Companion . . . 22.95
Graph. Lib 1,2 or 3 16.95
Fist 19.95
Parallax 19.95
Uridium 19.95
Uchi Mata 19.95
Wizard's Crown . . . 27.95
Roadwar 2000 . . . . 27.95
Warship 39.95
Wargame Constr. . . 20.95
Shard Of Spring . . 27.95
War In S. Pacific . . 39.95
Rings Of Zilfin . . . 27.95
2 On 2 Basketball . 22.95
Champ Baseball . . 19.95
GFL Football 22.95
Transformers 19.95
Gamemaker 25.95
Accolade Comics . 24.95
Phantasie 3 . . . . . 27.95
Pirates 25.95
Battlecruiser 39.95
Express Raider . . . 22.95
Realms/Darkness . 27.95
S.D.I. 22.95
Defender/Crown . . 22.95
Copy II 64 . . . . . 27.95

ACABUS 64

Chartpak 25.95
Cad-Pak 25.95
Powerplan 25.95
Super C Compiler 39.95
Super Pascal 39.95
Basic Compiler . . . 25.95
Forth 25.95
Geos Inside & Out 18.95
Personal Portfolio 25.95
Xper System 39.95

TIMEWORKS

Business Sytems 32.95
Wordwriter/Spell . . 32.95
Swiftcalc/Sideways 25.95
Data Manager II . . . 25.95
Sylvia Porter 32.95
Evelyn Wood 25.95
Sideways 19.95
Partner 64 . . . . . 39.95

SPRINGBOARD

Newsroom 30.95
Certificate Maker . . 30.95
Graphics Expander 21.95
Clip Art 1 Or 3 . . . . 18.95
Clip Art 2 . . . . . 21.95
Cert. Maker Lib 1 . . 21.95

ELECTRONIC ARTS

Archon 10.95
Archon II 10.95
Earth Orbit . . . . . Call
Fin. Cookbook 10.95
Heart Of Africa . . . 10.95
Mail Ord. Monsters 10.95
Mind Mirror 10.95
Movie Maker 10.95
Mule 10.95
Music Constr. 10.95
One On One 10.95
Pinball Constr. 10.95
Racing Destr. 10.95
Seven Cities of Gold 10.95
Skyfox 10.95

ELECTRONIC ARTS

Super Bould Dash 10.95
Touchdown Football 10.95
Ultimate Wizard . . . 10.95
Adv. Constr. 25.95
America's Cup 22.95
Arctic Fox 22.95
Autoduel 31.95
Bard's Tale 25.95
Bard's Tale II 25.95
Chickamauga 23.95
Battlefront 25.95
Carriers At War . . . 31.95
Europe Ablaze 31.95
Lords Of Conquest 22.95
Marble Madness . . 20.95

ELECTRONIC ARTS

Moebius 25.95
Murder Party 22.95
Ogre 25.95
Pegasus 23.95
Reach for the Stars 30.95
Road To Moscow . . 25.95
Robot Rascals 25.95
Russia 25.95
Scrabble 22.95
Chessmaster 2000 25.95
Ultima 1 . . . . . 25.95
Ultima 3 . . . . . 31.95
Ultima 4 . . . . . 38.95
World Tour Golf . . . 23.95

EPYX

Destroyer 22.95
Fast Load 22.95
Movie Monster 22.95
Champ Wrestling . . 22.95
World Games 22.95
Winter Games 22.95
Summer Games . . . 14.95
Apshai Trilogy 14.95
Vorpal Utility 14.95
Multiplan 27.95
Super Cycle 22.95
St. Sports Basket 22.95
St. Sports Baseball 22.95
Summer Games 2 22.95
Sub Battle 22.95

EST. 1982
ComputAbility™

P.O. Box 17882, Milwaukee, WI 53217
ORDER LINES OPEN
Mon.-Fri. 9 a.m. - 9 p.m. CST • Sat. 11 a.m. - 5 p.m.

To Order Call Toll Free

800-558-0003

For Technical Info, Order Inquiries, or for Wisc. Orders

414-357-8181

TELEX NUMBER 9102406440
(ANSERBACK • COMPUT MILW UQ)

COMMODORE 128 SOFTWARE

Superbase 128 . . . . 57.95
Superscript 128 . . . 46.95
Wordwriter 128 . . . . 42.95
Data Manger 128 . . . 42.95
Swiftcalc 128 . . . . . 42.95
Sylvia Porter 128 . . 42.95
Partner 128 . . . . . 42.95
Fleet System 4 . . . . 46.95
Perfect Writer 49.95
Pocket Writer 2 . . . 36.95
Pocket Filer 2 . . . . . 36.95
Pocket Planner 2 . . 36.95
Basic Comp. 128 . . 39.95
Cad Pak 128 . . . . . 39.95
Super C Compiler 39.95
Cobol 128 . . . . . 39.95
Super Pascal 128 . . 39.95
Speed Term 128 . . . 25.95
Chart Pak 128 . . . . 25.95
Fontmaster 128 . . . 36.95
Multiplan 128 . . . . . 27.95
Vizawrite 128 . . . . . 54.95
Vizastar 128 . . . . . 54.95
Merlin 128 . . . . . 69.95

Extended Hours M-F 9a.m.-9p.m. CST — IBM SOFTWARE — Extended Hours SAT 11a.m.-5p.m.

ACTIVISION
Alter Ego 31.95
Borrowed Time 25.95
Hacker 11.95
Tass Times 22.95
Shanghai 22.95
Portal 29.95
GFL Football 25.95
2/2 Basketball 25.95
Champ Baseball 25.95
Ghostbusters 11.95
Writer's Choice 25.95
Planner's Choice 25.95
Filer's Choice 25.95
Choice Collection 64.95
Term Paper Wter 32.95

SIERRA
King's Quest I, II or III 31.95
Smart Money 32.95
Space Quest 31.95
Black Cauldron 25.95
Helicopter Simul 25.95
Police Quest ..... Call

BRODERBUND
Print Shop 37.95
Karateka 21.95
Graphic Lib 1 or 2 21.95
P.S Companion 31.95
Toy Shop 42.95
Type 31.95
Variable Feast 39.95
Fin Independence 94.95
For Comment 124.95
Carmen-USA 29.95
Carmen-World 25.95

SPRINGBOARD
Certificate Maker 38.95
Newsroom Pro 62.95
SEE APPLE SECTION FOR REST OF ITEMS & PRICES

EPYX
SEE APPLE SECTION FOR ITEMS & PRICES

ELECTRONIC ARTS
Archon 10.95
Fin Cookbook 10.95
Music Construct 10.95
One on One 10.95
Pinball Constr. 10.95
7 Cities of Gold 10.95
Super Biderdash 10.95
Mind Mirror 10.95
Amnesia 29.95
Gr. Slam Bridge 39.95
Murder Party 25.95
Ogre 25.95
Radio Baseball 25.95

MINDSCAPE
Bop/Wrestle 19.95
Indoor Sports 19.95
Infiltrator 19.95
Balance/Power 31.95
Sub Mission 25.95
Amer Challenge 25.95
Racter 25.95
Mastertype 25.95
Crossword Magic 31.95
Perfect Score 49.95

ELECTRONIC ARTS
Scrabble 25.95
Starfleet I 34.95
Chessmaster 25.95
Ultima III 38.95
Ultima IV 39.95
Wrld Tr Golf (NoAp) 34.95
Starflight (NoAp) 34.95
M. Beacon Typing 25.95
Arctic Fox ..... Call
Starfleet 2 ..... 34.95
Ultima 1 ..... 26.95
Lords/Conquest ..... Call
Marble Madness ..... Call

TIME WORKS
Wordwriter 39.95
Swiftcalc 39.95
Data Manager 39.95
Sylvia Porter 64.95
Swiftax 44.95
Partner 39.95
PC Powerpack 77.95

INFOCOM
Ballyhoo 25.95
Hitchiker 19.95
Leather Goddess 25.95
Trinity 25.95
Hollywood Hijinx 25.95
Moonmist 25.95
Zork Trilogy 44.95
Bureaucracy 25.95

ACCOLADE
Hardball 24.95
Mean 18 ..... 31.95
PSI Trading Co. 24.95
Course Disk 16.95
Ace Of Aces 24.95

SSI
Rings of Zilfin 25.95
Kampfgruppe 39.95
Shard/Spring 25.95
Roadwar 2000 ..... 25.95
Phantasie 25.95

MISCELLANEOUS IBM
SDI 25.95
Defender/Crown 25.95
Alternate Reality 31.95
Flight Sim. II 38.95
Strip Poker 25.95
Orbiter 31.95
Pawn 29.95
Starglider 28.95
7th Spirits 29.95
Champion Golf 25.95
Airplane Constr 25.95
Conflict/Vietnam 25.95
Decision/Desert 25.95
F-15 Strike Eagle 21.95
Silent Service 21.95
Crusade/Europe 25.95
Gunship 22.95

UNISON WORLD
Printmaster 36.95
Art Gallery 1 ..... 24.95
Art Gallery 2 ..... 18.95
Newsmaster 62.95

Universe II 47.95
Wizardry 39.95
Up Periscope 19.95
Thunder Chopper 19.95
Jet 34.95
Sublog Baseball 34.95
Sublog Football 34.95
Falcon 31.95
Business Card Mkr. 36.95
Bumper Stick Mkr. 31.95
Button Maker 31.95
Millionaire 2 ..... 38.95
Generic Cadd 2.0 ..... 59.95
Isgur Portfolio 149.95
Silk ..... Call
Managing/Money 119.95
PT Boat 31.95
BL Powder Gray 32.95
Typesetter PC 48.95
Megafont PC 24.95
Mercenary 24.95
Tenth Frame 25.95
Commando 25.95
Tag Team Wrest 25.95

Extended Hours M-F 9a.m.-9p.m. CST — APPLE SOFTWARE — Extended Hours SAT 11a.m.-5p.m.

EPYX
Sports Basketball 25.95
Sports Baseball 25.95
World Karate 19.95
World Games 25.95
Destroyer 25.95
Movie Monster 25.95
Winter Games 25.95
Summer Games II 25.95
Apshai Trilogy 25.95
Sub Battle 25.95
Epyx Joystick 25.95

DATASOFT
Black Magic 16.95
Old Scores 19.95
Tobruk 22.95
Sworda/Sorcery 19.95
Bismarck 22.95

ELECTRONIC ARTS
Age/Adventure 10.95
Archon II 10.95
Movie Maker 10.95
Skyfox 10.95
Adv. Constr. 32.95
Artic Fox 25.95
Autoduel 32.95
Bard's Tale 29.95
Bard's Tale II 32.95
Lord/Conquest 25.95
Marble Madness 23.95
Moebius 36.95
Pegasus 25.95
Russia 26.95
Ultima I 26.95

SEE IBM SECTION FOR REST OF ITEMS AND PRICES

BRODERBUND
Airheart 22.95
Animate 44.95
Print Shop 31.95
On Balance 25.95
Toy Shop 39.95
Type 28.95
Science Tool Kit 44.95
Graph Lib 1,2 or 3 16.95
Bank ST Series ..... Call
Carmen-World 25.95
Carmen-USA 29.95

DATA EAST
Karate Champ 15.95
Kung Fu Master 15.95
Commando 25.95
Tag Team Wrest 25.95

INFOCOM
SEE IBM SECTION FOR ITEMS AND PRICES

MINDSCAPE
SEE IBM SECTION FOR ITEMS AND PRICES

SPRINGBOARD
Certificate Maker 31.95
Clip Art 1 or 3 ..... 19.95
Clip Art 2 ..... 25.95
Certificate Lib 1 ..... 22.95
Games/Children 22.95
Graphics Expander 25.95
Newsroom 37.95
Piece of Cake Math 22.95

APPLE MISCELLANEOUS
Gettysburg 39.95
Warship 39.95
Phantasie 25.95
Battle/Antietnam 31.95
Phantasie II 25.95
Battlecruiser 39.95
Realms/Darkness 25.95

ACTIVISION
Gamemaker 31.95
Hacker II 25.95
Labyrinth 19.95
Little Computer 11.95
Greeting Card 25.95
SEE IBM SECTION FOR REST OF ITEMS AND PRICES

Fight Night 21.95
Hardball 21.95
PSI Trading 21.95
Comix 31.95
The Hobbit 23.95
Up Periscope 19.95
Thunder Chopper 19.95
Alt. Reality 24.95
Math Blaster 31.95
Spell It 31.95
Champ Wrestling 25.95
Wizard's Crown 25.95
Gemstone Healer 19.95
Word Attack 31.95
Elite 23.95
Pawn 29.95
Starglider 29.95
Gunship 25.95
F-15 Strike Eagle 22.95
Silent Service 22.95

Smart Money 49.95
Sublog. Baseball 34.95
Sublog Football 34.95
Jet 27.95
Space Quest 31.95
Animat. Station 59.95
Printmaster + ... 24.95
Art Gallery 1 or 2 ..... 18.95
Publisher 89.95
War in S. Pacific 39.95
Star Trek II 25.95
Roadwar 2000 ..... 25.95
Phantasie 3 ..... 25.95
Business Card Mkr. 36.95
Bumper Stick Mkr. 31.95
Button Maker 31.95
Rebel Charge 31.95
Might N' Magic 35.95
Oper Copernicus 25.95
Internat Hockey 17.95

Extended Hours M-F 9a.m.-9p.m. CST — ATARI ST SOFTWARE — Extended Hours SAT 11a.m.-5p.m.

ABACUS
Textpro 32.95
Datrieve 32.95
Text Designer 32.95
Assempro 39.95
Powerplan 49.95
Paintpro 32.95
Abacus Books ..... Call

ST GRAPHICS
Degas Elite 48.95
Easy Draw 48.95
Cad 3-D 36.95
Graphic Artist 124.95
Paintworks 25.95
1st Cadd 31.95
Aegis Animator 48.95
Neochrome 34.95
Computereyes 189.00
Stereo Cad 3D 67.95

ST TELECOMMUNICATION
St Talk Ver 2.0 ..... 18.95
BB/ST 31.95
Flash 27.95
I.S Talk 31.95

ST EDUCATIONAL
Decimal Dungeon 24.95
Fraction Action 24.95
Kinderama 24.95
Read & Rhyme 24.95
Animal Kingdom 24.95
Speller Bee 31.95
Kid Talk 31.95
Math Talk 31.95
First Shapes 31.95
Winnie The Pooh 16.95
Donald Duck 16.95
Buzzword 29.95

ST DATABASES
DB Man 96.95
Regent Base 57.95
Data Manager St 48.95
Zoomracks II 96.95
Superbase Gem 94.95

ST WORD-PROCESSORS
Regent Word II 48.95
Wordwriter St 48.95
Thunder 24.95

MICHTRON
Cornerman 31.95
Echo 24.95
M-Disk 24.95
Major Motion 24.95
Michtron Utilities 37.95
Cards 24.95
Time Bandits 24.95
Pers. Money Mgr. 31.95
Pinball Factory 24.95
Eight Ball 18.95
Back-Up 24.95
Match-Point 24.95
Karate Kid II 24.95
GFA Basic 48.95
Trimbase 62.95
Space Shuttle 2 ..... 24.95
M-Cache 24.95
GFA Draft 62.95
Trivia Challenge 24.95
Journ To Lair 32.95
GFA Compiler 48.95

INFOCOM ST
SEE IBM SECTION FOR ITEMS & PRICES

ST BUSINESS
VIP Professional ..... Call
Swiftcalc St. 48.95
Isgur Portfolio 124.95
DAC Payroll 32.95
DAC Easy Accting 44.95
Dollars And Sense 64.95
Sylvia Porter 48.95
A-Calc Prime 39.95
Logistix Jr. 64.95

ST UTILITIES
Music Studio 32.95
Time Link 31.95
Micro Cookbook 32.95
CZ Droid 69.95
EZ Track 39.95
Write 90 ..... 18.95
Desk Cart 72.95
Lib 1/Cert Maker 21.95
Labelmast Elite 27.95

DESKTOP PUBLISHING
Drawrite 99.95
Publish. Partner 99.95
Certificate Maker 31.95

ST ADVENTURES
Sundog 24.95
Universe II 44.95
The Pawn 29.95
Starglider 29.95
Dungeonmaster 24.95
Alt. Reality 24.95
Tass Times 25.95
Mercenary 24.95
Autoduel 32.95
Ogre 25.95
Defender/Crown 32.95
S.D.I. 32.95
Balance Of Power 32.95
Ultima III Or IV 38.95
Portal 32.95
Kings Quest 1,2 or 3 ..... 32.95
Space Quest 32.95
Tracker 29.95
Colonial Conquest 25.95
Roadwar 2000 ..... 25.95
Wizard's Crown 25.95
Rings Of Zilfin 25.95
Phantasie 3 ..... 25.95
Bard's Tale ..... Call
Phantse 1,2 or 3 ..... 25.95

ST LANGUAGES
Personal Pascal 49.95
Mark Williams C. 114.95
Lattice C. 99.95
Cambridge Lisp. 139.95
LDW Basic 44.95
Fortran 77 ..... 124.95

ST PRINT UTILITIES
Rubber Stamp 24.95
Printmaster Plus 24.95
Megafont St. 24.95
Art Gallery 1 Or 2 18.95
Typesetter Elite 31.95
PM Interface 18.95

ST ARCADE GAMES
Winter Games 24.95
Rouge 24.95
Super Huey 25.95
Mean 18 ..... 25.95
Famous Course 14.95
Leader Board 25.95
Silent Service 25.95
Flight Simulator II 33.95
Champ Wrestling 24.95
World Games 24.95
Chessmaster 2000 32.95
Tenth Frame 24.95
Shanghai 25.95
Micro Baseball 39.95
WWF Wrestling 39.95
Skyfox 29.95
Super Cycle 24.95

Indoor Sports 32.95
F-15 Strike Eagle 27.95
High Roller Sim 32.95
Two/Two Basketball 25.95
Deep Space 29.95
Arena 22.95
Champ Baseball 25.95
GFL Football 25.95
Harrier Strike 31.95
Sub Battle Sim 24.95
Arctic Fox 26.95
RPV 24.95
Crystal Castles 20.95
Gridiron 42.95
Hardball 24.95
Palon Chess 36.95
Crystal Castles 20.95

Extended Hours M-F 9a.m.-9p.m. — ATARI XE/XL/400/800 SOFTWARE — Extended Hours SAT 11a.m.-5p.m.

BRODERBUND
Karateka 19.95
Print Shop 28.95
Bank Street Writer 32.95
Graph. Lib. 1,2 or 3 16.95
P.S. Companion 22.95

MICROPROSE
Silent Service 22.95
F-15 Strike Eagle 22.95
Decision In Desert 25.95
Ken. Approach 16.95
Crusade/Europe 25.95
Conflict/Vietnam 25.95
Top Gunner 16.95

INFOCOM
Hitchiker 17.95
Wishbringer 11.95
Ballyhoo 25.95
Moonmist 22.95
Leather Goddess 22.95
Hollywood Hijinx 22.95
Zork Trilogy 39.95

XLENT SOFTWARE
Megafont II 16.95
Page Designer 18.95
Megafiler 18.95
PS Interface 18.95
Rubber Stamp 18.95
Typesetter 21.95
Picture Disk 14.95

ELECTRONIC ARTS
Age Of Adventure 10.95
Archon 10.95
Archon II 10.95
Fin. Cookbook 10.95
Mail Ord. Monsters 10.95
Movie Maker 10.95
Music Constr. 10.95
One On One 10.95
Pinball Constr. 10.95
Racing Destr. 10.95
7 Cities Of Gold 10.95

Super Bould. Dash 10.95
Touch. Football 10.95
Chessmaster 2000 25.95
Chickaumauga 23.95
Lords/Conquest 22.95
Ogre 25.95
Rommel/Toburk 26.95
Starfleet I 33.95
Ultima I 27.95
Ultima III 31.95
Ultima IV 38.95

MISCELLANEOUS XE / XL / 400 / 800
Flight Simulator 33.95
Universe 57.95
Strip Poker 21.95
Micro League Baseball 25.95
Harc/Brace Sat 59.95
General Mgr. MLB 25.95
Fight Night 18.95
Hardball 18.95
Triple Pak 14.95
Leader Board 24.95
Tourn. Disk/LB 14.95

Executive Disk/LB 14.95
Tenth Frame 27.95
Infiltrator 19.95
Bop 'N' Wrestle 19.95
Super Huey 16.95
Home Planetarium 27.95
Atariwriter Plus 39.95
MLB Boxscore/Stat 16.95
Home Accountant 30.95
Apshai Trilogy 16.95
Summer Games 16.95
World Karate Championship 19.95

Mercenary 18.95
Wizard's Crown 27.95
Gettsburg 39.95
Warship 39.95
Synfile 32.95
Syncalc 32.95
Music Studio 22.95
Alt. Reality 24.95
A/R Dungeon 24.95
Rebel Charge 32.95
Ace Of Aces 16.95
Battlecruiser 39.95

Extended Hours M-F 9a.m.-9p.m. CST — AMIGA SOFTWARE — Extended Hours SAT 11a.m.-5p.m.

AMIGA SOFTWARE
AC Basic 169.95
Aegis Draw 159.95
Aegis Animator 84.95
Allen Fires 24.95
Alt Reality 24.95
CBM Textcraft 69.95
CBM Pascal 69.95
CBM Amiga C 99.95
CBM Assembler 69.95
CBM Lisp 134.95
Amiga Enhancer 11.95
Amigaterm 34.95
Analyze 2.0 ..... 94.95
Archon 26.95
Archon II 26.95
Artic Fox 26.95
Arena 22.95
Art Gallery 1 Or 2 ..... 18.95

AMIGA SOFTWARE
Autoduel 32.95
Aztec C Develop 179.95
Aztec C Commer 299.95
Balance /Power 32.95
Bard's Tale 32.95
Best Accnting 299.95
Blk Cauldron 26.95
Bumpr Stick Mkr 36.95
Bureaucracy 25.95
Business Crd Mkr 36.95
Button Maker 36.95
Cambridge Lisp 129.95
2/2 Basketball 29.95
Champion Golf 25.95
Champ Baseball 29.95
Chessmaster 29.95
DB Man 89.95
Decimal Dungeon 31.95

AMIGA SOFTWARE
Deep Space 29.95
Defender/Crown 32.95
Deja Vu 32.95
Deluxe Paint II 84.95
Deluxe Print 64.95
Del Print Dat 1 ..... 20.95
Del Paint Dat 1 ..... 20.95
Del Paint Dat 2 ..... 20.95
Deluxe Video 64.95
Deluxe Music 2.0 ..... 64.95
Digipaint 44.95
Digiview 149.95
Diacovry Spell 24.95
Dacovry Trivia 24.95
Dacovry Math 24.95
Earl Weaver 32.95
First Shapes 31.95
Flight Sim 2 ..... 34.95

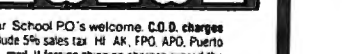
AMIGA SOFTWARE
Flip Side 31.95
Flow 64.95
Fortran 77 ..... 169.95
Fraction Action 31.95
Galileo 64.95
Gato 31.95
GFL Football 29.95
Goldspell 27.95
Grnd Slm Tennis 31.95
Graphicraft 34.95
Gridiron 44.95
Impact 54.95
Instant Music 32.95
Jet 34.95
Kampgruppe 39.95
Kid Talk 31.95
Kinderama 31.95
King Of Chicago 32.95

AMIGA SOFTWARE
KG Quest 1, 2 or 3 32.95
Leader Board 25.95
Logistix 89.95
LPD Filer 79.95
LPD Writer 79.95
LPD Planner 79.95
Marble Madness 32.95
Math Talk 31.95
Math Wizard 31.95
Mean 18 ..... 27.95
Microleague BB 39.95
Mindwalker 34.95
Module 2 ..... Call
Money Mentor 57.95
Music Studio 32.95
Ogre 32.95
On Line 42.95
One On One 26.95

AMIGA SOFTWARE
Organize 62.95
Page Setter 89.95
Par Real 79.95
Pawn 29.95
Phantasie 25.95
Police Quest 26.95
Power Windows 54.95
Printmaster + ... 31.95
Prowrite 79.95
Publisher 119.95
Roadwar 2000 ..... 25.95
S.D.I. 32.95
Scribble 62.95
7 Cities/Gold 26.95
Shanghai 25.95
Silent Service 25.95
Sinbad 34.95
Skyfox 26.95

AMIGA SOFTWARE
Sftwks Basic 64.95
Sonix 64.95
Space Quest 32.95
Starglider 29.95
Strip Poker 25.95
Superbase Pers 89.95
Tass Times 25.95
True Basic 99.95
TV Text 62.95
Typing Tutor 21.95
Ultima 3 or 4 ..... 36.95
VIP Professional 169.95
Winnie/Pooh 20.95
Winter Games 25.95
World Games 25.95
10th Frame 25.95
3D-Graphics 32.95

No surcharge for Mastercard or Visa



ORDERING INFORMATION:

Please **specify system.** For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School PO's welcome. **C.O.D. charges are $3.00.** In Continental U.S.A. include $3.00 for software orders. 4% shipping for hardware, minimum $4.00. MasterCard and Visa orders please include card #, expiration date and signature. WI residents please include 5% sales tax. HI, AK, FPO, APO, Puerto Rico and Canadian orders, please add 5% shipping, **minimum $5.00.** All other **foreign orders add 15% shipping, minimum $10.00.** All orders shipped outside the Continental U.S.A. are shipped first class insured U.S. mail. If foreign shipping charges exceed the minimum amount, you will be charged the additional amount to get your package to you quickly and safely. 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) 357-8181 to obtain an R.A. # or your return will not be accepted. Prices and availability subject to change without notice.

The Pierstorff Company, 131 W. Main St., Woodland, CA 95695
Circle Reader Service Number 208.

New Revision And Printer Driver For ST/MAC Emulator

Data Pacific has announced the fourth enhancement to its Magic Sac, which runs Macintosh software on the Atari ST, and a printer driver for Epson printers.

Revision 4.0 of the Magic Sac adds support for double-sided (800K) disk drives and limited color monitor support; it also adds GEM-based formatters and copiers, and compatibility with Apple's Finder 5.3/System 3.2 operating system (the latest revisions). It is available as an upgrade to registered owners for $10; suggested retail price is $149.95.

The Magic Printer Driver enables the Magic Sac to use Epson printers and compatibles like Citizen and Panasonic. It previously supported only the ImageWriter. Price is $45.

Data Pacific, 609 E. Speer Blvd., Denver, CO 80203
Circle Reader Service Number 209.

Electronic Card File For MS-DOS Machines

Tracker is an interactive card filing system that locates information quickly and easily via its extensive search and update functions. The program may be run in a memory-resident or non–memory resident mode; an autodial function is also included to provide instant access to online services through a modem.

Searches can be accomplished through any 8 of the 15 main working screens. Date-stamped notes may be appended to each file, as can up to 20 keywords that can be used later to include or exclude records from the four available reporting functions. For example, lists of people to contact may be produced onscreen with a couple of keystrokes as daily reminders.

Tracker runs on the IBM PC, AT, and compatibles, and retails for $99.

Adaptive (USA), 3701 Birch St., Newport Beach, CA 92660
Circle Reader Service Number 210.

Clip Art For *Flexidraw* And *Doodle!*

Inkwell Systems, manufacturer of *Flexidraw*—a high-resolution graphics program coupled with an industrial-quality light pen—recently released *The Graphics Galleria*, a collection of clip art and illustrations for use with the *Flexidraw* or *Doodle!* graphics programs. Each volume is a collection of clip art and illustrations based on one particular theme; the first four are *Borders & Signs, Clip Art Potpourri, Holiday Themes*, and *Maps of the World*. Each disk contains the *Flexidraw* format on one side and *Doodle!* on the other, and retails for $24.95. Additional disks are planned.

Inkwell Systems, P.O. Box 85152 MB290, 5710 Ruffin Rd., San Diego, CA 92138
Circle Reader Service Number 211.

Mouse Protector

H & H Enterprises has developed a product that offers protection as well as a new look for your computer's mouse. The MouseTop mouse cover fits most popular mouse devices, including those available for Apple, Commodore, IBM, Atari, and Tandy. Made from a silver/gray furlike fabric, the MouseTop comes in two different looks. One is slightly nearsighted and wears wire-rimmed glasses; it retails for $5.95. The other has 20/20 vision, and costs $5.49. This washable mouse cover protects the input device from the grime of daily use in addition to its aesthetic value.

H & H Enterprises, P.O. Box 2672, Corona, CA 91718
Circle Reader Service Number 212.

Popular ST Game Now Available For IBM PC

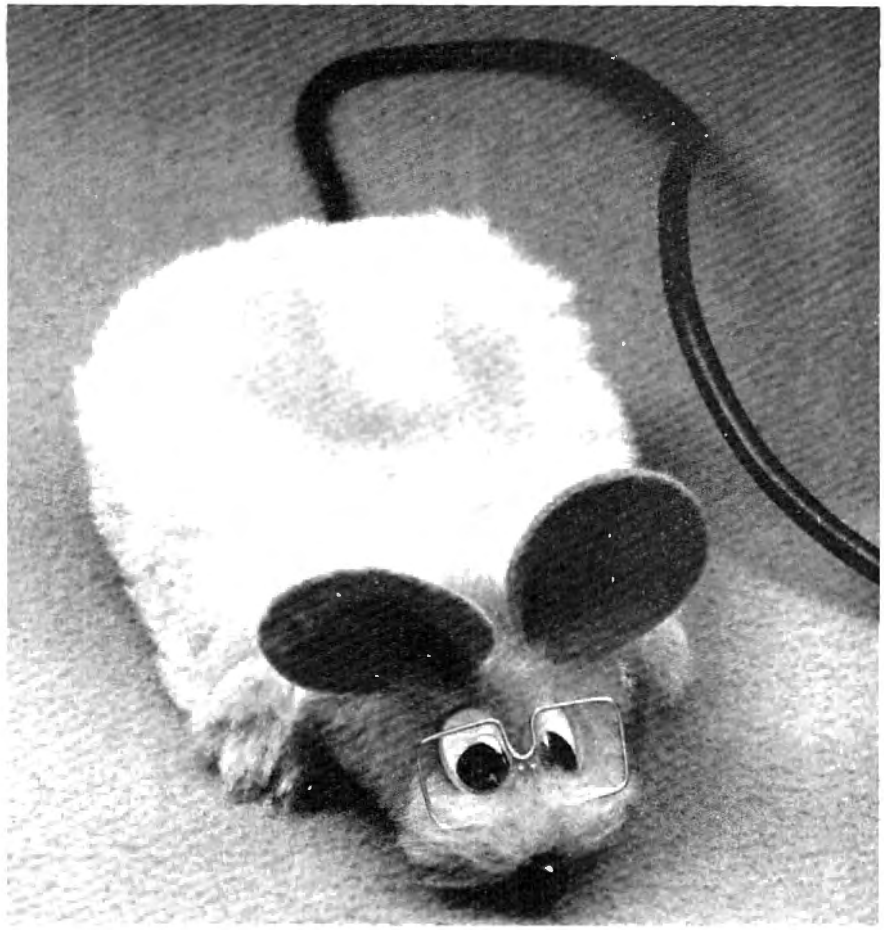
Starglider, an extremely popular game from Firebird Licensees, is now available for MS-DOS machines.

Starglider puts the player in command of an airborne ground attack vehicle which must ultimately do battle with the starship Starglider. Cunning, skill, and quick reflexes are all required to succeed in this deep-space conflict; 3-D animation and challenging maneuvers are arcade-quality. The program comes with a flight training manual, full-color poster, key guide, and a 64-page novella that sets the stage for the action. The MS-DOS (and upcoming Amiga and Apple II) versions retail for $44.95; a Commodore 64 version is also planned for $39.95.

Firebird Licensees, P.O. Box 49, Ramsey, NJ 07446
Circle Reader Service Number 213.

EA Game Available In New Format

Racing Destruction Set, a popular Electronic Arts game that lets two players compete using racetracks and vehicles of their own design is now available for the Atari 800/XL. The program features



Protect your computer's mouse with H & H Enterprise's mouse cover.

SOFTWARE DISCOUNTERS OF AMERICA

For Orders Only—1-800-225-7638
PA Orders—1-800-223-7784
Customer Service 412-361-5291

- Free shipping on orders over $100 in continental USA
- No Surcharge for VISA/MasterCard
- Your card is not charged until we ship

YOUR ENTERTAINMENT SOFTWARE SPECIALIST!

IBM SOFTWARE

ACCOLADE
Mean 18 Golf $30
Mean 18 Famous Course Disk ... $14

ACTIVISION
Hacker ... $9.88
Hacker 2 ... $24
Portal ... $27
Shanghai ... $23
Tass Times ... $23

ARTWORX
Bridge 4.0 ... $16
Strip Poker ... $25
Data Disk #1 Female ... $14
Data Disk #2 Male ... $14
Data Disk #3 Female ... $14

AVALON HILL
Super Bowl Sunday ... $21
S.B.S. Champs Disk ... $14
S.B.S. '85 Team Disk ... $14

BRODERBUND
Ancient Art of War ... $27
Carmen Sandiego (USA) ... $29
(World) ... $25
Karateka ... $21
Print Shop ... $36
P.S. Companion ... $32
P.S. Graphics Library #1 or #2 ... $21 Ea.
Toy Shop ... $32

CENTRAL POINT
Copy 2 ... $23
PC Tools ... $23

DATA EAST
Commando ... $24
Tag Team Wrestling ... $24

ELECTRONIC ARTS
Software Classic Series:
Archon ... $9.88
Mind Mirror ... $9.88
Music Const. Set ... $9.88
One-on-One ... $9.88
Pinball Const. Set ... $9.88
Seven Cities Gold ... $9.88
Super Boulder Dash ... $9.88

ELECTRONIC ARTS
Amnesia ... $29
Chessmaster 2000 ... $26
Grand Slam Bridge ... $39
Murder Party ... $26
Scrabble ... $26
Starfleet 1 ... $32
Starflight ... $32
World Tour Golf ... $32

EPYX
Create A Calendar ... $19
Pitstop 2 ... $9.88
Sub Battle Simulator ... $24
Summer Games 2 ... $24
World Games ... $24

GAMESTAR
Champ. Baseball ... $24
Champ. Basketball ... $24
Champ. Golf ... $24
GFL Ch. Football ... $24

HI-TECH EXPRESSIONS
Award Ware ... $9.88
Card Ware ... $6.88
Heart Ware ... $6.88
Party Ware ... $9.88

INFOCOM
Bureaucracy ... $24
Hitchhiker's Guide ... $19
Hollywood Hijinx ... $24
Leather Goddesses ... $24
Zork 1 ... $24
Zork Trilogy ... $44

MICROLEAGUE
Baseball ... $24
*Box Score/Stats Compiler ... $16
*General Mgr./Owner ... $19
*1986 Team Disk ... $12
Add-on programs!

MICROPROSE
F-15 Strike Eagle ... $21
Silent Service ... $21

MINDSCAPE
American Challenge ... $24
Balance of Power ... $30
Bop & Wrestle ... $19
Defender of the Crown ... $24
Infiltrator ... $19
Sub Mission ... $24

SSI
Battle of Antietam ... $30
Gettysburg ... $36
Kampfgruppe ... $36
Rings of Zilfin ... $24
Roadwar 2000 ... $24
Shard of Spring ... $24

SIERRA
Black Cauldron ... $24
King's Quest 1, 2, or 3 ... $32 Ea.
Space Quest ... $30

SPECTRUM HOLOBYTE
Gato ... $24
Orbiter ... $27
Wilderness ... $30

SPRINGBOARD
Certificate Maker ... $36
C.M. Library #1 ... $21
Early Games ... $24
Newsroom ... $36
Newsroom Pro ... $89
N.R. Clip Art #1 ... $19
N.R. Clip Art #2 ... $24
N.R. Clip Art #3 ... $19
P.S. Graphics Expander $24

SUBLOGIC
Football ... $26
Jet ... $32

TELARIUM
Amazon ... $9.88
Dragonworld ... $9.88
Fahrenheit 451 ... $9.88
Nine Princes ... $9.88
Perry Mason ... $9.88
Rendezvous ... $9.88

UNISON WORLD
Newsmaster ... $59
Printmaster Plus ... $36
Art Gallery 1 or 2 ... $19 Ea.

VALUE TIME
Art Library 1 or 2 ... $6.88 Ea.
Calendars & Stationery $6.88
Greeting Cards ... $6.88
Signs & Banners ... $6.88

COMMODORE 64 & 128

ACCESS
Tenth Frame ... $25
World Class Leader Board ... $25

ACCOLADE
Ace of Aces ... $19
Comics ... $19
Fight Night ... $19
Hardball ... $19
Killed Until Dead ... $19

ACTION SOFT
Up Periscope! ... $19

ACTIVISION
Aliens ... $23
Gamemaker ... $25
Gamemaker Library Disks Sports or Sci. Fi. ... $14 Ea.
Hacker ... $9.88
Hacker 2 ... $23
Music Studio ... $23
Portal ... $25
Shanghai ... $19
Transformers ... $19

ARTWORX
Bridge 4.0 ... $16
Highland Games ... $9.88
Police Cadet ... $9.88
Strip Poker ... $21
Data Disk #1 Female ... $14
Data Disk #2 Male ... $14
Data Disk #3 Female ... $14
Thai Boxing ... $9.88

AVALON HILL
Super Sunday ... $21
S.B.S. 1985 Team Disk ... $14
S.B.S. Champs Disk ... $14

AVANTAGE
Spy vs. Spy 1 & 2 ... $9.88

BERKELEY SOFTWORKS
GEOS 64 ... $39
GEOS add-ons ... Call

BRODERBUND
Print Shop ... $26
P.S. Graphics Library #1, #2 or #3 ... $16 Ea.
Toy Shop ... $26

DATA EAST
Breakthru ... $23
Commando ... $23
Express Raider ... $23
Ikari Warriors ... $23
Kung Fu Masters ... $14
Tag Team Wrestling ... $23

ELECTRONIC ARTS
Software Classic Series:
Archon 2: Adept ... $9.88
Heart of Africa ... $9.88
Mail Order Monsters $9.88
Movie Maker ... $9.88
Music Const. Set ... $9.88
One-on-One ... $9.88
Pinball Const. Set ... $9.88
Seven Cities Gold ... $9.88
Skyfox ... $9.88
Super Boulder Dash ... $9.88
Touchdown Football ... $9.88
Ultimate Wizard ... $9.88

ELECTRONIC ARTS
America's Cup Sailing $23
Arctic Fox ... $23
Bard's Tale 1 or 2 ... $26 Ea.
Chessmaster 2000 ... $26
Marble Madness ... $23
Pegasus ... $23
Star Fleet 1 ... $26
World Tour Golf ... $23

EPYX
Champ. Wrestling ... $23
Create A Calendar ... $19
Destroyer ... $23
Fast Load (cart) ... $23
Pitstop 2 ... $9.88
Sub Battle Simulator ... $25
Summer Games 2 ... $23
Super Cycle ... $23
World Games ... $23

FIREBIRD
Elite ... $21
Starglider ... $25
The Pawn ... $25

GAMESTAR
Champ. Baseball ... $23
Champ. Basketball ... $23
GFL Champ. Football ... $23

INFOCOM
Bureaucracy 128 ... $23
Hitchhiker's Guide ... $19
Hollywood Hijinx ... $23
Leather Goddesses ... $23
Zork Trilogy ... $39

MASTERTRONIC
Bounder ... $6.88
Captain Zap ... $6.88
Knight Games ... $6.88
Ninja ... $6.88
Pro Golf ... $6.88
Shogun ... $6.88
Vegas Poker ... $6.88

MICROPROSE
Acrojet ... $16
Gunship ... $23
Silent Service ... $23

MINDSCAPE
Bop & Wrestle ... $19
Defender of the Crown $25
Fist ... $19
Super Star Ice Hockey ... $23
Uchi Mata ... $19

ORIGIN
Autoduel ... $32
Moebius ... $26
Ultima 4 ... $39

SPRINGBOARD
Certificate Maker ... $32
C.M. Library #1 ... $23
Newsroom ... $32
N.R. Clip Art #1 or #3 ... $19
N.R. Clip Art #2 ... $25

SSI
Gettysburg ... $37
Phantasie 2 ... $25
Rings of Zilfin ... $25
Shard of Spring ... $25
Wizard's Crown ... $25

SUBLOGIC
Baseball ... $32
Flight Simulator 2 ... $32
F.S. Scenery Disks ... $14 Ea.
Football ... $26
Jet ... $26

APPLE II SERIES

ACCESS
Triple Pack: BH1, BH2, Raid Over Moscow ... $14
World Class Leader Board ... $25

ACCOLADE
Fight Night ... $19
Hardball ... $19

ACTIVISION
Gamemaker Library Disks:
Sci-Fi ... $16
Sports ... $16
Gamemaker ... $30
Ghostbusters ... $9.88
Hacker 2 ... $24
Music Studio GS ... $49
Paintworks GS ... $47
Portal ... $27
Shanghai ... $21
Shanghai GS ... $29
Tass Times GS ... $24

ARTWORX
Bridge 4.0 ... $16
International Hockey ... $19
Strip Poker ... $19
Data Disk #1 (Female) ... $14
Data Disk #2 (Male) ... $14

AVALON HILL
Dr. Ruth ... $19
Super Sunday ... $21

BANTAM
Walt Disney Series:
Card & Party Shop ... $24
Comic Strip Maker ... $24

BAUDVILLE
Award Maker ... $24
Video Vegas ... $19

BRODERBUND
Animate (128K) ... $42
Carmen San Diego USA ... $27
World ... $24
Dazzle Draw (128K) ... $34
Fantavision ... $30
Karateka ... $21
Print Shop (Enhanced) ... $27
Print Shop Companion $23
P.S. Graphics Library #1, 2 or 3 ... $14 Ea.
Toy Shop ... $32

CENTRAL POINT
Copy 2 ... $21

DATA EAST
Commando ... $21
Tag Team Wrestling ... $21

ELECTRONIC ARTS
Age of Adventure ... $9.88
Amnesia ... $29
Archon ... $9.88
Archon 2: Adept ... $9.88
Arctic Fox ... $26
Bard's Tale 2 ... $32
Battlefront ... $25
Carriers at War ... $30
Chessmaster 2000 ... $29
Deluxe Paint II GS ... $65
D. Paint Art #1 GS ... $19
D. Paint Art #2 GS ... $19
Europe Ablaze ... $30
Lords of Conquest ... $26
Murder Party ... $26
Marble Madness ... $23
Mind Mirror ... $9.88
Moebius ... $36
Movie Maker ... $9.88
Music Const. Set ... $9.88
Music Const. Set GS ... $32
Ogre ... $25
One-on-One ... $9.88
Pegasus ... $23
Pinball Const. Set ... $9.88
Russia (The Great War) $26
Scrabble ... $26
Seven Cities of Gold $9.88
Skyfox ... $9.88
Star Fleet 1 ... $32
Super Boulder Dash $9.88
Ultima 4 ... $36

EPYX
Champ. Wrestling ... $23
Create A Calendar ... $19
Summer Games 2 ... $23
Winter Games ... $23

GAMESTAR
Champ. Baseball ... $24
Champ. Basketball ... $24
Champ. Football ... $24

HI-TECH EXPRESSIONS
Award Ware ... $9.88
Card Ware ... $6.88
Heart Ware ... $6.88
Party Ware ... $9.88

INFOCOM
Bureaucracy ... $24
Hitchhiker's Guide ... $19
Hollywood Hijinx ... $24
Leather Goddesses ... $24
Zork Trilogy ... $44

MICROLEAGUE
Baseball ... $24
Box Score Stats ... $14
General Mgr./Owner $19
1986 Team Disk ... $12

MICROPROSE
F-15 Strike Eagle ... $21
Silent Service ... $21

MINDSCAPE
American Challenge ... $24
Bop & Wrestle ... $19
Infiltrator ... $19

SSI
Gettysburg ... $35
Phantasie 1, 2 or 3 ... $23 Ea.
Realms of Darkness ... $23
Rings of Zilfin ... $23
Roadwar 2000 ... $23
Shard of Spring ... $23
Wizard's Crown ... $23

SPRINGBOARD
Certificate Maker ... $29
C.M. Library Vol. 1 ... $21
Newsroom ... $32
N.R. Clip Art #1 or #3 ... $17
N.R. Clip Art #2 ... $23
Publisher ... Call

SUBLOGIC
Baseball ... $32
Flight Simulator 2 ... $30
Jet ... $26

VIP TECHNOLOGIES
VIP Professional GS ... $179

P.O. BOX 111327—DEPT. CP—BLAWNOX, PA 15238

*Please Read The Following Ordering Terms & Conditions Carefully Before Placing Your Order: Orders with cashiers check or money order shipped immediately on in stock items. Personal & Company checks, allow 3 weeks clearance. No C.O.D.'s! Shipping: Continental U.S.A.—Orders under $100 add $3; free shipping on orders over $100. AK, HI, FPO, APO—add $5 on all orders. Canada & Puerto Rico—add $10 on all orders. Sorry, no other International orders accepted! PA residents add 6% sales tax on the total amount of order including shipping charges! REASONS FOR CALLING CUSTOMER SERVICE—412-361-5291 (1) Status of order or back order (2) If any merchandise purchased within 60 days from S.D. of A. is defective, please call for a return authorization number. Defective merchandise will be replaced with the same merchandise only. Other returns subject to a 15% restocking charge. After 60 days please refer to the manufacturers warranty included with the merchandise & return directly to the manufacturer. Customer service will not accept collect calls or calls on S.D. OF A.'s 800# order lines! Have you seen our on line catalog of 1000 software titles for Commodore, Atari, Apple, IBM and Amiga! It's on Compuserve's Electronic Mall—just type GO SDA and shopping for software will never be the same again! HOURS: Mon.-Fri. 9AM-5:30 PM, Sat. 10AM-4PM Eastern Time. Because this ad had to be written 2-3 mos. before it was published, prices & availability are subject to change! When sending a mail order, please specify make & model# of your computer!

a unique "isolated camera" display with true split-screen scrolling so that each player can view his or her own car and position on the track no matter where the opponent is. The game also comes with predesigned tracks representing some of the most famous racetracks in the world.

The 800/XL version retails for $32.95.

Electronic Arts, 1820 Gateway Dr., San Mateo, CA 94404

Circle Reader Service Number 214.

Inexpensive Productivity Software

Easy Working Software, the newest division of Spinnaker Software, has announced the release of three inexpensive new productivity programs: *The Writer*, *The Planner*, and *The Filer*.

The Writer is designed for the first-time word processor user. Standard editing functions are included, like cut-and-paste, automatic page numbering, search and replace, and underlining. The spellchecking feature recognizes over 99 percent of spelling and typing errors.

The Filer functions as a database manager to store and organize records more efficiently. It can be used for a variety of applications, like creating and storing mailing lists, club membership files, and inventories. Over 50,000 records can be stored per file; form letters can be generated by integrating *The Filer* and *The Writer*.

The Planner is an electronic spreadsheet, easy enough for an inexperienced spreadsheet user to plan home budgets, as well as tax and other financial statements. It features extensive math calculations, standard editing functions, onscreen help, and allows easy manipulation of existing data.

All three programs can be fully integrated and require little prior computer knowledge. Each is available for IBM PC and compatibles, Apple II series, and Commodore 64 computers for $9.95.

Spinnaker Software, One Kendall Sq., Cambridge, MA 02139

Circle Reader Service Number 215.

World War II Conflict Game Available From Avalon Hill

Named after the German general who commanded on the WWII Eastern front and designed the Blitzkrieg tactics that revolutionized armor tactics, *Guderian* simulates a critical campaign of the Second World War using Avalon Hill's joystick-driven system from *Gulf Strike*.

Driving towards Moscow, the German army encounters resistance around the city of Smolensk. As the German commander, you must use the Blitzkrieg tactics of encirclement and deep penetration behind enemy lines to trap and destroy the enemy. The Soviet commander, meanwhile, must organize an effective resistance and organize local counterattacks to halt the German forces. Time becomes a critical factor as the German units must cross the map within 12 turns or lose the game.

Guderian has a solitaire option which allows the player to take command of either the German or Soviet armies, as well as to control overrun attacks, supply rules, Soviet leaders, rail movement, and optional reinforcements. The rulebook introduces new players to historical gaming, explaining concepts like zones of control, supply, combat, and movement.

Available for Atari, Commodore, and Apple eight-bit machines, *Guderian* retails for $30.

The Avalon Hill Game Company, 4517 Harford Rd., Baltimore, MD 21214

Circle Reader Service Number 216.

Miniature Golf Program For IBM and ST

Artwork Software has begun shipping *Minigolf* for IBM PC and compatibles and Atari ST computers.

Minigolf is a miniature golf game that comes with three different courses. (The ST version provides an editor to create or modify your own courses.) Each course confronts you with various obstacles to decrease the chances of making a hole-in-one. The program's scorecard displays the low score for each course, and tracks up to four players. *Minigolf* courses are designed to conform to professional and tournament-class standards. Suggested retail price is $29.95 for IBM PC and compatibles and $19.95 for the Atari ST version.

Artworx Software, 1844 Penfield Rd., Penfield, NY 14526

Circle Reader Service Number 217.

File Command Window For IBM PC

MicroMath is announcing *DirectoryWindow*, a memory-resident directory utility that can be popped up at any time to provide access to various file commands without disturbing the computer's current activity. It allows users to scroll through file directories; sort directories by date, by size, or alphabetically; compare directories by popping up two or three simultaneously; rename files by typing over the filename on the screen display; delete files; and print directories for archival purposes.

In addition to accepting the full array of file specifications that the DIR command accepts, *DirectoryWindow* allows the display to be redistricted to subdirectories within a given directory. Directory trees may also be traversed. *DirectoryWindow* displays the number of bytes occupied by the specified files, as well as space remaining on the disk. The program, available for MS-DOS machines, costs $14.95 plus $2 shipping and handling.

MicroMath Scientific Software, 3690 E. Fort Union Blvd., Ste. 204, Salt Lake City, UT 84121-4550

Circle Reader Service Number 218.

Interface Support For Eight-Bit Ataris

Two new products from ICD offer Atari eight-bit computer users new interface options.

The P:R:Connection plugs directly into the serial port of any eight-bit Atari and provides the user with a standard Centronics printer port and two RS-232 type serial ports. It draws energy from the computer itself, which means one less cord needing an outlet. Its serial ports resemble those of the Atari 850 interface, possessing the same signals and functions and using a fully compatible built-in R: handler. The P:R:Connection retails for $89.95.

The Multi I/O offers five functions in one box for your Atari 130XE or 800XL: a 256K or one-megabyte RAM disk; centronics parallel interface; serial printer/modem interface; print spooler; and hard disk interface. The 256K version retails for $199.95; the one-version, for $349.95. A 130XE Adapter (adding two cartridge ports) is also available for $19.95.

ICD, 1220 Rock St., Rockford, IL 61101-1437

Circle Reader Service Number 219.

New Baseball Strategy Game For Commodore 64

Monday Morning Manager puts 64 major league baseball teams with full rosters (over 1500 players) at your command. It's a statistic-based baseball strategy game whose master disk contains real information on the greatest teams from 1905 through 1985 and four All Star teams.

Nine main menu selections let you create your own teams; make trades; draft teams; keep personal, current statistics on Little League, softball, or major league teams; and save and print stats, game score cards, rosters, and won-lost records. The program stores up to 65,000 at-bats per player and over 20,000 innings pitched per pitcher. Designed for

THE LOWEST PRICES | THE BEST SERVICE

ELECTRONIC ONE*

PHONE LINES OPEN 10-6 E.S.T. M-F

CALL (614) 864-9994 • P.O. Box 13428 • COLUMBUS, OHIO 43213

ATARI 1040 ST
Personal Computer System

- 1,048,576 BYTES RAM
- 196,608 BYTES ROM
- BUILT IN TOS
- 3½ INCH DBL. SIDED DISK DRIVE
- TWO BUTTON MOUSE
- COLOR MONITOR
- 512 COLORS
- MIDI IN/OUT PORTS

$869.99 COMPLETE COLOR SYSTEM

B/W SYSTEM $689.99

PC COMPATIBLE

COMMODORE PC-10-2
IBM Compatible System

- 8088 MICROPROCESSOR
- 640K B RAM
- 2-360K DISK DRIVE
- PC-XT COMPATIBLE BIOS
- 85 KEYS
- CENTRONICS PARALLEL PORT
- RS-232 SERIAL PORT
- EXPANSION SLOTS
- POWER SUPPLY W/FAN
- FULLY IBM COMPATIBLE
- TENIKA MJ22 COLOR RGB W/CABLE 239.99 OPTIONAL PRICE

$889.99

PC10-1 ONE DISK DRIVE 512K **$719.99**

ATARI 520ST

520ST SINGLE DISK DRIVE B/W Monitor Complete	520ST SINGLE DISK DRIVE Color Monitor Complete
459.99	**699.99**

- SF354 SGL DISK DRIVE 129.99
- SF 350-20 MEG HARD DRIVE 499.99
- SF 314 DBL DISK DRIVE 199.99
- COLOR MONITOR 299.99
- B/W MONITOR 129.99

COMPUTERS FOR THE HOME AND OFFICE AT LOW, LOW PRICES ELECTRONIC ONE

PC COMPATIBLE

IBM Compatible

Franklin Computers
PC 8000

- 4.77 MHZ
- 512K RAM
- 2-360K DISK DRIVES
- PARALLEL PRINTER PORT
- COLOR ADAPTER
- MODEM ADAPTER
- EXPANSION SLOTS

$599.99

W/RGB MONITOR 819.99 | W/GREEN MONITOR 689.99

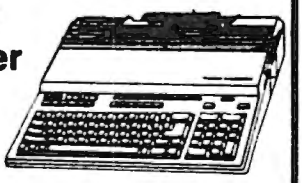
LOW, LOW PRICE HOME COMPUTERS

ATARI 800XL (64k)	$69.99
ATARI 65XE (64k) W/SOFTWARE	$89.99
ATARI 130XE (128k)	$129.99
COMMODORE 64 (REFURBISH)	$129.99
FULL 90 DAY WARRANTY	
COMMODORE 64C (W/GEOS)	$169.99
COMMODORE 128 (128k)	$239.99

(WE CARRY VIDEO GAMES TOO!)

APPLE 2C COMPATIBLE

Franklin Ace 500
Apple 2C Computer



- FULLY 2C COMPATIBLE
- 256K RAM
- 5¼" 164K DRIVE
- KEY PAD
- SWITCHABLE 40/80 COLUMN
- EARPHONE JACK
- 90 KEY KEYBOARD

$399.99

W/GREEN MONITOR $489.99 | W/RGB COLOR $619.99

CALL FOR COMPLETE CATALOG

HOW TO ORDER: CASHIER CHECK, MONEY ORDER, MASTERCARD* OR VISA* (ADD 4% FOR CHARGE CARDS) . . . NO PERSONAL CHECKS . . . NO C.O.D.'S . . . SHIPPED U.P.S. . . . ALL PRICES SUBJECT TO CHANGE WITHOUT NOTICE.
SHIPPING: ADD $3.00 ON ALL ORDERS UNDER $100.00 . . . ADD $5.00 ON ALL ORDERS OVER $100.00. ACTUAL FREIGHT CHARGED ON MULTIPLE ORDERS.
INTERNATIONAL: ACTUAL FREIGHT CHARGED ON ALL ORDERS OUTSIDE THE CONTINENTAL UNITED STATES INCLUDING A.P.O.
POLICIES: NO RETURNS WITHOUT A RETURN AUTHORIZATION . . . NO RETURNS UNLESS DEFECTIVE. ALL DEFECTIVES WILL BE EXCHANGED . . . NO EXCEPTIONS. PLEASE SPECIFY . . .

CALL OR WRITE FOR FREE CATALOG

CALL ELECTRONIC ONE (614) 864-9994 P.O. BOX 13428 COLUMBUS, OHIO 43213

the Atari eight-bit machines, *Monday Morning Manager* is now available for the Commodore 64 for $39.95.

TK Computer Products also offers a line of inexpensive software called White Bag Software for the 64: *Money $pin*, a puzzle-solving program, and *Crypto-Mania*, an educational word game. Each retails for $14.95.

TK Computer Products, P.O. Box 9617, Downers Grove, IL 60515

Circle Reader Service Number 220.

Recording Studio For The ST

Midisoft Studio lets you edit, compose, and record music using an Atari ST and any instrument that has a MIDI connection. The program's features include real-time record, playback, overdub, rewind, and fast-forward; 32 polyphonic independently-controlled tracks; 30,000 notes per song; full-track editing and flexible region editing; and many other features standard in such a program. In addition, it offers external sequencer control to recognize and send codes to other equipment, and instrument set-up for control over special options on synthesizers.

The price of *Midisoft Studio* is $99; a special studio demo disk is available for $10.

Midisoft, P.O. Box 1000, Bellevue, WA 98009

Circle Reader Service Number 221.

SDI Simulation For The 64

Now you can play "star wars" as if your computer were actually linked to an existing SDI (Strategic Defense Initiative) system with *DEF CON 5* from Cosmi. With this interstellar defense simulation, you control 16 orbiting visual reconnaissance satellites that provide 23 different geostationary and close-up maps of the earth's surface. Using a series of command and operator control functions, you bring your SDI system into action against incoming enemy ballistic missiles. Weapons such as ground-based chemical lasers, orbiting laser reflectors, free-electron lasers, neutral particle beams, electro-magnetic launchers, nuclear-pulsed x-ray lasers, and antispace mine robots are available to intercept and destroy the enemy warheads.

Retail price is $19.95.

COSMI, 415 N. Figueroa St., Wilmington, CA 90744

Circle Reader Service Number 222.

Electronic Arts To Distribute Sierra On-Line Software

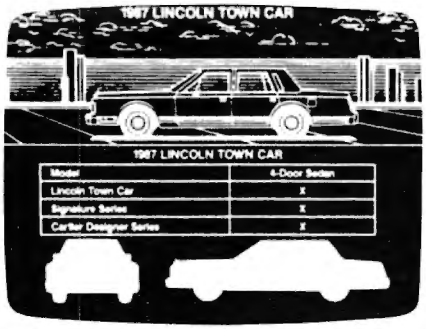
Electronic Arts has announced an agreement to distribute the first five Sierra On-Line Amiga products under a new affiliated label agreement. *King's Quest* and its sequels, *Romancing the Throne* and *To Heir Is Human*, make up a trilogy of 3-D animated fantasy adventure games; each is $49.95. *Space Quest*, a science fiction parody game, also carries a suggested retail price of $49.95; and *Winnie the Pooh in the Hundred Acre Wood* is a children's entertainment/educational product with a price of $24.95.

Electronic Arts, 1820 Gateway Dr., San Mateo, CA 94404

Circle Reader Service Number 223.

Lease Or Buy?

Lincoln-Mercury is offering a free program that explains the advantages of leasing new Lincoln-Mercury and Merkur XR4Ti model cars. The program also features color graphics illustrations of all Lincoln-Mercury car models.



Lincoln-Mercury offers a free disk that explores vehicle leasing options.

The disk is available free of charge for the IBM PC and compatibles, Macintosh, and Commdore 64/128.

Lease Program Headquarters, P.O. Box 2909, Farmington Hills, MI 48018-2909

Circle Reader Service Number 224.

WWII Wargame Simulation For The Amiga

Strategic Simulations has released an Amiga version of *Kampfgruppe*, an advanced-level war simulation. This game explores the tactical opportunities between the German and Soviet units on the Eastern Front during WWII. You can create your own scenarios or follow the four historic models. There are over 45 armored vehicles, plus guns and infantrymen for you to command.

Suggested retail price is $59.95.

Strategic Simulations, 1046 N. Rengstorff Ave., Mountain View, CA 94043

Circle Reader Service Number 225.

Educational Ants

Nine busy ants in Sunburst's *Ant Farm* help kids in grades 3 and up learn to become better problem solvers. The object of this game is to find the correct workstations for the ants based on each ant's work pattern. To do so, students must gather information about the ants and place them in the farm accordingly. There are different levels of play and challenges plus an onscreen toolbox for assistance.

Ant Farm is available for Apple II–series computers. Each package includes a disk, backup, and teacher's guide.

Retail price for *Ant Farm* is $59.

Sunburst Communications, 39 Washington Ave., Pleasantville, NY 10570

Circle Reader Service Number 226.

Amiga Astronomy Program

Infinity Software has released *Galileo*, a full-featured desktop astronomy program for the Amiga. This program uses a database of over 1600 stars and over 400 deep-sky objects. Users can scroll to any part of the sky to identify constellations, planets, and a solar eclipse. *Galileo* shows the sky from any point on Earth for any date in this century and shows each star in nine different levels of brightness.

Galileo retails for $99.95.

Infinity Software, 1331 61st St., Ste. F, Emeryville, CA 94608

Circle Reader Service Number 227.

Math Tutorial For Apple

Scholastic Software has introduced *Math Tutor*, a comprehensive series of four math tutorials for the Apple II–series of computers (48K RAM minimum). Designed for students in grades 1–6, the series includes programs in addition, subtraction, multiplication, and division. Each gives students an introduction to the arithmetic operation or concept and reinforces learning with practice problems.

With *Math Tutor*, students learn in a logical step-by-step sequence. First, they take a placement test that evaluates their strengths and weaknesses and places them at one of seven levels. They receive instruction and repeated practice at that level, and then move to the next. Student progress is continually evaluated to provide the learner with additional feedback and instruction as needed.

Math Tutor also offers teachers and parents a management system to track and monitor students' progress. It can print out records of each student's performance or can be turned off to allow for home use or specific teacher needs. Each package contains worksheets that correspond to each lesson on the curriculum disks.

All four *Math Tutor* packages are available in both school and home editions. The home editions, retailing for $59.95 each, contain two lesson disks, a management disk, and a user handbook

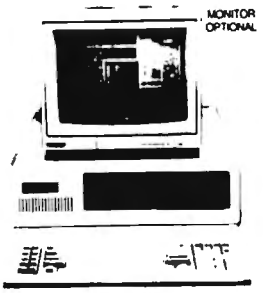
ATARI® ST PC COMPATIBLES

STation
The one STop shop

ORDERS ONLY
CALL TOLL FREE
1-800-225-5878
ORDER STATUS, INFORMATION, R.A.'s
1-313-595-0788

PC-XT SYSTEMS $399

- IBM* COMPATIBLE
- 360 K FLOPPY DRIVE
- FLOPPY DRIVE CARD
- 256k MEMORY
- XT STYLE KEYBOARD
- 8 EXPANSION SLOTS
- 150 WATT P.S.
- RGB/COMPOSITE MONITOR OUTPUTS
- FCC APPROVED



PC-AT SYSTEMS $995

- PC-AT COMPATIBLE
- RGB/COMPOSITE OUTPUT
- 1.2 MEG DRIVE
- 200 WATT P.S.
- AT STYLE KEYBOARD
- FLOPPY/HARD CONTROLLER
- 512k RAM
- FCC APPROVED
- 6/8 MHZ

*IBM IS A REGISTERED TRADEMARK OF INTERNATIONAL BUSINESS MACHINES

PC Additional Equipment

• PARALLEL PRINTER CARD	$24
• RS-232 INTERFACE	29
• CLOCK/CALENDAR	36
• COLOR VIDEO CARD	69
• TTL VIDEO CARD	85
• MULTI I/O CARD	79
• 360K H-H DRIVE	94

BULK DISKS

LIFETIME WARRANTY!

5¼ $.29 SS/DD QTY. 1000 — $.31 DS/DD QTY. 100

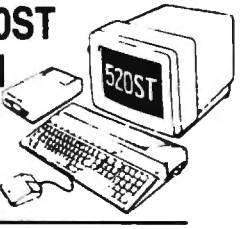
3½ $.99 SS/DD QTY. 25 — $1.29 SS/DD QTY. 25

520ST SYSTEMS 1040ST

$659 COLOR **$call**

$469 MONO **$call**

OR CALL FOR LATEST PRICES



SUPER SPECIAL SEAGATE PC-XT HARD DRIVE KIT

$389

20 MEG

PANASONIC 1080

★ 120 CPS
★ EPSON IBM
★ NLQ MODE
★ FRICTION TRACTOR

$199 2 YEAR WARRANTY!

DATA CASE SPECIAL

SD/DS/DD DISKETTES PACKED IN A FLIP TOP DATA CASE WITH SLEEVES, LABELS, & WRITE PROTECTS

All Just $24.95 5¼"

1200 BAUD MODEM KIT FOR ST

★ AVATEX 1200 HC
★ ST TALK SOFTWARE
★ COMPUSERVE STARTER KIT
★ RS-232 CABLE M-F

COMPLETE KIT *JUST* **$139**

ATARI SH 204 OR SUPRA DRIVE 20 MEG HARD DRIVE

YOUR CHOICE SAME LOW PRICE

$549

CABLES, INTERFACES, ETC

AB Switch Box Parallel	39 95
AB Switch Box RS-232	39 95
AB-X Crossover Box Par	49 95
IBM/ST Printer Cable 10'	9 95
IBM/ST Printer Cable 6'	6 95
Microstuffer	59 95
P R Connection	79 95
RS-232 F/F 6'	9.95
RS-232 M/F 10'	12.95
RS-232 M/F 6'	8.95
RS-232 M/M 6'	9.95
ST Drive Cable 6'	14.95
ST Serial 6'	9 95

CONTINUOUS CARD STOCK

3 X 5	4 99
4 X 6	7.99
5 X 7	9 99

CONTROLLERS

Atari Economy	4 95
Epyx 500XJ	14 95
Replacement Mouse (ST)	49 95
Tac 2	8 95
TAC-10 IBM & APPLE	27 95
Wico 3-Way	23 95
Wico Boss	12.95

DISKETTES

3½" Drive Cleaner	11 95
Bulk 3½" DS/DD	1 49
Bulk 3½" SS/DD	99
Bulk 5¼" DS/DD	39
Maxell 3½" DS/DD (Box)	19 95
Maxell 3½" SS/DD (Box)	14 95
Sony 3½" DS/DD (Box)	19.95
Sony 3½" SS/DD (Box)	13.95
Tyvek Sleeves	05

DUST COVERS

1040 ST	9 95
520ST	6 95
520ST System	24 95
Gemini 10	6 95
IBM PC Keyboard	6 95
IBM PC System	23.95
Panasonic 1080/90/91	6 95
SF 314	5 95
SF 354	5.95
SM 1224	11.95
SM 124	11 95

HARDWARE

1200 Baud Modem Internal	119 00
20 Megabyte Seagate w/controller	389 00
Amber TTL Monitor	119 00
Atari 1040ST	Call
Atari 314 Drive	199 00
Atari 354 Drive	105 00
Atari 520ST	259 00
Atari SC1224 Color Monitor	299 00
Atari SH204 Hard Drive	539 00
Atari SM124 Mono Monitor	119 00
Atari SMM804 Printer	169 00
Atari SX212 Modem	79 00
Clock Calendar Card DS/DD ½ Height	36 00
360k Drive	99 00
Floppy Drive Controller	39 00
Game Controller Port	29 95
Green Composite Monitor	79 00
Keyboard 5151	85 00
Keyboard 5160	85 00
Multi I/O + II	79.00
Okimate 20 w/Plug-n-Print	198.00
Panasonic 1080i (120 CPS)	195 00
Panasonic 1091i (180 CPS)	249 00
Panasonic 1592	429 00
Parallel Port	24 00
RS-232 Port	29 00
Star NP-10	158 00
Star NX-10	229 00
Supra 20 Meg Hard Drive	549 00
Upgrade, EZ-Ram	159 00
Upgrade, Meg-A-Ram	129 00
Video Card - Mono TTL	85.00
Video Card - RGB/Composite	85 00
XT Turbo Upgrade 4 77/8 MHZ	20 00

COLOR PAPER

BLUE, YELLOW, GREEN, PINK WHITE, SALMON & GOLD
500 Sheets (Any Color)

20# LZ cut	6 95
Rainbow Pack #1 (P-Y-B)	6 95
Rainbow Pack #2 (GR-S-GO)	6 95

RIBBONS

Epson FX/MX RX-80	5 99
IBM Proprinter	9 95
sonic 1080/90/(1	9 99
Star/Okidata	2.99
XMM 801 Legend/BMC	7 99
XMM 804	7 99

CONTINOUS LABELS

Audio Cassette, White	9 99
Disk (2 ⅜ x 2) 250 White	4 99
File Folder (3½ X 5/8) 750	9 99
Mail (3½ X 15/16) 1000	6 99

MISC. ACCESSORIES

14"Twist-N-Tilt	14 95
5 ¼ Disk Notcher	7 95
Amaray Printer Stand	14 95
Mouse House	4 95
Mouse Mat	7 95
Printer Stand 2-Piece	12 95
Six Outlet Surge Strip	14 95

DISK STORAGE

Disk Directory (3 1/2)	22 99
Disk File 40 - 3½	6 99
Disk File 60 - 5¼	6 99
Library Case - 10 (3½)	1 99
Library Case-3 (5¼)	99
Library Case-5 (3½)	99
Pocket Pack (3½)	14 99

ATARI ST SOFTWARE

Alternate Reality	32 95
Animal Kingdom	26 95
Arctic Fox	27 95
Black Cauldron	26 95
Chessmaster 2000	30 95
Crystal Castles	19.95
Dac Easy Accounting	46 95
Data Manager ST	52 95
DB Man	119 95
DB Master One	39 95
Degas Elite	53 95
Dollars & Sense	66 95
Donald Duck's Playground	19 95
Easy Draw	53 95
First Cadd	32 95
First Shapes	32.95
First Word	34 95
Flight Simulator II	34.95
Fraction Action	26.95
GFA Basic	52.95
GFA Basic Compiler	52.95
High Roller	32 95
I S Talk	34.95
Isgur Portfolio	132 95
Joust	18.95
K-Switch	25.95
Kinderama	26 95
Kings Quest I, II, or III	32 95
Lattice C	99.95
Leader Board	26.95
Mark Williams C	119.95
Math Talk	32.95
Music Studio	39 95
Phantasie II	26 95
Phantasie	26 95
Print Master Plus	26 95
Psion Chess	39 95
Publishing Partner	99 95
S D.I	32 95
Silent Service	26.95
Skyfox	31.95
Space Quest	32 95
Speller Bee	32 95
Swiftcalc ST	52 95
Tenth Frame	26 95
Time Bandits	26 95
VIP Professional	166 95
Word Writer ST	52.95
Z-Time 1040	49.95
Z-Time 520	39.95

IBM SOFTWARE

Alternate Reality the City	26 95
Archon	10 95
Balance of Power	33 95
Black Cauldron	26 95
Championship Lode Runner	23.95
Chessmaster 2000	27.95
Clip Art Vol 1 - Newsroom	19 95
Clip Art Vol. 2 - Newsroom	26.95
Clip Art Vol 3 - Newsroom	19.95
Crosstalk	129.95
Dbase II	333.95
Dbase III Plus	466.95
Dollars and Sense with Forecast	120.95
Donald Ducks Playground	19 95
F-15 Strike Eagle	23 95
Flight Simulator	33 95
Gato	26.95
Graphics Library I or II	23.95
I S Talk	39.95
Jet	33.95
Karateka	23.95
King's Quest I, II or III	33.95
Lotus 1-2-3	333.95
Mark Williams C	333.95
Mastertype	26.95
Math Blaster	33.95
Mean 18	26.95
Newsroom	40 95
Pawn	29.95
Print Shop	40.95
Print Shop Companion	33.95
Radio Baseball	27 95
Sargon III	26 95
Shanghai	29.95
Sidekick	43.95
Silent Service	123.95
Space Quest	33.95
Summer Games II	26.95
Symphony	466.95
Temple of Apshai Trilogy	26.95
Ultima III	39.95
Winnie The Pooh	16.95
Winter Games	26.95
Wizardy Proving Grounds	40.95
Word Attack	33 95
Word Perfect	330 95
Wordstar 2000 +	396 95
World Games	26.95
Zork Trilogy	46 95

No surcharge for MasterCard or Visa

Ordering Information Credit Card – Have number, expiration date, shipping address and DAY phone number ready with order Personal and company checks Call order in – you will receive an order number, send order number and check to the P O Box listed Allow 14 Business Days for check to clear Money Orders Call order in – you will receive an order number, send number with money order to P O Box listed Shipping Continental USA – **All Orders add 3%**, minimum $3 00 HI, AK, Canada add 5%, minimum $5 00 All other order areas add 15% minimum $10 00 No C O D s over $100 00, add an additional $3 00 shipping charges for C O D All areas not serviced by U P S will be shipped first class insured mail All shipping charges in excess of the above terms will be added All goods are new and include factory warranty **All defectives must have a return authorization number.** Please call (313)595-0788 to obtain an R A # or return will not be accepted Prices and availability subject to change without notice All correspondence should be sent to P O. Box 75 – Wayne, MI 48184

in a loose-leaf binder.

Scholastic Software, 730 Broadway, New York, NY 10003

Circle Reader Service Number 228.

Desk Accessory For Macintosh

Target Software has begun shipping *Memorandum*, a desk accessory that allows users to attach electronic "sticky notes" to files and documents. They can be attached to cells in spreadsheets, fields in a database, or a portion of text in a word processing document. These notes can then be popped up at the user's discretion. All note files are managed via a window called the Note Manager, which allows files to be imported (appended) and exported (saved to another file). It also allows users to perform substring searches through all notes with the click of the mouse, as well as several other housekeeping functions.

Memorandum is compatible with the Macintosh 512K, Plus, SE, and II. It supports all hard disk drives and runs with virtually all Macintosh applications that support desk accessories. Suggested retail price is $99.95.

Target Software, 14206 S.W. 136th St., Miami, FL 33186

Circle Reader Service Number 229.

New Amiga Word Processor

New Horizons Software, publisher of *Flow*, an outline processor for the Amiga, is now shipping *ProWrite*, a sophisticated word processor for the Amiga. Its many features allow you to do things like create and edit documents using multiple fonts, sizes, styles, and colors; include IFF color graphics in your word processing documents; edit up to eight documents at one time, then cut and paste between them; and use the mouse or keyboard for all commands. It fully supports multitasking and all foreign characters, and is compatible with *Flow*.

ProWrite retails for $124.95.

New Horizons Software, P.O. Box 43167, Austin, TX 78745

Circle Reader Service Number 230.

Poetry Generator

Users of IBM PC and compatibles, Apple II, Macintosh, and Atari ST computers can now have their computers compose personalized poems for use in greeting cards or other correspondence.

Your Personal Poet composes customized poems after the user answers questions like *Who is this poem going to?, What is your message to him/her/them?, How can you best describe him/her/them?, Would you like your poem light and limerical or sentimental and serious?*, and *Personalize your card even more with a date or P.S. message?*. Taking your answers to these questions, the program composes and displays a unique, personal poem for and about the person or people you want to send a card to. Then you can print the poem using a decorative font onto special 5 × 7 greeting card-sized form-fed paper, remove the peel-off strip from the back, and fit it into one of the cards.

The $9.95 package includes the software disk, six sheets of greeting card paper, four greeting card covers, and four envelopes.

Computer Poet, P.O. Box 7707, Incline Village, NV 89450

Circle Reader Service Number 231.

New Scenery Disk For *Flight Simulator II* And *Jet*

SubLOGIC has released Scenery Disk #7 for Commodore 64 users of *Flight Simulator II* and *Jet*. It covers the East Coast of the U.S. in detail, from Washington D.C. through Key West, Florida. SD #7 features hundreds of miles of coastline, many rivers and roads, railroads, racetracks, transmitter towers (some with blinking lights at night), and elevated bridges that cast shadows. Highly detailed sight-seeing areas are now available, along with more generic scenery areas that include enough radio NAVaids and refueling facilitators to keep users occupied for a long time. Over 130 airports and a dozen military airports are included.

Scenery Disk #7 is available for $19.95 (plus $2 shipping and handling for mail order).

SubLOGIC, 713 Edgebrook Dr., Champaign, IL 61820

Circle Reader Service Number 232.

Educational Software For Commodore, Apple

Gamco Industries announces *Chance It!: A Game of Details* for Commodore 64 and Apple II–series computers.

This two-player educational game uses a TV game-show format to encourage students to read for detail, and includes a complete student management system as well. On each turn, students are asked a question and whether they want to "chance it." If they don't, they read a paragraph and answer a question concerning details. If they answer correctly, they get to place one of their markers on the game board. If students choose to chance it, they are shown a randomly chosen message which may instruct them to place a free marker, read a paragraph and answer the questions, lose a marker, or give the other player a free marker.

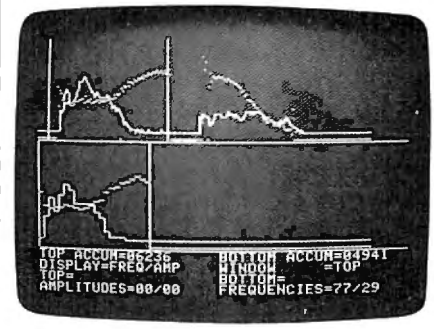
Apple II and Commodore 64 versions are available for third- and fourth-grade levels, and for fifth- and sixth-grade levels at $39.95 each.

Gamco Industries, P.O. Box 1911, Big Spring, TX 79721

Circle Reader Service Number 233.

Speech Construction Set

Covox has introduced *Speech Construction Set*, a powerful speech-editing tool that facilitates the creation of high-quality speech from data encoded using Covox's Voice Master module. Since the output is digitally constructed from the user's own voice patterns, the resulting speech is clear and natural-sounding. *Speech Construction Set* provides sophisticated routines for editing and modifying the amplitude portions of stored speech for refined quality during playback.



Speech Construction Set, *from Covox, is a multi-featured speech-editing tool.*

The program is available for Apple II–series computers at a suggested retail price of $49.95, and for Commodore 64 at $39.95.

Covox, 675-D Conger St., Eugene, OR 97402

Circle Reader Service Number 234.

Classic Adventure Game Now Available For Macintosh

A spate of adventure games have come down the pike since Don Woods and William Crowther codeveloped the acknowledged first of this popular computer genre, *Adventure* (sometimes referred to as *Original Adventure*). This classic is now available for the Macintosh (128/512K) for $29.95.

Other versions available include IBM PC and compatibles ($24.95; *Norell Data Systems, P.O. Box 70127, 3400 Wilshire Blvd., Los Angeles, CA 90010*) and Atari eight-bit ($14.95; *Lotsabytes, 15445 Yentura Blvd., Suite 10G, Sherman Oaks, CA 91413*).

L.W. James & Assoc., 1525 E. County. Rd. 58, Ft. Collins, CO 80524

Circle Reader Service Number 235.

$ WIN $ THE LOTTO

With Your Computer!

Forget random numbers. This program for home computers does an actual analysis of the past winning numbers. This amazing program will quickly provide you with all the data you need to predict which numbers will likely come up in subsequent drawings. All consistent lottery winners use some kind of system based on the past winners. Using the real power of your computer gives you a definite edge. It's menu driven and all you do is add the latest winners each week and the program does the rest in seconds. On screen or printer it shows hot and cold numbers, frequency, groups, sums-of-digits, odd/even, wheels numbers and more. No thick manual to read. It even has a built-in tutorial.

Ask your software dealer or call or write:

SOFT BYTE

SOFT-BYTE
P.O. Box 556 F. Park
Dayton, Ohio 45405
(513) 233-2200

THE LOTTO PROGRAM is designed for all 6 & 7 draw lotto games (up to 49 numbers)!

DON'T PLAY LOTTO WITHOUT IT!

APPLE & M/S DOS IBM 24.95
COMMODORE & ATARI 21.95
TRS-80 & MOD III, IV 21.95
MACINTOSH (super version) 29.95

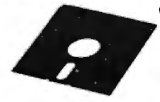
Please add $2.00 shipping/handling. Fast service on charge cards.

MasterCard VISA

Precision Data Products™

POLY PACK 5¼" DISKETTES

(From Leading Mfr.)



36¢ Each

- Blank Jackets
- WP Tabs
- Envelopes

100% Error Free
Lifetime Warranty
Sold in Lots of 100 Only

SALE!

SONY POLY PACK 3.5" DISKETTES

High Capacity Data Storage. Meets the requirements of all 3.5" Micro Diskette Drives.

SS 135TPI . . . . . $1.17 ea.
DS 135TPI . . . . . $1.29 ea.

PRINTER RIBBONS

Quality replacements for most popular printers. Min./6.

Apple Imagewriter	Black	**$3.95 ea.**
Apple Scribe	Black	**$2.95 ea.**
Epson LX 80/90	Black	**$2.95 ea.**
Okidata 80/82/83	Black	**$1.75 ea.**
Toshiba 1350	Black	**$4.89 ea.**

Color Ribbons Available At Great Savings Too

Min. Order $25.00. Add 10% for less than 50 disks. S&H: Continental USA $4.00/100 or fewer disks. $2.00 per dozen ribbons. Reduced shipping charge on larger quantities. Foreign orders, APO/FPO, please call. MI residents add 4% tax. Prices subject to change without notice. Hours: 8:30 AM - 7:00 PM ET.

Precision Data Products
P.O. Box 8367, Grand Rapids, MI 49518
(616) 452-3457 • Michigan 1-800-632-2468
Outside Michigan 1-800-258-0028

FIFTEEN TIMES THE POWER!

90% + CONFIDENCE LEVEL

Why purchase only 1 thoroughbred handicapping strategy when you can have the POWER of the 15 most popular handicapping strategies available?

A proven computer program, designed by an M.B.A. of Finance, combines improved variations of the 15 most popular handicapping strategies into one easy program. This POWERFUL program called Multi-Strats can analyze a race using 15 strategies in a fraction of the time you analyze a race using just 1 strategy.

Simply type in the answers to the program questions. All the info is in the Daily Racing Form. The results of the 15 strategies will automatically appear on your screen or printer. Multi-Strats then tabulates the 15 strategy totals to give you an ultimate number for each horse.

When 10 or more strategies select the same horse to win, that horse has over a 90% chance of winning.

Multi-Strats package includes: ★ 15 strategies ★ 40 page book (with money management) ★ 5¼" or 3½" disk or tape ★ telephone hotline ★ first class delivery ★ BONUS #1 Pick 6-10 Horse Program ★ BONUS #2 Lottery Program ★ BONUS #3 Free Las Vegas trip eligibility ★ All for $69.95 (Add $2 S & H)

Multi-Strats' Video Tutorial is available for beginning computer operators and/or handicappers on VHS or Beta for $19.95. (Add $3 S & H)

Order by Money Order, Check, Visa, MC. AMEX, or C.O.D. to Banana Software, Inc. Dept. SA. 6531 Park Avenue, Kent, OH 44240.

ORDERS (216) 673-6969 (24 hrs.)
INQUIRIES (216) 673-6167 (recording)

ALL MODELS

COMMODORE **IBM**
APPLE **RADIO SHACK**
ATARI **COLECO**

PASCAL TO BASIC TRANSLATOR

PAS-BAS 1.0 translator/tutor is all you need to learn and use the powerful PASCAL programming language.

- PASCAL to BASIC translator. A first of it's kind software writing tool. It makes a dynamic analysis of a PASCAL program and generates a compact and efficient equivalent in BASIC. The BASIC code generated by PAS-BAS is 100% ready to run in any personal computer.
- Interactive on-line tutorial. It covers everything, from first principles to advanced techniques.
- Easy to use integrated Editor.

ORDER YOURS NOW! **PAS-BAS ISIT!**
IBM APPLE COMMODORE

To order send check or money order for $49.95 + $3.00 shipping and handling to:

ISIT
International Software for Information and Translation
P.O. Box 773406
Houston, TX 77215-3406
(713) 932-1691

Save on BROWNOUT protection!

LINE CONDITIONER

keeps power constant during voltage sags and power surges!

Prevents damage and downtime on computers, phone systems, cash registers, etc. by providing full voltage support when AC input power varies up or down. Maintains constant output of 120V. *Line Stabilizer* is a stepped transformer system that has *higher* efficiency than CVT's (constant voltage transformers) and gives lower waveform distortion at FAR LOWER COST. Built-in spike and noise suppression. 1-Year warranty!

1200-Watt, 4 Outlet	1800-Watt, 6 Outlet
ONLY $199	**ONLY $259**

Order toll free 1-800-662-5021
IN ILLINOIS CALL 1-312-648-2191 OR MAIL COUPON

INDUS-TOOL, 730 W. Lake Street
Dept. C, Chicago, IL 60606
Enclosed is $_____or charge on
☐ MasterCard ☐ Visa ☐ *Expires*_____
Card No. _____
Send ☐ 1200-Watt @ $199 ☐ 1800-Watt @ $259

Name _____
Company _____
Address _____
City,State,Zip _____
Phone _____

LOTTO CIPHER™

GET THE BEST ODDS ON ANY LOTTERY
SIX NUMBER - PICK FOUR - DAILY GAME

- PRODUCES FOUR COMBINATIONS OF NUMBERS TO CHOOSE FROM.
- ANY AMOUNT OF BALLS AND NUMBERS CAN BE PROGRAMMED.
- PRINTS OUT PAST LOTTO NUMBERS DRAWN, PAST COMPUTER PICKS, AND NUMBER DRAW FREQUENCY LIST.
- RANDOM NUMBER GENERATOR INCLUDED.

$29.95 C-64/128

Window Magic

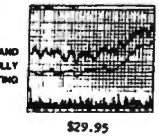
SUPER HI-RESOLUTION DRAWING IN MULTI OR MONO COLOR

- COPY
- FILL
- LINES
- DRAW
- COLOR SQUARES
- TYPES LETTERS AND GRAPHICS
- MIRROR, FLIP, AND SCROLLING WINDOWS
- ZOOM-EXPANDS A WINDOW TO DOUBLE SIZE
- SAVE AND LOAD YOUR WINDOWS ON DISK
- PRINTS ON STANDARD DOT MATRIX PRINTER
- CLONE COLOR ATTRIBUTES
- POLYGON SHAPES-EXPAND, SHRINK AND ROTATE, THEN STAMP ANYWHERE
- ZOOM PLOT-DRAW ON AN EXPANDED WINDOW AND YOUR DRAWING AT THE SAME TIME

$34.95 C-64/128

STOCK BROKER.

PROFITS GUARANTEED OR YOUR MONEY BACK



BUYING GOOD QUALITY, VOLATILE ISSUES AND USING THIS TRADING SYSTEM WILL HAVE YOU FULLY INVESTED AT THE LOWEST PRICES AND CONVERTING TO CASH AS THE STOCK NEARS ITS PEAK.

- TECHNICAL TRADING THAT WORKS.
- BAR GRAPH PRINT-OUTS.
- RECORD UP TO 144 STOCKS ON A DISK.

$29.95

COMMODORE, COMMODORE AMIGA, AND IBM SYSTEMS

ACORN OF INDIANA, INC.

2721 OHIO STREET
MICHIGAN CITY, IN 46360

219-879-2284

SHIPPING AND HANDLING, ADD $1.50 - C.O.D.'S ACCEPTED
VISA AND MASTER CARD ORDERS ADD 4%
INDIANA RESIDENTS ADD 5% SALES TAX

COMPUTE!'s Author's Guide

Most of the following suggestions serve to improve the speed and accuracy of publication. COMPUTE! is primarily interested in new and timely articles on the Commodore 64/128, Atari, Apple, IBM PC/PCjr, Amiga, and Atari ST. We are much more concerned with the content of an article than with its style, but articles should be clear and well-explained.

The guidelines below will permit your good ideas and programs to be more easily edited and published:

1. The upper left corner of the first page should contain your name, address, telephone number, and the date of submission.

2. The following information should appear in the upper right corner of the first page: If your article is specifically directed to one make of computer, please state the brand name and, if applicable, the BASIC or ROM or DOS version(s) involved. In addition, *please indicate the memory requirements of programs.*

3. The underlined title of the article should be placed about ⅔ of the way down the first page.

4. Following pages should be typed normally, except that in the upper right corner there should be an abbreviation of the title, your last name, and the page number—for example: Memory Map/Smith/2.

5. All lines within the text of the article must be double- or triple-spaced. A one-inch margin should be left at the right, left, top, and bottom of each page. No words should be divided at the ends of lines. And please do not right-justify. Leave the lines ragged.

6. Standard typing paper should be used (no erasable, onionskin, or other thin paper), and typing should be on one side of the paper only (upper- and lowercase).

7. If you are submitting more than one article, send each one in a separate mailer with its own tape or disk.

8. Short programs (under 20 lines) can easily be included within the text. Longer programs should be separate listings. *It is essential that we have a copy of the program, recorded twice, on a tape or disk.* If your article was written with a word processor, we request that you include a copy of the text file on the tape or disk. If you include a copy of your article on disk, please save the article as plain text, without any special formatting characters or control codes. Most word processors provide an option for saving a document as plain ASCII text or in unformatted form. Please use high-quality 10- or 30-minute tapes with the program recorded on both sides. The tape or disk should be labeled with your name, the title of the article, and, if applicable, the BASIC/ROM/DOS version(s). Tapes are fairly sturdy, but disks need to be enclosed within plastic or cardboard mailers (available at photography, stationery, or computer supply stores). If possible, programs written in machine language or a compiled language should include source code (or an annotated disassembly if the program was written with a machine language monitor).

9. A good general rule is to spell out the numbers zero through ten in your article and write higher numbers as numerals (1024). The exceptions to this are: Figure 5, Table 3, TAB(4), and so on. Within ordinary text, however, the zero through ten should appear as words, not numbers. Also, symbols and abbreviations should not be used within text: Use *and* (not &), *reference* (not ref.), *through* (not thru).

10. For greater clarity, use all capitals when referring to keys (RETURN, TAB, ESC, SHIFT), BASIC words (LIST, RND, GOTO), and three languages (BASIC, APL, PILOT). Headlines and subheads should, however, be initial caps only, and emphasized words are not capitalized. If you wish to emphasize, underline the word; then it will be italicized during typesetting.

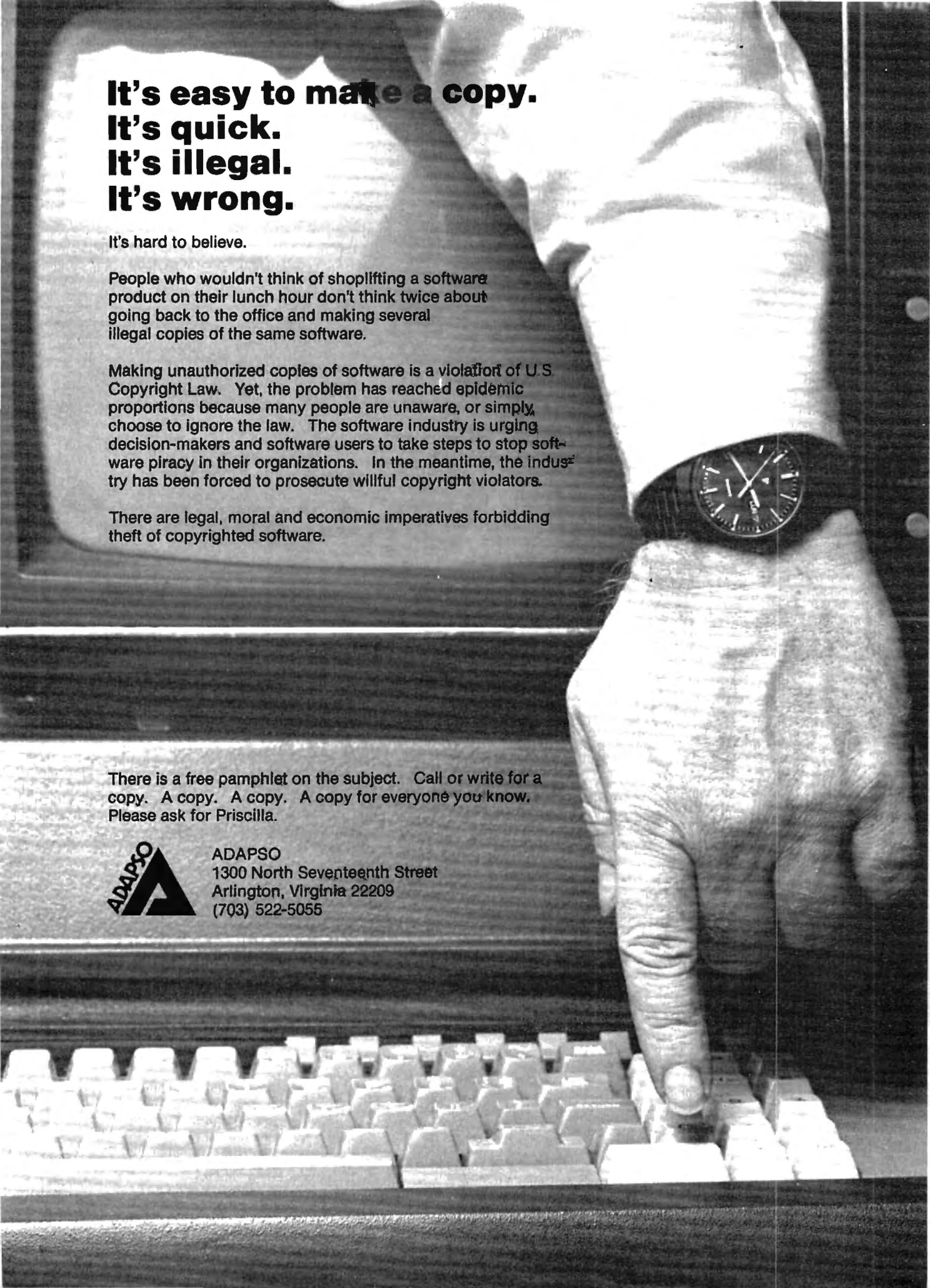
11. Articles can be of any length—from a single-line routine to a multiple-issue series. The average article is about four to eight double-spaced, typed pages.

12. We do not consider articles which are submitted simultaneously to other publishers. If you wish to send an article to another magazine for consideration, please do not submit it to us.

13. COMPUTE! pays between $70 and $800 for published articles. In general, the rate reflects the length and quality of the article. Payment is made upon acceptance. Following submission (to Editorial Department, COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC 27403), it will take from three to six weeks for us to reply. If your work is accepted, you will be notified by a letter which will include a contract for you to sign and return. *Rejected manuscripts are returned to authors who enclose a self-addressed, stamped envelope.*

14. If your article is accepted and you have since made improvements to the program, please submit an entirely new tape or disk and a new copy of the article reflecting the update. We cannot easily make revisions to programs and articles. It is necessary that you send the revised version as if it were a new submission entirely, but be sure to indicate that your submission is a revised version by writing *Revision* on the envelope and the article.

15. COMPUTE! does not accept unsolicited product reviews. If you are interested in serving on our panel of reviewers, contact the Features Editor for details.


It's easy to make a copy.
It's quick.
It's illegal.
It's wrong.
It's hard to believe.
People who wouldn't think of shoplifting a software product on their lunch hour don't think twice about going back to the office and making several illegal copies of the same software.
Making unauthorized copies of software is a violation of U.S. Copyright Law. Yet, the problem has reached epidemic proportions because many people are unaware, or simply choose to ignore the law. The software industry is urging decision-makers and software users to take steps to stop software piracy in their organizations. In the meantime, the industry has been forced to prosecute willful copyright violators.
There are legal, moral and economic imperatives forbidding theft of copyrighted software.
There is a free pamphlet on the subject. Call or write for a copy. A copy. A copy. A copy for everyone you know. Please ask for Priscilla.
ADAPSO
ADAPSO
1300 North Seventeenth Street
Arlington, Virginia 22209
(703) 522-5055

COMPUTE!'s Guide To Typing In Programs

Computers are precise—type the program *exactly* as listed, including necessary punctuation and symbols, except for special characters noted below. We have provided a special listing convention as well as a program to check your typing—"The Automatic Proofreader."

Programs for the IBM, TI-99/4A, and Atari ST models should be typed exactly as listed; no special characters are used. Programs for Commodore, Apple, and Atari 400/800/XL/XE computers may contain some hard-to-read special characters, so we have a listing system that indicates these control characters. You will find these Commodore and Atari characters in curly braces; *do not type the braces.* For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. A complete list of these symbols is shown in the tables below. For Commodore, Apple, and Atari, a single symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple.

Graphics characters entered with the Commodore logo key are enclosed in a special bracket: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (white on black) should be entered with the inverse video

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -		Cursor Up
{DOWN}	ESC CTRL =		Cursor Down
{LEFT}	ESC CTRL +		Cursor Left
{RIGHT}	ESC CTRL *		Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Read:	Press:	See:
{CLR}	SHIFT + CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT + ↑ CRSR ↓	
{DOWN}	↑ CRSR ↓	
{LEFT}	SHIFT + ← CRSR →	
{RIGHT}	← CRSR →	
{RVS}	CTRL 9	
{OFF}	CTRL 0	
{BLK}	CTRL 1	
{WHT}	CTRL 2	
{RED}	CTRL 3	
{CYN}	CTRL 4	
{PUR}	CTRL 5	
{GRN}	CTRL 6	
{BLU}	CTRL 7	
{YEL}	CTRL 8	

When You Read:	Press:	See:
[<1>]	COMMODORE 1	
[<2>]	COMMODORE 2	
[<3>]	COMMODORE 3	
[<4>]	COMMODORE 4	
[<5>]	COMMODORE 5	
[<6>]	COMMODORE 6	
[<7>]	COMMODORE 7	
[<8>]	COMMODORE 8	
{ F1 }	f1	
{ F2 }	SHIFT + f1	
{ F3 }	f3	
{ F4 }	SHIFT + f3	
{ F5 }	f5	
{ F6 }	SHIFT + f5	
{ F7 }	f7	
{ F8 }	SHIFT + f7	
←	←	

key (Atari logo key on 400/800 models).

Whenever more than two spaces appear in a row, they are listed in a special format. For example, {6 SPACES} means press the space bar six times. Our Commodore listings never leave a single space at the end of a line, instead moving it to the next printed line as {SPACE}.

Amiga program listings contain only one special character, the left arrow (←) symbol. This character marks the end of each program line. Wherever you see a left arrow, press RETURN or move the cursor off the line to enter that line into memory. Don't try to type in the left arrow symbol; it's there only as a marker to indicate where each program line ends.

The Automatic Proofreader

Type in the appropriate program listed below, then save it for future use. The Commodore Proofreader works on the Commodore 128, 64, Plus/4, 16, and VIC-20. Don't omit any lines, even if they contain unfamiliar commands or you think they don't apply to your computer. When you run the program, it installs a machine language program in memory and erases its BASIC portion automatically (so be sure to save several copies before running the program for the first time). If you're using a Commodore 128, Plus/4 or 16, do *not* use any GRAPHIC commands while the Proofreader is active. You should disable the Commodore Proofreader before running any other program. To do this, either turn the computer off and on or enter SYS 64738 (for the 64), SYS 65341 (128), SYS 64802 (VIC-20), or SYS 65526 (Plus/4 or 16). To reenable the Proofreader, reload the program and run it as usual. Unlike the original VIC/64 Proofreader, this version works the same with disk or tape.

On the Atari, run the Proofreader to activate it (the Proofreader remains active in memory as a machine language program); you must then enter NEW to erase the BASIC loader. Pressing SYSTEM RESET deactivates the Atari Proofreader; enter PRINT USR(1536) to reenable it.

The Apple Proofreader erases the BASIC portion of itself after you run it, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program.

The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate. Be sure to leave Caps Lock on, except when typing lowercase characters.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a hexadecimal number (on the Apple) or a pair of letters (on the Commodore, Atari, or IBM) appears. The number or pair of letters is called a *checksum*.

Compare the value displayed on the screen by the Proofreader with the checksum printed in the program listing in the magazine. The checksum is given to the left of each line number. Just type in the program a line at a time (without the printed checksum), press RETURN or Enter, and compare the checksums. If they match, go on to the next line. If not, check your typing; you've made a mistake. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. On the Atari and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Atari Proofreader does not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. The Commodore Proofreader catches transposition errors and ignores spaces unless they're enclosed in quotation marks. The IBM Proofreader detects errors in spacing and transposition.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader prompts you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert an existing BASIC program to Proofreader format, save it to disk with SAVE *"filename"*,A.

Program 1: Atari Proofreader

By Charles Brannon

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:REA
    D A:POKE I,A:CK=CK+A:N
    EXT I
120 IF CK<>19072 THEN ? "E
    rror in DATA Statement
    s.  Check Typing.":END

130 A=USR(1536)
140 ? :? "Automatic Proofr
    eader Now Activated."
150 END
160 DATA 104,160,0,185,26,
    3,201,69,240,7
170 DATA 200,200,192,34,20
    8,243,96,200,169,74
180 DATA 153,26,3,200,169,
    6,153,26,3,162
190 DATA 0,189,0,228,157,7
    4,6,232,224,16
200 DATA 208,245,169,93,14
    1,78,6,169,6,141
210 DATA 79,6,24,173,4,228
    ,105,1,141,95
220 DATA 6,173,5,228,105,0
    ,141,96,6,169
230 DATA 0,133,203,96,247,
    238,125,241,93,6
240 DATA 244,241,115,241,1
    24,241,76,205,238
250 DATA 0,0,0,0,0,32,62,2
    46,8,201
260 DATA 155,240,13,201,32
    ,240,7,72,24,101
270 DATA 203,133,203,104,4
    0,96,72,152,72,138
280 DATA 72,160,0,169,128,
    145,88,200,192,40
290 DATA 208,249,165,203,7
    4,74,74,74,24,105
300 DATA 161,160,3,145,88,
    165,203,41,15,24
310 DATA 105,161,200,145,8
    8,169,0,133,203,104
320 DATA 170,104,168,104,4
    0,96
```

Program 2: IBM Proofreader

By Charles Brannon

```
10 'Automatic Proofreader Vers
   ion 3.0 (Lines 205,206 adde
   d/190 deleted/470,490 chang
   ed from V2.0)
100 DIM L$(500),LNUM(500):COLO
    R 0,7,7:KEY OFF:CLS:MAX=0:
    LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,C
    HR$(4)+CHR$(70):ON KEY(15)
     GOSUB 640:KEY (15) ON:GOT
    O 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:PR
    INT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-INT
    (LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:POK
    E 1052,34:POKE 1054,0:POKE
     1055,79:POKE 1056,13:POKE
     1057,28:LINE INPUT L$:DEF
     SEG:IF L$="" THEN 150
170 IF LEFT$(L$,1)=" " THEN L$
    =MID$(L$,2):GOTO 170
```

```
180 IF VAL(LEFT$(L$,2))=0 AND
    MID$(L$,3,1)=" " THEN L$=M
    ID$(L$,4)
200 IF ASC(L$)>57 THEN 260 'no
     line number, therefore co
    mmand
205 BL=INSTR(L$," "):IF BL=0 T
    HEN BL$=L$:GOTO 206 ELSE B
    L$=LEFT$(L$,BL-1)
206 LNUM=VAL(BL$):TEXT$=MID$(L
    $,LEN(STR$(LNUM))+1)
210 IF TEXT$="" THEN GOSUB 540
    :IF LNUM=LNUM(P) THEN GOSU
    B 560:GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TO LEN(L$)
    :CKSUM=(CKSUM+ASC(MID$(L$,
    I))*I) AND 255:NEXT:LOCATE
     Y,1:PRINT CHR$(65+CKSUM/1
    6)+CHR$(65+(CKSUM AND 15))
    +" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM
    THEN L$(P)=TEXT$:GOTO 150
    'replace line
240 GOSUB 580:GOTO 150 'insert
     the line
260 TEXT$="":FOR I=1 TO LEN(L$
    ):A=ASC(MID$(L$,I)):TEXT$=
    TEXT$+CHR$(A+32*(A>96 AND
    A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," ")
    :COMMAND$=TEXT$:ARG$="":IF
     DELIMITER THEN COMMAND$=L
    EFT$(TEXT$,DELIMITER-1):AR
    G$=MID$(TEXT$,DELIMITER+1)
     ELSE DELIMITER=INSTR(TEXT
    $,CHR$(34)):IF DELIMITER T
    HEN COMMAND$=LEFT$(TEXT$,D
    ELIMITER-1):ARG$=MID$(TEXT
    $,DELIMITER)
280 IF COMMAND$<>"LIST" THEN 4
    10
290 OPEN "scrn:" FOR OUTPUT AS
     #1
300 IF ARG$="" THEN FIRST=0:P=
    MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-"):
    IF DELIMITER=0 THEN LNUM=V
    AL(ARG$):GOSUB 540:FIRST=P
    :GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELIM
    ITER)):LAST=VAL(MID$(ARG$,
    DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRST
    =P:LNUM=LAST:GOSUB 540:IF
    P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(S
    TR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GOT
    O 370
360 CKSUM=0:A$=N$+L$(X):FOR I=
    1 TO LEN(A$):CKSUM=(CKSUM+
    ASC(MID$(A$,I))*I) AND 255
    :NEXT:A$=CHR$(65+CKSUM/16)
    +CHR$(65+(CKSUM AND 15))+"
     "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN O
    PEN "lpt1:" FOR OUTPUT AS
    #1:GOTO 300
420 IF COMMAND$="CHECK" THEN C
    KFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN 4
    50
440 GOSUB 600:OPEN ARG$ FOR OU
    TPUT AS #1:ARG$="":GOTO 30
    0
450 IF COMMAND$<>"LOAD" THEN 4
    90
460 GOSUB 600:OPEN ARG$ FOR IN
    PUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INPU
    T #1,L$:BL=INSTR(L$," "):B
    L$=LEFT$(L$,BL-1):LNUM(P)=
    VAL(BL$):L$(P)=MID$(L$,LEN
    (STR$(VAL(BL$)))+1):P=P+1:
    WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN INP
    UT "Erase program - Are yo
    u sure";L$:IF LEFT$(L$,1)=
    "y" OR LEFT$(L$,1)="Y" THE
    N MAX=0:LNUM(0)=65536!:GOT
    O 130:ELSE 130
500 IF COMMAND$="BASIC" THEN C
    OLOR 7,0,0:ON ERROR GOTO 0
    :CLS:END
510 IF COMMAND$<>"FILES" THEN
    520
515 IF ARG$="" THEN ARG$="A:"
    ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO 1
    30
540 P=0:WHILE LNUM>LNUM(P) AND
     P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:L
    NUM(X)=LNUM(X+1):L$(X)=L$(
    X+1):NEXT:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+1
     STEP -1:LNUM(X)=LNUM(X-1)
    :L$(X)=L$(X-1):NEXT:L$(P)=
    TEXT$:LNUM(P)=LNUM:RETURN
600 IF LEFT$(ARG$,1)<>CHR$(34)
     THEN 520 ELSE ARG$=MID$(A
    RG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34)
     THEN ARG$=LEFT$(ARG$,LEN(
    ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,".
    ")=0 THEN ARG$=ARG$+".BAS"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"St
    opped.":RETURN 150
650 PRINT "Error #";ERR:RESUME
     150
```

Program 3: Commodore Proofreader

By Philip Nelson, Assistant Editor

```
10 VEC=PEEK(772)+256*PEEK(773)
   :LO=43:HI=44
20 PRINT "AUTOMATIC PROOFREADE
   R FOR ";:IF VEC=42364 THEN
   {SPACE}PRINT "C-64"
30 IF VEC=50556 THEN PRINT "VI
   C-20"
40 IF VEC=35158 THEN GRAPHIC C
   LR:PRINT "PLUS/4 & 16"
50 IF VEC=17165 THEN LO=45:HI=
   46:GRAPHIC CLR:PRINT"128"
60 SA=(PEEK(LO)+256*PEEK(HI))+
   6:ADR=SA
70 FOR J=0 TO 166:READ BYT:POK
   E ADR,BYT:ADR=ADR+1:CHK=CHK
   +BYT:NEXT
80 IF CHK<>20570 THEN PRINT "*
    ERROR* CHECK TYPING IN DATA
    STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:
   RS=SA+RF:HB=INT(RS/256):LB=
   RS-(256*HB)
100 CHK=CHK+RF+LF+HF:POKE SA+L
   F,LB:POKE SA+HF,HB:NEXT
110 IF CHK<>22054 THEN PRINT "
   *ERROR* RELOAD PROGRAM AND
   {SPACE}CHECK FINAL LINE":EN
   D
120 POKE SA+149,PEEK(772):POKE
    SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+
   14,22:POKE SA+18,23:POKESA+
   29,224:POKESA+139,224
140 PRINT CHR$(147);CHR$(17);"
   PROOFREADER ACTIVE":SYS SA
150 POKE HI,PEEK(HI)+1:POKE (P
   EEK(LO)+256*PEEK(HI))-1,0:N
   EW
160 DATA 120,169,73,141,4,3,16
   9,3,141,5,3
170 DATA 88,96,165,20,133,167,
   165,21,133,168,169
180 DATA 0,141,0,255,162,31,18
   1,199,157,227,3
190 DATA 202,16,248,169,19,32,
   210,255,169,18,32
200 DATA 210,255,160,0,132,180
   ,132,176,136,230,180
210 DATA 200,185,0,2,240,46,20
   1,34,208,8,72
220 DATA 165,176,73,255,133,17
   6,104,72,201,32,208
230 DATA 7,165,176,208,3,104,2
   08,226,104,166,180
240 DATA 24,165,167,121,0,2,13
   3,167,165,168,105
250 DATA 0,133,168,202,208,239
   ,240,202,165,167,69
260 DATA 168,72,41,15,168,185,
   211,3,32,210,255
270 DATA 104,74,74,74,74,168,1
   85,211,3,32,210
280 DATA 255,162,31,189,227,3,
   149,199,202,16,248
290 DATA 169,146,32,210,255,76
   ,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,
   77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,
   116,117,151,128,129,167,136
   ,137
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
10 C = 0: FOR I = 768 TO 768 +
   68: READ A:C = C + A: POKE I
   ,A: NEXT
20 IF C < > 7258 THEN PRINT "ER
   ROR IN PROOFREADER DATA STAT
   EMENTS": END
30 IF PEEK (190 * 256) < > 76 T
   HEN POKE 56,0: POKE 57,3: CA
   LL 1002: GOTO 50
40 PRINT CHR$ (4);"IN#A$300"
50 POKE 34,0: HOME : POKE 34,1:
    VTAB 2: PRINT "PROOFREADER
   INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,141
110 DATA 208,60,138,72,169,0
120 DATA 72,189,255,1,201,160
130 DATA 240,8,104,10,125,255
140 DATA 1,105,0,72,202,208
150 DATA 238,104,170,41,15,9
160 DATA 48,201,58,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,41,15,9
190 DATA 48,201,58,144,2,233
200 DATA 57,141,0,4,104,170
210 DATA 169,141,96
```

MLX Machine Language Entry Program For Commodore 64 And 128

Ottis Cowper, Technical Editor

"MLX" is a labor-saving utility that allows almost fail-safe entry of machine language programs. Included are versions for the Commodore 64 and 128.

Type in and save some copies of whichever version of MLX is appropriate for your computer (you'll want to use it to enter future ML programs from COMPUTE!). Program 1 is for the Commodore 64, and Program 2 is for the 128 (128 MLX can also be used to enter Commodore 64 ML programs for use in 64 mode). When you're ready to enter an ML program, load and run MLX. It asks you for a starting address and an ending address. These addresses appear in the article accompanying the MLX-format program listing you're typing.

If you're unfamiliar with machine language, the addresses (and all other values you enter in MLX) may appear strange. Instead of the usual decimal numbers you're accustomed to, these numbers are in *hexadecimal*—a base 16 numbering system commonly used by ML programmers. Hexadecimal—hex for short—includes the numerals 0–9 and the letters A–F. But don't worry—even if you know nothing about ML or hex, you should have no trouble using MLX.

After you enter the starting and ending addresses, you'll be offered the option of clearing the workspace. Choose this option if you're starting to enter a new listing. If you're continuing a listing that's partially typed from a previous session, don't choose this option.

A functions menu will appear. The first option in the menu is ENTER DATA. If you're just starting to type in a program, pick this. Press the E key, and type the first number in the first line of the program listing. If you've already typed in part of a program, type the line number where you left off typing at the end of the previous session (be sure to load the partially completed program before you resume entry). In any case, make sure the address you enter corresponds to the address of a line in the listing you are entering. Otherwise, you'll be unable to enter the data correctly. If you pressed E by mistake, you can return to the command menu by pressing RETURN alone when asked for the address. (You can get back to the menu from most options by pressing RETURN with no other input.)

Entering A Listing

Once you're in Enter mode, MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight data bytes and a checksum. Although an MLX-format listing appears similar to the "hex dump" listings from a machine language monitor program, the extra checksum number on the end allows MLX to check your typing. (Commodore 128 users *can* enter the data from an MLX listing using the built-in monitor if the rightmost column of data is omitted, but we recommend against it. It's much easier to let MLX do the proofreading and error checking for you.)

Figure 1: 64 MLX Keypad

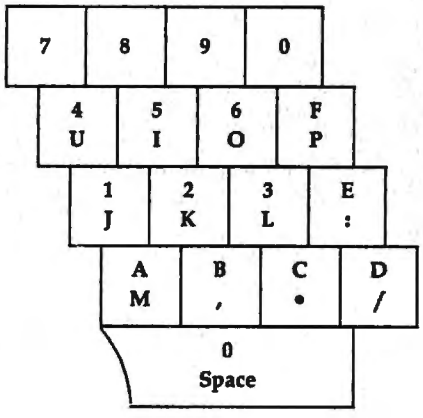
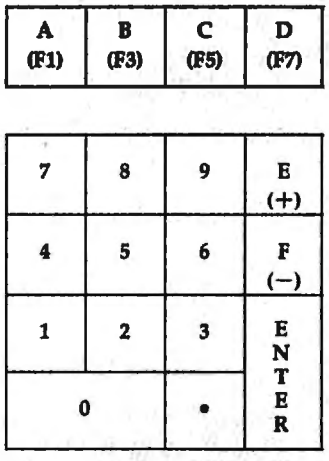


Figure 2: 128 MLX Keypad



When you enter a line, MLX recalculates the checksum from the eight bytes and the address and compares this value to the number from the ninth column. If the values match, you'll hear a bell tone, the data will be added to the workspace area, and the prompt for the next line of data will appear. But if MLX detects a typing error, you'll hear a low buzz and see an error message. The line will then be redisplayed for editing.

Invalid Characters Banned

Only a few keys are active while you're entering data, so you may have to unlearn some habits. You *do not* type spaces between the columns; MLX automatically inserts these for you. You *do not* press RETURN after typing the last number in a line; MLX automatically enters and checks the line after you type the last digit.

Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), you'll hear a warning buzz. To simplify typing, 128 MLX redefines the function keys and + and − keys on the numeric keypad so that you can enter data one-handed. In either case, the keypad is active only while entering data. Addresses must be entered with the normal letter and number keys. The figures below show the keypad configurations for each version.

MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, MLX will catch your mistake. There is one error that can slip past MLX: Because of the checksum formula used, MLX won't notice if you accidentally type FF in place of 00, and vice versa. And there's a very slim chance that you could garble a line and still end up with a combination of characters that adds up to the proper checksum. However, these mistakes should not occur if you take reasonable care while entering data.

Editing Features

To correct typing mistakes before finishing a line, use the INST/DEL key to delete the character to the left of the cursor. (The cursor-left key also deletes.) If you mess up a line really badly, press CLR/HOME to start the line over. The RETURN key is also active, but only before any data is typed on a line. Pressing RETURN at this point returns you to the command menu. After you

type a character of data, MLX disables RETURN until the cursor returns to the start of a line. Remember, you can press CLR/HOME to quickly get to a line number prompt.

More editing features are available when correcting lines in which MLX has detected an error. To make corrections in a line that MLX has redisplayed for editing, compare the line on the screen with the one printed in the listing, then move the cursor to the mistake and type the correct key. The cursor left and right keys provide the normal cursor controls. (The INST/DEL key now works as an alternative cursor-left key.) You cannot move left beyond the first character in the line. If you try to move beyond the rightmost character, you'll reenter the line. During editing, RETURN is active; pressing it tells MLX to recheck the line. You can press the CLR/HOME key to clear the entire line if you want to start from scratch, or if you want to get to a line number prompt to use RETURN to get back to the menu.

Display Data

The second menu choice, DISPLAY DATA, examines memory and shows the contents in the same format as the program listing (including the checksum). When you press D, MLX asks you for a starting address. Be sure that the starting address you give corresponds to a line number in the listing. Otherwise, the checksum display will be meaningless. MLX displays program lines until it reaches the end of the program, at which point the menu is redisplayed. You can pause the display by pressing the space bar. (MLX finishes printing the current line before halting.) Press space again to restart the display. To break out of the display and get back to the menu before the ending address is reached, press RETURN.

Other Menu Options

Two more menu selections let you save programs and load them back into the computer. These are SAVE FILE and LOAD FILE; their operation is quite straightforward. When you press S or L, MLX asks you for the filename. You'll then be asked to press either D or T to select disk or tape.

You'll notice the disk drive starting and stopping several times during a load or save (save only for the 128 version). Don't panic; this is normal behavior. MLX opens and reads from or writes to the file instead of using the usual LOAD and SAVE commands (128 MLX makes use of BLOAD). Disk users should also note that the drive prefix 0: is automatically added to the filename (line 750 in 64 MLX), so this should *not* be included when entering the name. This also precludes the use of @ for Save-with-Replace, so remember to give each version you save a different name. The 128 version makes up for this by giving you the option of scratching the existing file if you want to reuse a filename.

Remember that MLX saves the entire workspace area from the starting address to the ending address, so the save or load may take longer than you might expect if you've entered only a small amount of data from a long listing. When saving a partially completed listing, make sure to note the address where you stopped typing so you'll know where to resume entry when you reload.

MLX reports the standard disk or tape error messages if any problems are detected during the save or load. (Tape users should bear in mind that Commodore computers are never able to detect errors during a save to tape.) MLX also has three special load error messages: INCORRECT STARTING ADDRESS, which means the file you're trying to load does not have the starting address you specified when you ran MLX; LOAD ENDED AT *address*, which means the file you're trying to load ends before the ending address you specified when you started MLX; and TRUNCATED AT ENDING ADDRESS, which means the file you're trying to load extends beyond the ending address you specified when you started MLX. If you see one of these messages and feel certain that you've loaded the right file, exit and rerun MLX, being careful to enter the correct starting and ending addresses.

The 128 version also has a CATALOG DISK option so you can view the contents of the disk directory before saving or loading.

The QUIT menu option has the obvious effect—it stops MLX and enters BASIC. The RUN/STOP key is disabled, so the Q option lets you exit the program without turning off the computer. (Of course, RUN/STOP–RESTORE also gets you out.) You'll be asked for verification; press Y to exit to BASIC, or any other key to return to the menu. After quitting, you can type RUN again and reenter MLX without losing your data, as long as you don't use the clear workspace option.

The Finished Product

When you've finished typing all the data for an ML program and saved your work, you're ready to see the results. The instructions for loading and using the finished product vary from program to program. Some ML programs are designed to be loaded and run like BASIC programs, so all you need to type is LOAD *"filename"*,8 for disk (DLOAD *"filename"* on the 128) or LOAD *"filename"* for tape, and then RUN. Such programs will usually have a starting address of 0801 for the 64 or 1C01 for the 128. Other programs must be reloaded to specific addresses with a command such as LOAD *"filename"*,8,1 for disk (BLOAD *"filename"* on the 128) or LOAD *"filename"*,1,1 for tape, then started with a SYS to a particular memory address. On the Commodore 64, the most common starting address for such programs is 49152, which corresponds to MLX address C000. In either case, you should always refer to the article which accompanies the ML listing for information on loading and running the program.

An Ounce Of Prevention

By the time you finish typing in the data for a long ML program, you may have several hours invested in the project. Don't take chances—use our "Automatic Proofreader" to type the new MLX, and then test your copy *thoroughly* before first using it to enter any significant amount of data. Make sure all the menu options work as they should. Enter fragments of the program starting at several different addresses, then use the Display option to verify that the data has been entered correctly. And be sure to test the Save and Load options several times to insure that you can recall your work from disk or tape. Don't let a simple typing error in the new MLX cost you several nights of hard work.

Program 1: MLX For Commodore 64

```
SS 10 REM VERSION 1.1: LINES 8
      30,950 MODIFIED, LINES 4
      85-487 ADDED
EK 100 POKE 56,50:CLR:DIM IN$,
      I,J,A,B,A$,B$,A(7),N$
DM 110 C4=48:C6=16:C7=7:Z2=2:Z
      4=254:Z5=255:Z6=256:Z7=
      127
CJ 120 FA=PEEK(45)+Z6*PEEK(46)
      :BS=PEEK(55)+Z6*PEEK(56
      ):H$="0123456789ABCDEF"
SB 130 R$=CHR$(13):L$="{LEFT}"
      :S$=" ":D$=CHR$(20):Z$=
      CHR$(0):T$="{13 RIGHT}"
CQ 140 SD=54272:FOR I=SD TO SD
      +23:POKE I,0:NEXT:POKE
      {SPACE}SD+24,15:POKE 78
      8,52
FC 150 PRINT"{CLR}"CHR$(142)CH
      R$(8):POKE 53280,15:POK
      E 53281,15
EJ 160 PRINT T$" {RED}{RVS}
      {2 SPACES}{8 @}
      {2 SPACES}"SPC(28)"
      {2 SPACES}{OFF}{BLU} ML
      X II {RED}{RVS}
      {2 SPACES}"SPC(28)"
      {12 SPACES}{BLU}"
FR 170 PRINT"{3 DOWN}
      {3 SPACES}COMPUTE!'S MA
```

```
      CHINE LANGUAGE EDITOR
      {3 DOWN}"
JB 180 PRINT"{BLK}STARTING ADD
      RESS[4$]";:GOSUB300:SA=A
      D:GOSUB1040:IF F THEN18
      0
GF 190 PRINT"{BLK}{2 SPACES}EN
      DING ADDRESS[4$]";:GOSUB
      300:EA=AD:GOSUB1030:IF
      [SPACE}F THEN190
KR 200 INPUT"{3 DOWN}{BLK}CLEA
      R WORKSPACE [Y/N][4$]";A
      $:IF LEFT$(A$,1)<>"Y"TH
      EN220
PG 210 PRINT"{2 DOWN}{BLU}WORK
      ING...";:FORI=BS TO BS+
      EA-SA+7:POKE I,0:NEXT:P
      RINT"DONE"
DR 220 PRINTTAB(10)"{2 DOWN}
      {BLK}{RVS} MLX COMMAND
      {SPACE}MENU {DOWN}[4$]":
      PRINT T$"{RVS}E{OFF}NTE
      R DATA"
BD 230 PRINT T$"{RVS}D{OFF}ISP
      LAY DATA":PRINT T$"
      {RVS}L{OFF}OAD FILE"
JS 240 PRINT T$"{RVS}S{OFF}AVE
       FILE":PRINT T$"{RVS}Q
      {OFF}UIT{2 DOWN}{BLK}"
JH 250 GET A$:IF A$=N$ THEN250
HK 260 A=0:FOR I=1 TO 5:IF A$=
      MID$("EDLSQ",I,1)THEN A
      =I:I=5
FD 270 NEXT:ON A GOTO420,610,6
      90,700,280:GOSUB1060:GO
      TO250
EJ 280 PRINT"{RVS} QUIT ":INPU
      T"{DOWN}[4$]ARE YOU SURE
       [Y/N]";A$:IF LEFT$(A$,
      1)<>"Y"THEN220
EM 290 POKE SD+24,0:END
JX 300 IN$=N$:AD=0:INPUTIN$:IF
      LEN(IN$)<>4THENRETURN
KF 310 B$=IN$:GOSUB320:AD=A:B$
      =MID$(IN$,3):GOSUB320:A
      D=AD*256+A:RETURN
PP 320 A=0:FOR J=1 TO 2:A$=MID
      $(B$,J,1):B=ASC(A$)-C4+
      (A$>"@")*C7:A=A*C6+B
JA 330 IF B<0 OR B>15 THEN AD=
      0:A=-1:J=2
GX 340 NEXT:RETURN
CH 350 B=INT(A/C6):PRINT MID$(
      H$,B+1,1);:B=A-B*C6:PRI
      NT MID$(H$,B+1,1);:RETU
      RN
RR 360 A=INT(AD/Z6):GOSUB350:A
      =AD-A*Z6:GOSUB350:PRINT
      ":";
BE 370 CK=INT(AD/Z6):CK=AD-Z4*
      CK+Z5*(CK>Z7):GOTO390
PX 380 CK=CK*Z2+Z5*(CK>Z7)+A
JC 390 CK=CK+Z5*(CK>Z5):RETURN
QS 400 PRINT"{DOWN}STARTING AT
      [4$]";:GOSUB300:IF IN$<>
      N$ THEN GOSUB1030:IF F
      {SPACE}THEN400
EX 410 RETURN
HD 420 PRINT"{RVS} ENTER DATA
      {SPACE}":GOSUB400:IF IN
      $=N$ THEN220
JK 430 OPEN3,3:PRINT
SK 440 POKE198,0:GOSUB360:IF F
       THEN PRINT IN$:PRINT"
      {UP}{5 RIGHT}";
GC 450 FOR I=0 TO 24 STEP 3:B$
      =S$:FOR J=1 TO 2:IF F T
      HEN B$=MID$(IN$,I+J,1)
HA 460 PRINT"{RVS}"B$L$;:IF I<
      24THEN PRINT"{OFF}";
HD 470 GET A$:IF A$=N$ THEN470
FK 480 IF(A$>"/"ANDA$<":")OR(A
      $>"@"ANDA$<"G")THEN540
GS 485 A=-(A$="M")-2*(A$=",")-
      3*(A$=".")-4*(A$="/")-5
      *(A$="J")-6*(A$="K")
FX 486 A=A-7*(A$="L")-8*(A$="
      ")-9*(A$="U")-10*(A$="I
      ")-11*(A$="O")-12*(A$="
      P")
CM 487 A=A-13*(A$=S$):IF A THE
      N A$=MID$("ABCD123E456F
      0",A,1):GOTO 540
MP 490 IF A$=R$ AND((I=0)AND(J
      =1)OR F)THEN PRINT B$;:
      J=2:NEXT:I=24:GOTO550
KC 500 IF A$="{HOME}" THEN PRI
      NT B$:J=2:NEXT:I=24:NEX
      T:F=0:GOTO440
MX 510 IF(A$="{RIGHT}")ANDF TH
      ENPRINT B$L$;:GOTO540
GK 520 IF A$<>L$ AND A$<>D$ OR
      ((I=0)AND(J=1))THEN GOS
      UB1060:GOTO470
HG 530 A$=L$+S$+L$:PRINT B$L$;
      :J=2-J:IF J THEN PRINT
      {SPACE}L$;:I=I-3
QS 540 PRINT A$;:NEXT J:PRINT
      {SPACE}S$;
PM 550 NEXT I:PRINT:PRINT"{UP}
      {5 RIGHT}";:INPUT#3,IN$
      :IF IN$=N$ THEN CLOSE3:
      GOTO220
QC 560 FOR I=1 TO 25 STEP3:B$=
      MID$(IN$,I):GOSUB320:IF
       I<25 THEN GOSUB380:A(I
      /3)=A
PK 570 NEXT:IF A<>CK THEN GOSU
      B1060:PRINT"{BLK}{RVS}
      {SPACE}ERROR: REENTER L
      INE [4$]":F=1:GOTO440
HJ 580 GOSUB1080:B=BS+AD-SA:FO
      R I=0 TO 7:POKE B+I,A(I
      ):NEXT
QQ 590 AD=AD+8:IF AD>EA THEN C
      LOSE3:PRINT"{DOWN}{BLU}
      ** END OF ENTRY **{BLK}
      {2 DOWN}":GOTO700
GQ 600 F=0:GOTO440
QA 610 PRINT"{CLR}{DOWN}{RVS}
      {SPACE}DISPLAY DATA ":G
      OSUB400:IF IN$=N$ THEN2
      20
RJ 620 PRINT"{DOWN}{BLU}PRESS:
       {RVS}SPACE{OFF} TO PAU
      SE, {RVS}RETURN{OFF} TO
       BREAK[4$]{DOWN}"
KS 630 GOSUB360:B=BS+AD-SA:FOR
      I=BTO B+7:A=PEEK(I):GOS
      UB350:GOSUB380:PRINT S$
      ;
CC 640 NEXT:PRINT"{RVS}";:A=CK
      :GOSUB350:PRINT
KH 650 F=1:AD=AD+8:IF AD>EA TH
      ENPRINT"{DOWN}{BLU}** E
      ND OF DATA **":GOTO220
KC 660 GET A$:IF A$=R$ THEN GO
      SUB1080:GOTO220
EQ 670 IF A$=S$ THEN F=F+1:GOS
      UB1080
AD 680 ONFGOTO630,660,630
CM 690 PRINT"{DOWN}{RVS} LOAD
      {SPACE}DATA ":OP=1:GOTO
      710
PC 700 PRINT"{DOWN}{RVS} SAVE
      {SPACE}FILE ":OP=0
RX 710 IN$=N$:INPUT"{DOWN}FILE
      NAME[4$]";IN$:IF IN$=N$
      {SPACE}THEN220
PR 720 F=0:PRINT"{DOWN}{BLK}
      {RVS}T{OFF}APE OR {RVS}
      D{OFF}ISK: [4$]";
FP 730 GET A$:IF A$="T"THEN PR
      INT"T{DOWN}":GOTO880
HQ 740 IF A$<>"D"THEN730
HH 750 PRINT"D{DOWN}":OPEN15,8
      ,15,"I0:":B=EA-SA:IN$="
      0:"+IN$:IF OP THEN810
SQ 760 OPEN 1,8,8,IN$+",P,W":G
      OSUB860:IF A THEN220
FJ 770 AH=INT(SA/256):AL=SA-(A
      H*256):PRINT#1,CHR$(AL)
      ;CHR$(AH);
PE 780 FOR I=0 TO B:PRINT#1,CH
      R$(PEEK(BS+I));:IF ST T
      HEN800
FC 790 NEXT:CLOSE1:CLOSE15:GOT
      O940
GS 800 GOSUB1060:PRINT"{DOWN}
      {BLK}ERROR DURING SAVE:
      [4$]":GOSUB860:GOTO220
MA 810 OPEN 1,8,8,IN$+",P,R":G
      OSUB860:IF A THEN220
GE 820 GET#1,A$,B$:AD=ASC(A$+Z
      $)+256*ASC(B$+Z$):IF AD
      <>SA THEN F=1:GOTO850
RX 830 FOR I=0 TO B:GET#1,A$:P
      OKE BS+I,ASC(A$+Z$):IF(
      I<>B)AND ST THEN F=2:AD
      =I:I=B
FA 840 NEXT:IF ST<>64 THEN F=3
FQ 850 CLOSE1:CLOSE15:ON ABS(F
      >0)+1 GOTO960,970
SA 860 INPUT#15,A,A$:IF A THEN
       CLOSE1:CLOSE15:GOSUB10
      60:PRINT"{RVS}ERROR: "A
      $
GQ 870 RETURN
EJ 880 POKE183,PEEK(FA+2):POKE
      187,PEEK(FA+3):POKE188,
      PEEK(FA+4):IFOP=0THEN92
      0
HJ 890 SYS 63466:IF(PEEK(783)A
      ND1)THEN GOSUB1060:PRIN
      T"{DOWN}{RVS} FILE NOT
      {SPACE}FOUND ":GOTO690
CS 900 AD=PEEK(829)+256*PEEK(8
      30):IF AD<>SA THEN F=1:
      GOTO970
SC 910 A=PEEK(831)+256*PEEK(83
      2)-1:F=F-2*(A<EA)-3*(A>
      EA):AD=A-AD:GOTO930
KM 920 A=SA:B=EA+1:GOSUB1010:P
      OKE780,3:SYS 63338
JF 930 A=BS:B=BS+(EA-SA)+1:GOS
      UB1010:ON OP GOTO950:SY
      S 63591
AE 940 GOSUB1080:PRINT"{BLU}**
       SAVE COMPLETED **":GOT
      O220
XP 950 POKE147,0:SYS 63562:IF
      {SPACE}ST>0 THEN970
FR 960 GOSUB1080:PRINT"{BLU}**
       LOAD COMPLETED **":GOT
      O220
DP 970 GOSUB1060:PRINT"{BLK}
      {RVS}ERROR DURING LOAD:
      {DOWN}[4$]":ON F GOSUB98
      0,990,1000:GOTO220
PP 980 PRINT"INCORRECT STARTIN
      G ADDRESS (";:GOSUB360:
      PRINT")":RETURN
GR 990 PRINT"LOAD ENDED AT ";:
      AD=SA+AD:GOSUB360:PRINT
       D$:RETURN
FD 1000 PRINT"TRUNCATED AT END
      ING ADDRESS":RETURN
RX 1010 AH=INT(A/256):AL=A-(AH
      *256):POKE193,AL:POKE1
      94,AH
FF 1020 AH=INT(B/256):AL=B-(AH
      *256):POKE174,AL:POKE1
      75,AH:RETURN
```

```
FX 1030 IF AD<SA OR AD>EA THEN
        1050
HA 1040 IF(AD>511 AND AD<40960
        )OR(AD>49151 AND AD<53
        248)THEN GOSUB1080:F=0
        :RETURN
HC 1050 GOSUB1060:PRINT"{RVS}
        {SPACE}INVALID ADDRESS
         {DOWN}{BLK}":F=1:RETU
        RN
AR 1060 POKE SD+5,31:POKE SD+6
        ,208:POKE SD,240:POKE
        {SPACE}SD+1,4:POKE SD+
        4,33
DX 1070 FOR S=1 TO 100:NEXT:GO
        TO1090
PF 1080 POKE SD+5,8:POKE SD+6,
        240:POKE SD,0:POKE SD+
        1,90:POKE SD+4,17
AC 1090 FOR S=1 TO 100:NEXT:PO
        KE SD+4,0:POKE SD,0:PO
        KE SD+1,0:RETURN
```

Program 2: MLX For Commodore 128

```
AE 100 TRAP 960:POKE 4627,128:
       DIM NL$,A(7)
XP 110 Z2=2:Z4=254:Z5=255:Z6=2
       56:Z7=127:BS=256*PEEK(4
       627):EA=65280
FB 120 BE$=CHR$(7):RT$=CHR$(13
       ):DL$=CHR$(20):SP$=CHR$
       (32):LF$=CHR$(157)
KE 130 DEF FNHB(A)=INT(A/256):
       DEF FNLB(A)=A-FNHB(A)*2
       56:DEF FNAD(A)=PEEK(A)+
       256*PEEK(A+1)
JB 140 KEY 1,"A":KEY 3,"B":KEY
        5,"C":KEY 7,"D":VOL 15
       :IF RGR(0)=5 THEN FAST
FJ 150 PRINT"{CLR}"CHR$(142);C
       HR$(8):COLOR 0,15:COLOR
        4,15:COLOR 6,15
GQ 160 PRINT TAB(12)"{RED}
       {RVS}{2 SPACES}[9 @]
       {2 SPACES}"RT$;TAB(12)"
       {RVS}{2 SPACES}{OFF}
       {BLU} 128 MLX {RED}
       {RVS}{2 SPACES}"RT$;TAB
       (12)"{RVS}{13 SPACES}
       {BLU}"
FE 170 PRINT"{2 DOWN}
       {3 SPACES}COMPUTE!'S MA
       CHINE LANGUAGE EDITOR
       {2 DOWN}"
DK 180 PRINT"{BLK}STARTING ADD
       RESS[4]";:GOSUB 260:IF
       {SPACE}AD THEN SA=AD:EL
       SE 180
FH 190 PRINT"{BLK}{2 SPACES}EN
       DING ADDRESS[4]";:GOSUB
        260:IF AD THEN EA=AD:E
       LSE 190
MF 200 PRINT"{DOWN}{BLK}CLEAR
       {SPACE}WORKSPACE [Y/N]?
       [4]":GETKEY A$:IF A$<>"
       Y" THEN 220
QH 210 PRINT"{DOWN}{BLU}WORKIN
       G...";:BANK 0:FOR A=BS
       {SPACE}TO BS+(EA-SA)+7:
       POKE A,0:NEXT A:PRINT"D
       ONE"
DC 220 PRINT TAB(10)"{DOWN}
       {BLK}{RVS} MLX COMMAND
       {SPACE}MENU [4]{DOWN}":
       PRINT TAB(13)"{RVS}E
       {OFF}NTER DATA"RT$;TAB(
       13)"{RVS}D{OFF}ISPLAY D
       ATA"RT$;TAB(13)"{RVS}L
       {OFF}OAD FILE"
HB 230 PRINT TAB(13)"{RVS}S
       {OFF}AVE FILE"RT$;TAB(1
       3)"{RVS}C{OFF}ATALOG DI
       SK"RT$;TAB(13)"{RVS}Q
       {OFF}UIT{DOWN}{BLK}"
AP 240 GETKEY A$:A=INSTR("EDLS
       CQ",A$):ON A GOTO 340,5
       50,640,650,930,940:GOSU
       B 950:GOTO 240
SX 250 PRINT"STARTING AT";:GOS
       UB 260:IF(AD<>0)OR(A$=N
       L$)THEN RETURN:ELSE 250
BG 260 A$=NL$:INPUT A$:IF LEN(
       A$)=4 THEN AD=DEC(A$)
PP 270 IF AD=0 THEN BEGIN:IF A
       $<>NL$ THEN 300:ELSE RE
       TURN:BEND
MA 280 IF AD<SA OR AD>EA THEN
       {SPACE}300
PM 290 IF AD>511 AND AD<65280
       {SPACE}THEN PRINT BE$;:
       RETURN
SQ 300 GOSUB 950:PRINT"{RVS} I
       NVALID ADDRESS {DOWN}
       {BLK}":AD=0:RETURN
RD 310 CK=FNHB(AD):CK=AD-Z4*CK
       +Z5*(CK>Z7):GOTO 330
DD 320 CK=CK*Z2+Z5*(CK>Z7)+A
AH 330 CK=CK+Z5*(CK>Z5):RETURN
QD 340 PRINT BE$;"{RVS} ENTER
       {SPACE}DATA ":GOSUB 250
       :IF A$=NL$ THEN 220
JA 350 BANK 0:PRINT:F=0:OPEN 3
       ,3
BR 360 GOSUB 310:PRINT HEX$(AD
       )+":";:IF F THEN PRINT
       {SPACE}L$:PRINT"{UP}
       {5 RIGHT}";
QA 370 FOR I=0 TO 24 STEP 3:B$
       =SP$:FOR J=1 TO 2:IF F
       {SPACE}THEN B$=MID$(L$,
       I+J,1)
PS 380 PRINT"{RVS}"B$+LF$;:IF
       {SPACE}I<24 THEN PRINT"
       {OFF}";
RC 390 GETKEY A$:IF (A$>"/" AN
       D A$<":") OR(A$>"@" AND
        A$<"G") THEN 470
AC 400 IF A$="+" THEN A$="E":G
       OTO 470
QB 410 IF A$="-" THEN A$="F":G
       OTO 470
FB 420 IF A$=RT$ AND ((I=0) AN
       D (J=1) OR F) THEN PRIN
       T B$;:J=2:NEXT:I=24:GOT
       O 480
RD 430 IF A$="{HOME}" THEN PRI
       NT B$:J=2:NEXT:I=24:NEX
       T:F=0:GOTO 360
XB 440 IF (A$="{RIGHT}") AND F
        THEN PRINT B$+LF$;:GOT
       O 470
JP 450 IF A$<>LF$ AND A$<>DL$
       {SPACE}OR ((I=0) AND (J
       =1)) THEN GOSUB 950:GOT
       O 390
PS 460 A$=LF$+SP$+LF$:PRINT B$
       +LF$;:J=2-J:IF J THEN P
       RINT LF$;:I=I-3
GB 470 PRINT A$;:NEXT J:PRINT
       {SPACE}SP$;
HA 480 NEXT I:PRINT:PRINT"{UP}
       {5 RIGHT}";:L$="
       {27 SPACES}"
DP 490 FOR I=1 TO 25 STEP 3:GE
       T#3,A$,B$:IF A$=SP$ THE
       N I=25:NEXT:CLOSE 3:GOT
       O 220
BA 500 A$=A$+B$:A=DEC(A$):MID$
       (L$,I,2)=A$:IF I<25 THE
       N GOSUB 320:A(I/3)=A:GE
       T#3,A$
AR 510 NEXT I:IF A<>CK THEN GO
       SUB 950:PRINT:PRINT"
       {RVS} ERROR: REENTER LI
       NE ":F=1:GOTO 360
DX 520 PRINT BE$:B=BS+AD-SA:FO
       R I=0 TO 7:POKE B+I,A(I
       ):NEXT I
XB 530 F=0:AD=AD+8:IF AD<=EA T
       HEN 360
CA 540 CLOSE 3:PRINT"{DOWN}
       {BLU}** END OF ENTRY **
       {BLK}{2 DOWN}":GOTO 650
MC 550 PRINT BE$;"{CLR}{DOWN}
       {RVS} DISPLAY DATA ":GO
       SUB 250:IF A$=NL$ THEN
       {SPACE}220
JF 560 BANK 0:PRINT"{DOWN}
       {BLU}PRESS: {RVS}SPACE
       {OFF} TO PAUSE, {RVS}RE
       TURN{OFF} TO BREAK[4]
       {DOWN}"
XA 570 PRINT HEX$(AD)+":";:GOS
       UB 310:B=BS+AD-SA
DJ 580 FOR I=B TO B+7:A=PEEK(I
       ):PRINT RIGHT$(HEX$(A),
       2);SP$;:GOSUB 320:NEXT
       {SPACE}I
XB 590 PRINT"{RVS}";RIGHT$(HEX
       $(CK),2)
GR 600 F=1:AD=AD+8:IF AD>EA TH
       EN PRINT"{BLU}** END OF
        DATA **":GOTO 220
EB 610 GET A$:IF A$=RT$ THEN P
       RINT BE$:GOTO 220
QK 620 IF A$=SP$ THEN F=F+1:PR
       INT BE$;
XS 630 ON F GOTO 570,610,570
RF 640 PRINT BE$"{DOWN}{RVS} L
       OAD DATA ":OP=1:GOTO 66
       0
BP 650 PRINT BE$"{DOWN}{RVS} S
       AVE FILE ":OP=0
DM 660 F=0:F$=NL$:INPUT"FILENA
       ME[4]";F$:IF F$=NL$ THE
       N 220
RF 670 PRINT"{DOWN}{BLK}{RVS}T
       {OFF}APE OR {RVS}D{OFF}
       ISK: [4]";
SQ 680 GETKEY A$:IF A$="T" THE
       N 850:ELSE IF A$<>"D" T
       HEN 680
SP 690 PRINT"DISK{DOWN}":IF OP
        THEN 760
EH 700 DOPEN#1,(F$+",P"),W:IF
       {SPACE}DS THEN A$=D$:GO
       TO 740
JH 710 BANK 0:POKE BS-2,FNLB(S
       A):POKE BS-1,FNHB(SA):P
       RINT"SAVING ";F$:PRINT
MC 720 FOR A=BS-2 TO BS+EA-SA:
       PRINT#1,CHR$(PEEK(A));:
       IF ST THEN A$="DISK WRI
       TE ERROR":GOTO 750
GC 730 NEXT A:CLOSE 1:PRINT"
       {BLU}** SAVE COMPLETED
       {SPACE}WITHOUT ERRORS *
       *":GOTO 220
RA 740 IF DS=63 THEN BEGIN:CLO
       SE 1:INPUT"{BLK}REPLACE
        EXISTING FILE [Y/N][4]
       ";A$:IF A$="Y" THEN SCR
       ATCH(F$):PRINT:GOTO 700
       :ELSE PRINT"{BLK}":GOTO
        660:BEND
GA 750 CLOSE 1:GOSUB 950:PRINT
       "{BLK}{RVS} ERROR DURIN
       G SAVE: [4]":PRINT A$:G
       OTO 220
FD 760 DOPEN#1,(F$+",P"):IF DS
        THEN A$=DS$:F=4:CLOSE
       {SPACE}1:GOTO 790
```

```
PX 770 GET#1,A$,B$:CLOSE 1:AD=
       ASC(A$)+256*ASC(B$):IF
       {SPACE}AD<>SA THEN F=1:
       GOTO 790
KB 780 PRINT"LOADING ";F$:PRIN
       T:BLOAD(F$),B0,P(BS):AD
       =SA+FNAD(174)-BS-1:F=-2
       *(AD<EA)-3*(AD>EA)
RQ 790 IF F THEN 800:ELSE PRIN
       T"{BLU}** LOAD COMPLETE
       D WITHOUT ERRORS **":GO
       TO 220
ER 800 GOSUB 950:PRINT"{BLK}
       {RVS} ERROR DURING LOAD
       : {4}":ON F GOSUB 810,8
       20,830,840:GOTO220
QJ 810 PRINT"INCORRECT STARTIN
       G ADDRESS (";HEX$(AD);"
       )":RETURN
DP 820 PRINT"LOAD ENDED AT ";H
       EX$(AD):RETURN
EB 830 PRINT"TRUNCATED AT ENDI
       NG ADDRESS ("HEX$(EA)")
       ":RETURN
FP 840 PRINT"DISK ERROR ";A$:R
       ETURN
KS 850 PRINT"TAPE":AD=POINTER(
       F$):BANK 1:A=PEEK(AD):A
       L=PEEK(AD+1):AH=PEEK(AD
       +2)
XX 860 BANK 15:SYS DEC("FF68")
       ,0,1:SYS DEC("FFBA"),1,
       1,0:SYS DEC("FFBD"),A,A
       L,AH:SYS DEC("FF90"),12
       8:IF OP THEN 890
FG 870 PRINT:A=SA:B=EA+1:GOSUB
        920:SYS DEC("E919"),3:
       PRINT"SAVING ";F$
AB 880 A=BS:B=BS+(EA-SA)+1:GOS
       UB 920:SYS DEC("EA18"):
       PRINT"{DOWN}{BLU}** TAP
       E SAVE COMPLETED **":GO
       TO 220
CP 890 SYS DEC("E99A"):PRINT:I
       F PEEK(2816)=5 THEN GOS
       UB 950:PRINT"{DOWN}
       {BLK}{RVS} FILE NOT FOU
       ND ":GOTO 220
GQ 900 PRINT"LOADING ...{DOWN}
       ":AD=FNAD(2817):IF AD<>
       SA THEN F=1:GOTO 800:EL
       SE AD=FNAD(2819)-1:F=-2
       *(AD<EA)-3*(AD>EA)
JD 910 A=BS:B=BS+(EA-SA)+1:GOS
       UB 920:SYS DEC("E9FB"):
       IF ST>0 THEN 800:ELSE 7
       90
XB 920 POKE193,FNLB(A):POKE194
       ,FNHB(A):POKE 174,FNLB(
       B):POKE 175,FNHB(B):RET
       URN
CP 930 CATALOG:PRINT"{DOWN}
       {BLU}** PRESS ANY KEY F
       OR MENU **":GETKEY A$:G
       OTO 220
MM 940 PRINT BE$"{RVS} QUIT
       {4}";RT$;"ARE YOU SURE
       {SPACE}[Y/N]?":GETKEY A
       $:IF A$<>"Y" THEN 220:E
       LSE PRINT"{CLR}":BANK 1
       5:END
JE 950 SOUND 1,500,10:RETURN
AF 960 IF ER=14 AND EL=260 THE
       N RESUME 300
MK 970 IF ER=14 AND EL=500 THE
       N RESUME NEXT
KJ 980 IF ER=4 AND EL=780 THEN
        F=4:A$=DS$:RESUME 800
DQ 990 IF ER=30 THEN RESUME:EL
       SE PRINT ERR$(ER);" ERR
       OR IN LINE";EL
```

MLX Machine Language Entry Program For Apple

Tim Victor, Editorial Programmer

To make it easier to enter machine language programs into your computer without typos, COMPUTE! is introducing its "MLX" entry program for the Apple II series. It's our best MLX yet. It runs on the II, II+, IIe, and IIc, and with either DOS 3.3 or ProDOS.

A machine language (ML) program is usually listed as a long series of numbers. It's hard to keep your place and even harder to avoid making mistakes as you type in the listing, since an incorrect line looks almost identical to a correct one. To make error-free entry easier, COMPUTE! generally lists ML programs for Commodore and Atari computers in a format designed to be typed in with a utility called "MLX." The MLX program uses a checksum system to catch typing errors almost as soon as they happen.

Apple MLX checks your typing on a line-by-line basis. It won't let you enter invalid characters or let you continue if there's a mistake in a line. It won't even let you enter a line or digit out of sequence. Best of all, you don't have to know anything about machine language to enter ML programs with MLX. Apple MLX makes typing ML programs almost foolproof.

Using Apple MLX

Type in and save some copies of Apple MLX on disk (you'll want to use MLX to enter future ML programs in COMPUTE!). It doesn't matter whether you type it in on a disk formatted for DOS 3.3 or ProDOS. Programs entered with Apple MLX, however, must be saved to a disk formatted with the same operating system as Apple MLX itself.

If you have an Apple IIe or IIc, make sure that the key marked CAPS LOCK is in the down position. Type RUN. You'll be asked for the starting and ending addresses of the ML program. These values vary for each program, so they're given at the beginning of the ML program listing and in the program's accompanying article. Find them and type them in.

Invalid Characters Banned

Apple MLX is fairly flexible about how you type in the numbers. You can put extra spaces between numbers or leave the spaces out entirely, compressing a line into 18 keypresses. Be careful not to put a space between two digits in the middle of a number. Apple MLX will read two single-digit numbers instead of one two-digit number (F 6 means F and 6, not F6).

You can't enter an invalid character with Apple MLX. Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), nothing happens. This safeguards against entering extraneous characters. Even better, Apple MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, Apple MLX will catch your mistake.

The next thing you'll see is a menu asking you to select a function. The first is (E)NTER DATA. If you're just starting to type in a program, pick this. Press the E key, and the program asks for the address where you want to begin entering data. Type the first number in the first line of the program listing if you're just starting, or the line number where you left off if you've already typed in part of a program. Hit the RETURN key and begin entering the data.

Once you're in Enter mode, Apple MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight bytes and a checksum. When you enter a line and hit RETURN, Apple MLX recalculates the checksum from the eight bytes and the address. If you enter more or less than nine numbers, or the checksum doesn't exactly match, Apple MLX erases the line you just entered and prompts you again for the same line.

Apple MLX also checks to make sure you're typing in the right line. The address (the number to the left of the colon) is part of the checksum recalculation. If you accidentally skip a line and try to enter incorrect values, Apple MLX won't let you continue. Just make sure you enter the correct starting address; if you don't, you won't be able to enter any of the following lines. Apple MLX will stop you.

Editing Features

Apple MLX also includes some editing features. The left- and right-arrow keys allow you to back up and go forward on the line that you are entering, so you can retype data. Pressing the CONTROL (CTRL) and D keys at the same time (*delete*) removes the character under the

cursor, shortening the line by one character. Pressing CTRL-I (*insert*) puts a space under the cursor and shifts the rest of the line to the right, making the line one character longer. If the cursor is at the right end of the line, neither CTRL-D nor CTRL-I has any effect.

When you've entered the entire listing (up to the ending address that you specified earlier), Apple MLX automatically leaves Enter mode and redisplays the functions menu. If you want to leave Enter mode before then, press the RETURN key when Apple MLX prompts you with a new line address. (For instance, you may want to leave Enter mode to enter a program listing in more than one sitting; see below.)

Display Data

The second menu choice, (D)ISPLAY DATA, examines memory and shows the contents in the same format as the program listing. You can use it to check your work or to see how far you've gotten. When you press D, Apple MLX asks you for a starting address. Type in the address of the first line you want to see and hit RETURN. Apple MLX displays program lines until you press any key or until it reaches the end of the program.

Save And Load

Two more menu selections let you save programs on disk and load them back into the computer. These are (S)AVE FILE and (L)OAD FILE. When you press S or L, Apple MLX asks you for the filename. The first time you save an ML program, the name you assign will be the program's filename on the disk. If you press L and specify a filename that doesn't exist on the disk, you'll see a disk error message.

If you're not sure why a disk error has occurred, check the drive. Make sure there's a formatted disk in the drive and that it was formatted by the same operating system you're using for Apple MLX (ProDOS or DOS 3.3). If you're trying to save a file and see an error message, the disk might be full. Either save the file on another disk or quit Apple MLX (by pressing the Q key), delete an old file or two, then run Apple MLX again. Your typing should still be safe in memory.

Apple MLX: Machine Language Entry Program

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
8F 100 N = 9: HOME : NORMAL : PR
       INT "APPLE MLX": POKE 34,
       2: ONERR GOTO 610
CC 110 VTAB 1: HTAB 20: PRINT "S
       TART ADDRESS";: GOSUB 530
       : IF A = 0 THEN PRINT CHR
       $ (7): GOTO 110
8C 120 S = A
E3 130 VTAB 2: HTAB 20: PRINT "E
       ND ADDRESS ";: GOSUB 530
       : IF S > = A OR A = 0 THE
       N PRINT CHR$ (7): GOTO 13
       0
20 140 E = A
B5 150 PRINT : PRINT "CHOOSE:(E)
       NTER DATA";: HTAB 22: PRI
       NT "(D)ISPLAY DATA": HTAB
        8: PRINT "(L)OAD FILE  (
       S)AVE FILE  (Q)UIT": PRIN
       T
AE 160 GET A$: FOR I = 1 TO 5: I
       F A$ < > MID$ ("EDLSQ",I,
       1) THEN NEXT : GOTO 160
93 170 ON I GOTO 270,220,180,200
       : POKE 34,0: END
AF 180 INPUT "FILENAME: ";A$: IF
        A$ < > "" THEN PRINT CHR
       $ (4);"BLOAD";A$;",A";S
A1 190 GOTO 150
38 200 INPUT "FILENAME: ";A$: IF
        A$ < > "" THEN PRINT CHR
       $ (4);"BSAVE";A$;",A";S;"
       ,L";(E - S) + 1
92 210 GOTO 150
C2 220 GOSUB 590: IF B = 0 THEN
       150
9E 230 FOR B = B TO E STEP 8:L =
       4:A = B: GOSUB 580: PRIN
       T A$;": ";:L = 2
B5 240 FOR F = 0 TO 7:V(F + 1) =
        PEEK (B + F): NEXT : GOS
       UB 560:V(9) = C
F2 250 FOR F = 1 TO N:A = V(F):
       GOSUB 580: PRINT A$" ";:
       NEXT : PRINT : IF PEEK (4
       9152) < 128 THEN NEXT
94 260 POKE 49168,0: GOTO 150
CC 270 GOSUB 590: IF B = 0 THEN
       150
48 280 FOR B = B TO E STEP 8
A6 290 HTAB 1:A = B:L = 4: GOSUB
        580: PRINT A$;": ";: CAL
       L 64668:A$ = "":P = 0: GO
       SUB 330: IF L = 0 THEN 15
       0
F9 300 GOSUB 470: IF F < > N THE
       N PRINT CHR$ (7);: GOTO 2
       90
27 310 IF N = 9 THEN GOSUB 560:
       IF C < > V(9) THEN PRINT
       CHR$ (7);: GOTO 290
72 320 FOR F = 1 TO 8: POKE B +
       F - 1,V(F): NEXT : PRINT
       : NEXT : GOTO 150
8E 330 IF LEN (A$) = 33 THEN A$
       = 0$:P = 0: PRINT CHR$ (7
       );
22 340 L = LEN (A$):0$ = A$:0 =
       P:L$ = "": IF P > 0 THEN
       L$ = LEFT$ (A$,P)
E9 350 R$ = "": IF P < L - 1 THE
       N R$ = RIGHT$ (A$,L - P -
       1)
55 360 HTAB 7: PRINT L$;: FLASH
       : IF P < L THEN PRINT MID
       $ (A$,P + 1,1);: NORMAL :
        PRINT R$;
78 370 PRINT " ";: NORMAL
E6 380 K = PEEK (49152): IF K <
       128 THEN 380
C1 390 POKE 49168,0:K = K - 128
5B 400 IF K = 13 THEN HTAB 7: PR
       INT A$;" ";: RETURN
8A 410 IF K = 32 OR K > 47 AND K
       < 58 OR K > 64 AND K < 7
       1 THEN A$ = L$ + CHR$ (K)
        + R$:P = P + 1
C1 420 IF K = 4 THEN A$ = L$ + R
       $
9F 430 IF K = 9 THEN A$ = L$ + "
        " + MID$ (A$,P + 1,1) +
       R$
8A 440 IF K = 8 THEN P = P - (P
       > 0)
93 450 IF K = 21 THEN P = P + (P
       < L)
9D 460 GOTO 330
37 470 F = 1:D = 0: FOR P = 1 TO
        LEN (A$):C$ = MID$ (A$,P
       ,1): IF F > N AND C$ < >
       " " THEN RETURN
8D 480 IF C$ < > " " THEN GOSUB
       520:V(F) = J + 16 * (D =
       1) * V(F):D = D + 1
5F 490 IF D > 0 AND C$ = " " OR
       D = 2 THEN D = 0:F = F +
       1
08 500 NEXT : IF D = 0 THEN F =
       F - 1
17 510 RETURN
B5 520 J = ASC (C$):J = J - 48 -
        7 * (J > 64): RETURN
AB 530 A = 0: INPUT A$:A$ = LEFT
       $ (A$,4): IF LEN (A$) = 0
        THEN RETURN
6F 540 FOR P = 1 TO LEN (A$):C$
       = MID$ (A$,P,1): IF C$ <
       "0" OR C$ > "9" AND C$ <
       "A" OR C$ > "Z" THEN A =
       0: RETURN
2D 550 GOSUB 520:A = A * 16 + J:
        NEXT : RETURN
28 560 C = INT (B / 256):C = B -
        254 * C - 255 * (C > 127
       ):C = C - 255 * (C > 255)
2B 570 FOR F = 1 TO 8:C = C * 2
       - 255 * (C > 127) + V(F):
       C = C - 255 * (C > 255):
       NEXT : RETURN
DA 580 I = FRE (0):A$ = "": FOR
       I = 1 TO L:T = INT (A / 1
       6):A$ = MID$ ("0123456789
       ABCDEF",A - 16 * T + 1,1)
        + A$:A = T: NEXT : RETUR
       N
1F 590 PRINT "FROM ADDRESS ";: G
       OSUB 530: IF S > A OR E <
        A OR A = 0 THEN B = 0: R
       ETURN
0D 600 B = S + 8 * INT ((A - S)
       / 8): RETURN
B6 610 PRINT "DISK ERROR": GOTO
       150
```

©

Classified

SOFTWARE

COMMODORE: TRY BEFORE YOU BUY. Best selling games, utilities, educational, + classics and new releases. 100's of titles. Visa/MC. Free brochure. RENT-A-DISC, Frederick Bldg. #345, Hunt'n, WV 25701 (304) 529-3232

FREE APPLE SOFTWARE
Over 1000 Public Domain Programs on 50 diskettes. $5 each plus $1 for shipping per order. Send $1 for catalog. Refundable with order.
C&H ENTERPRISES
PO Box 29243, Memphis, TN 38127

TI-99/4A QUALITY SOFTWARE for Business, Home, Entertainment **Bonus Software Offer!** Send for FREE catalog to MICRO-BIZ HAWAII, BOX 1108, PEARL CITY, HI 96782

TI-99/4A Software/Hardware bargains. Hard-to-find items. Huge selection. Fast service. Free catalog. DYNAMIC, Box 690, Hicksville, NY 11801

DISCOUNT SOFTWARE for most computers. FREE CATALOG. Sale: 5.25" DSDD Disks 25 for $13.95 ppd. WMJ DATA SYSTEMS-C, 4 Butterfly Dr., Hauppauge, NY 11788

CHEAP SOFTWARE FOR PC/MS-DOS/PCjr... Games, Business, Educational and Utility Disk. For catalog write: Morning Star, P.O. Box 3095, Ann Arbor, MI 48106

FREE! PUBLIC DOMAIN SOFTWARE! MS-DOS, IBM & Compatibles - Save $$$ @ $3.50 per disk! Free flyer: AP-JP, Inc, P.O. Box 1155, W. Babylon, NY 11704

FREE SOFTWARE for C64, C128, IBM & CPM send SASE for info (specify computer) to: PUBLIC DOMAIN USERS GROUP PO Box 1442-A1, Orange Park, FL 32067

MILLION $ LOTTO BUSTER PROGRAM GUARANTEED. SCIENTIFIC. HITS MILLIONS. AUTHOR HITS WEEKLY. NOT A R/N GEN. Z-WAY, P.O. BOX 9017, CANTON, OH 44711

$$ WIN $$
with our
SPORTS
Handicapping
Software

THOROUGHBRED/HARNESS Handicapping. $29.95, enhanced, $49.95. **GREYHOUND** Handicapping. $29.95, enhanced. $49.95. **PRO FOOTBALL** Handicapping System. $39.95. Specify disk/tape. Apple II+/e, Atari, IBM PC, COM 64/128, TI, TRS-80 Mod 3/4, Mod 100/200. Color. Add $2 post/handl. MC/VISA/CODs accepted. Free information. **SOFTWARE EXCHANGE**, Box 5382 CP, W. Bloomfield, MI 48033. Orders 1-800-527-9467

Quality **IBM SOFTWARE** from **$1** per disk. Games/WProc/DBases/Educ/Sprsht/Util/More. Public Domain-Latest Versions! Free Catalog. SOF-TO-GO, Box 2737, W. Lafayette, IN 47906 (317) 497-3301 CALL OR WRITE TODAY!

HOME & BUSINESS
Savgs/Loans, Cost Schedule, Calculator Files: List, Search, Create, Read, Add Data. Calclg Files: Charge Acct, Auto, Budget/Inventory, Check & Bank Stmt/Bal Tally/Exps, Phone, Payroll, TEXT-FILE, Invoice Files: Sales, Service, Pymts, Rtns. Editing, Sorts, Help Access, Instr Manual, Programmers...List Code. 256K Min. IBM/Compatible. COD/MO/CK: $39.95 + $5 s/h, APT RENTAL Files Incl'd: $44.95 + $5 s/h. Calc-Data, Inc. CD1.0(C) & ARI.0(C) Tamarind Dr., Hallandale, FL 33009 Orders (800) 247-7893 (305) 456-0417

T.I.99/4A OWNERS
1-800-USA-994A
Tech Help Catalogs
Newsletters
IN FLORIDA 305-962-8846

IBM PUBLIC DOMAIN SOFTWARE **$3** PER DISK. Send for free list. We have dbases/games/spreadsheets/finance/educational/and more. For home or business. Disks are new DSDD. JDX/C P.O. Box 1561, Corona, CA 91718

HARDWARE

IBM-APPLE
Compatibility Card
Now run Apple programs on all MS-DOS systems including IBM, clones and Tandy systems
Free Brochure
1-800-872-9942 In Fla: 305-962-8846

MISCELLANEOUS

SAFEWARE INSURES COMPUTERS against fire, theft, & power surges for as little as $39. Call Safeware, The Insurance Agency Inc. at 800/848-3469, Columbus, Ohio.

BBS Numbers $5 BBS Software! Order by modem (300 BAUD) 818-840-8066 or send $5 to BBS-FUNPAK, Box 6055, Burbank, CA 91510 Multi-User Modem Party Line: 818-842-3322

ATARI ST OWNERS! Padded pro soft carry bag by top manuf. w/pockets, handls, strap. For keybd & dd: $74.95, for monitor: $55. Add $4 s/h. 10% off with $17 membshp to USA ST USERS GROUP. Includes monthly newsletter, prod discounts, free softw. USA ST UG, 10 Cornwall St., Boston, MA 02130

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.) Inquire about display rates.
Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted. Make checks payable to COMPUTE! Publications.
Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.
General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Ad will appear in next available issue after receipt.
Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.
Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

COMPUTE! Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine
P.O. Box 10954
Des Moines, IA 50340

or call the Toll Free number listed below.

Change Of Address. Please allow us 6–8 weeks to effect the change; send your current mailing label along with your new address.

Renewal. Should you wish to renew your **COMPUTE!** subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

New Subscription. A one year (12 month) US subscription to **COMPUTE!** is $24.00 (2 years, $45.00; 3 years, $65.00. For subscription rates outside the US, see staff page). Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of **COMPUTE!**, if you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!
1-800-247-5470
In IA 1-800-532-1272

Advertisers Index

THE MORE YOU USE COMPUTE!, THE MORE VALUABLE YOUR COMPUTER BECOMES...

COMPUTE! magazine maximizes the value of your home computer by providing you with powerful do-it-yourself programs for games, household management, business and school! Your computer will be more practical, more fun, more valuable to you and your family. Take advantage of Compute! at our introductory rate.

☐ 1 Year (12 Issues) **$24** ☐ 2 Years (24 Issues) **$45**

Name _____

Address _____

City _____ State _____ Zip _____

☐ Payment Enclosed. ☐ Please Bill Me.

J6717

To Order Call: 1-800-547-5470.

For Foreign & Canadian Subscribers, please add $6 (U.S.) per year postage.

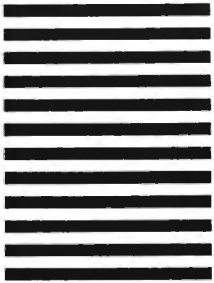
Offer subject to change without notice.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7478 DES MOINES, IA

POSTAGE WILL BE PAID BY ADDRESSEE



COMPUTE!

P.O. Box 10955
Des Moines, IA 50347-0955

COMPUTE!'s Disk

SPECIAL INTRODUCTORY OFFER

YES! I want to save time and money. Please enter my quarterly subscription to the following COMPUTE! Disk:

☐ Commodore ☐ Apple ☐ Atari ☐ IBM

☐ Save 33% off the single issue price. 1 year subscription, $39.95

☐ Save even more! 2 year subscription, $69.95

☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa

Or Call: 1-800-247-5470

Acct. No. _____ Exp. Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

Save 33%

(Outside U.S. and Canada, add $9.00 per year for shipping and handling.)

J67S5

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7551 DES MOINES, IA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!'s DISK
P.O. BOX 10036
DES MOINES, IA 50347-0036

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

Circle 101 for a one year new U.S. subscription to COMPUTE!; you will be billed for $24.

Please let us know. Do you own: plan to buy:

- ☐ 270 Apple _____ ☐ 271
- ☐ 272 Atari _____ ☐ 273
- ☐ 274 Commodore _____ ☐ 275
- ☐ 276 IBM _____ ☐ 277
- ☐ 278 TI-99/4A _____ ☐ 279
- ☐ 280 Other _____ (specify model) ☐ 281

Please print or type name and address.
Limit one card per person.

Name _____

Address _____

City _____

State/Province _____ Zip _____

Country _____

Phone _____

Please include ZIP Code

Expiration Date 8/31/87 CO687

SUBSCRIBE TO COMPUTE!

For Fastest Service, Call Our **Toll-Free** US Order Line
800-247-5470

☐ $24.00 One Year US Subscription
☐ $45.00 Two Year US Subscription

Name _____

Address _____

City _____ State _____ Zip _____

☐ Payment Enclosed ☐ Bill me
Charge my: ☐ VISA ☐ MasterCard ☐ American Express
Account No. _____ Expires ____/____

Your subscription will begin with the next available issue. Please allow 4-6 weeks for delivery of first issue. Subscription prices subject to change at any time. Outside the U.S.A. please add $6 for each subscription.

J6741

Place
Stamp
Here

COMPUTE! Reader Service
P.O. Box 2141
Radnor, PA 19089

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 7478 DES MOINES, IA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!

P.O. Box 10955
Des Moines, IA 50347-0955

Amiga screen

SO REAL IT HURTS

GFL Championship Football™

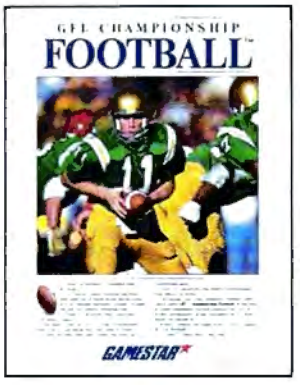
The way computer football *should* be.

Other football games put you in the grandstands, looking down on the action. Now see what it's like from the *player's* perspective—looking out of your helmet at an angry linebacker headed straight for you, and no blockers in sight.

With **GFL Championship Football,**™ you've got the first football simulation that actually takes you down on the field, taking the hits and making the plays. And it's more than just a pretty picture—you really get the feel of *playing* football.

No other football simulation gives you so many features:

- *In-the-helmet perspective* puts you at ground level on the playing field.
- *Scrolling-screen animation* moves you up and down the playing field.
- *Realistic sound effects* let you hear everything from the quarterback calling the signals to the sound of your own footsteps.
- *Team selection screens* allow you to set the playing style of your team and that of your opponent.



Whether you're taking on bone-crunching action against a friend, or going up against any of the 27 computer-controlled teams in the **GFL**, this is the one that puts you where the action is!

Available now for the Commodore 64/128, IBM PC and Tandy 1000, Apple IIe, IIc, Amiga, Atari ST and 100% compatible computers.

Look for Activision products at your local software dealer.
Or you can buy by mail at suggested list price by calling 1-800-227-6900.

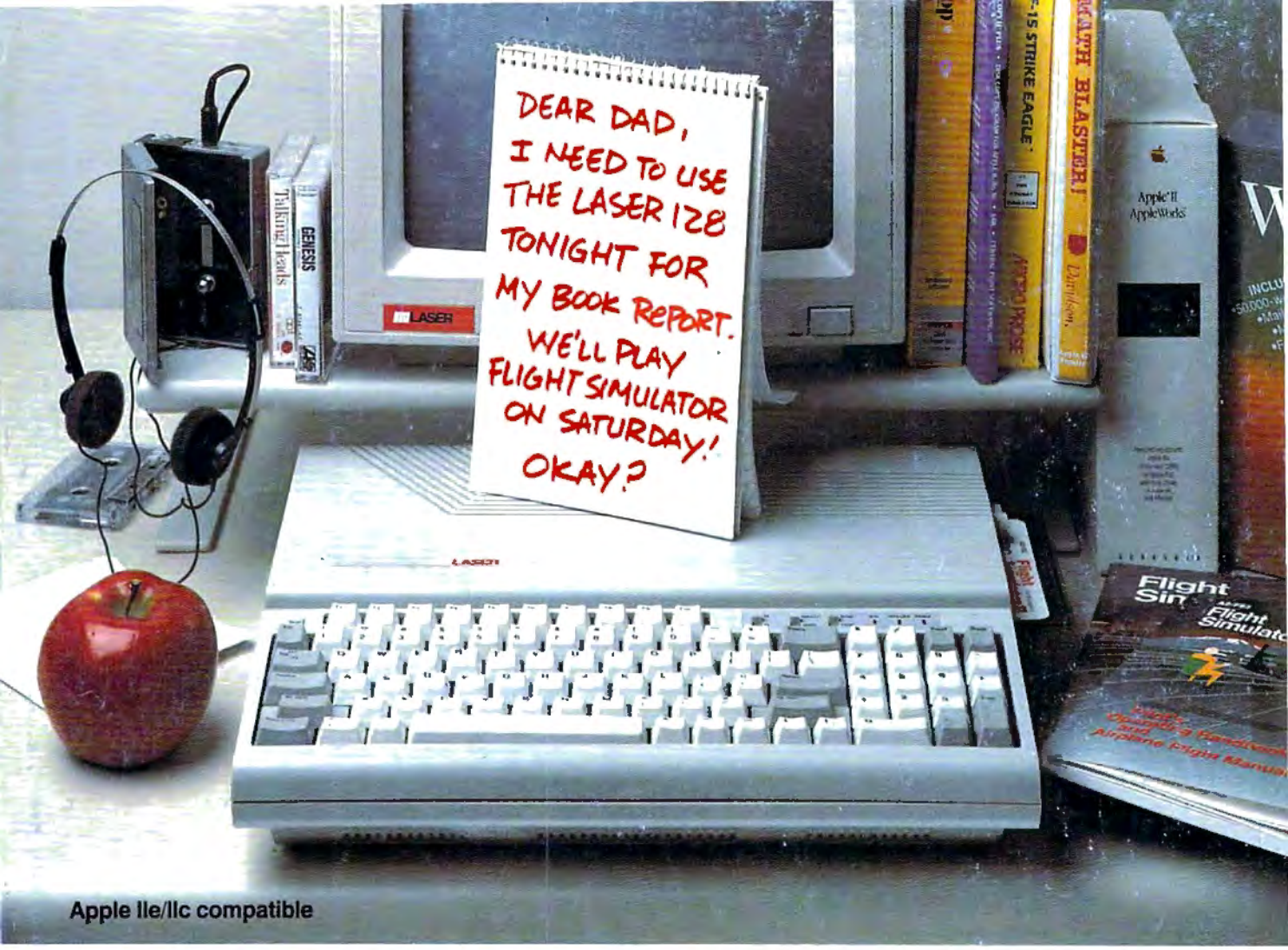
Amiga screen

Commodore 64-128 screen

Commodore 64-128 screen

Commodore 64-128 screen

GAMESTAR



Apple IIe/IIc compatible

Now your kids can afford to do their homework

More and more students are learning with computers. However most parents haven't been able to work a computer into their budget. The Laser 128 Apple-compatible computer will let you do all those things that you and your family want to or have to do – homework, write reports, even play games for a fraction of the cost of an Apple. With a Laser you can work out your budget on a computer, instead of breaking it with one. The Laser lets you take advantage of the largest software library available, so your child can learn more at home with the same programs they learn on in school. And, you can do your work at home on the Laser, too.

The Laser 128 with all its features: built-in disk drive; 128K RAM (expandable to 1 megabyte); serial, parallel, modem and mouse interfaces; 80 column text mode; numeric keypad; and an expansion slot; makes for a pretty educated buy. When you do your homework on which computer to buy, you'll find the Laser 128 at the head of the class with value. For more information on the Laser 128 and the name of your nearest dealer, contact Video Technology Computers, Inc., 400 Anthony Trail, Northbrook, IL 60062, or call (312) 272-6760.

Apple, Apple IIe and Apple IIc are registered trademarks of Apple Computer, Inc.

LASER 128

Manufactured by
VIDEO TECHNOLOGY COMPUTERS, INC.

MAKING COMPUTERS AFFORDABLE