68000: The Powerhouse Behind The Macintosh, Atari ST, & Amiga

COMPUTE!

$2.95
February
1986
Issue 69
Vol. 8, No. 2
$3.75 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

Apple SpeedCalc
A Powerful Spreadsheet
Programs Inside For II+, IIe, IIc

High Rise
Exceptional Arcade Game
For Commodore 64, 128,
Atari, And Apple

Commodore
Speedy Strings
A New Technique
For Fast-Loading Data

ST Doodler
A Drawing Program
For 520ST Logo

IBM Screen Swapping
Instant Animation
With Page-Flipping

High-Speed Search
For Atari BASIC

Introduction To
AmigaDOS, Part 2

Here's what you can do with your Apple II.

INTRODUCING

Discovery Software from World Book™

When discovery occurs, learning begins.

Choose great software gifts from World Book—long a trusted name in educational information. Discovery Software from World Book reinforces your child's basic education in continually entertaining program formats.

Each program in Discovery Software from World Book has been tested and evaluated by educators, consultants, and children. Included with each program is a unique Discovery Workbook of activities and projects which you can use to strengthen learning skills presented in the software.

Give your child the gift of educational discovery. Give Discovery Software from World Book. See your local software dealer or call 1-800-292-9090. (In Ohio call 1-800-423-7755).

Great Gifts for your child

Settling America
A survival simulation program featuring
- practice in making value judgements
- practice in making decisions
- helps expand knowledge of post-Revolutionary America

Spell Bound
A critical thinking skills review program featuring
- vocabulary reinforcement through use of verbal analogies
- four levels of difficulty

WhizCalc I
An arithmetic skills review program featuring
- practice of basic arithmetic operations
- nine levels of difficulty
- a "Create Your Own Problems" option

Data Hurdles
A data use skills review program featuring
- ten activities that review ordering, rounding, sets, totals, fast math, time, money, percentages, measuring, and comparing
- on-screen tutorial skill reviews
- three levels of difficulty

Fast Break
A punctuation skills review program featuring
- explanation of commonly used punctuation marks
- practice in using punctuation marks in context
- four levels of difficulty

Run for President
A social studies program featuring
- review of geography facts about the United States
- a review of U.S. state facts

Educational software that understands the learning process

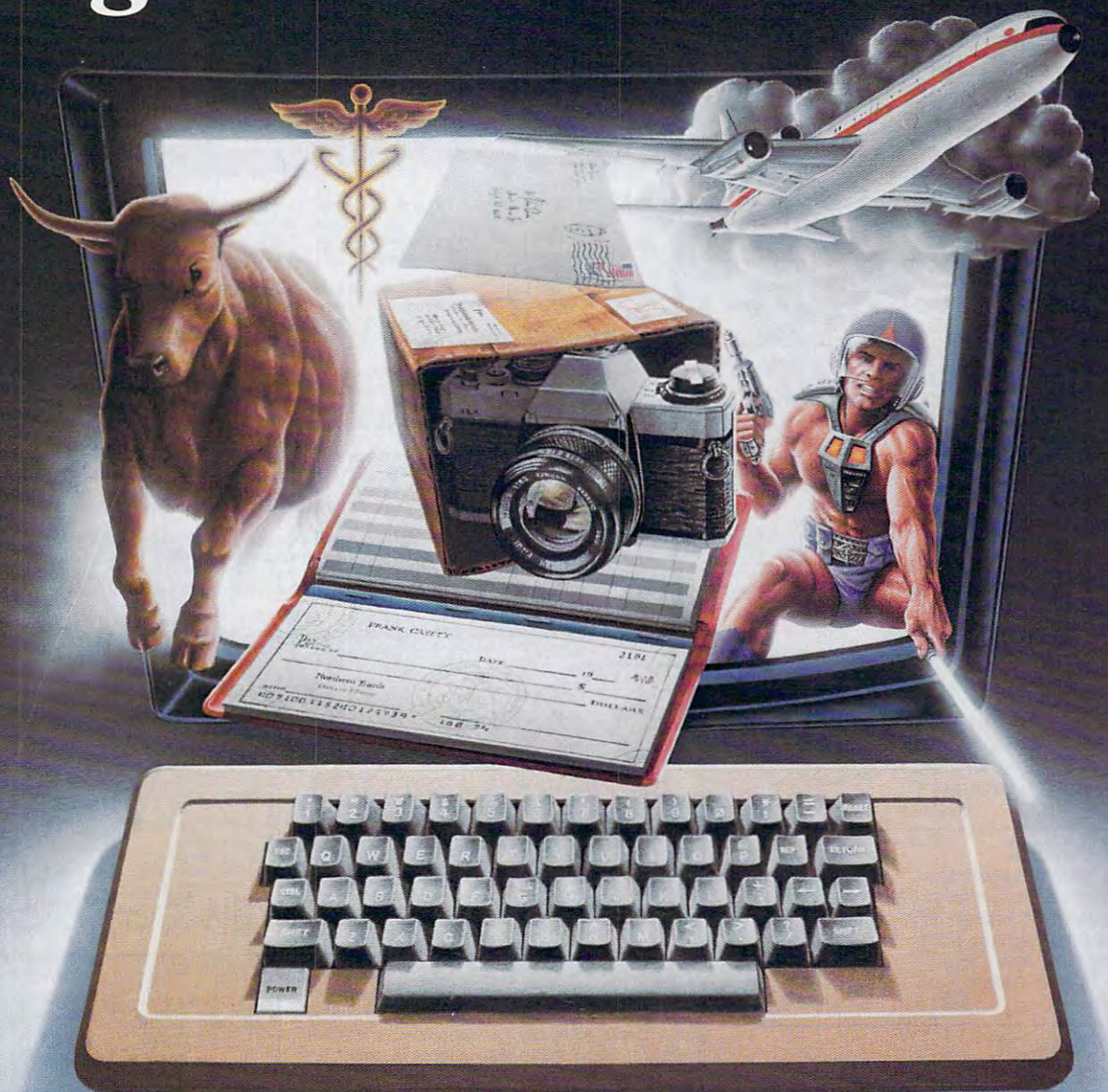
Discovery Software from World Book
For Apple® IIe/IIc
- Six programs now available (Intermediate — age 10 and up)
- Seven more programs available January, 1986 (Primary — age 6-10)

For other microcomputers:
- 21 programs available now (Seven each in Preschool age 3-5, Primary, and Intermediate)

World Book Discovery, Inc.

5700 Lombardo Centre, Suite 120
Seven Hills, OH 44131
1-800-292-9090 (In Ohio 1-800-423-7755)

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



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

CompuServe is the world's largest information service designed for the personal computer user and managed by the communications professionals who provide business information services to over one quarter of the FORTUNE 500 companies.

Subscribers get a wealth of useful, profitable or just plain interesting information like national news wires, home shopping and banking, travel and sophisticated financial data. Plus electronic mail, national bulletin boards, forums (special interest groups), and a multichannel CB simulator.

You get games and entertainment, too. Board, parlor, sports, space and educational games. Trivia and the first online TV-style game show played for real prizes.

To buy a CompuServe Subscription Kit, see your nearest computer dealer. To receive our informative brochure or to order direct call or write:

CompuServe®

Consumer Information Service, P. O. Box 20212
5000 Arlington Centre Blvd., Columbus, OH 43220
800-848-8199 In Ohio Call 614-457-0802

An H&R Block Company

Trusted Software and

Language Software

ADA TRAINING COURSE

Teaches you the language of the future. Comprehensive subset of language. Includes: editor; syntax checker, compiler; assembler; disassembler, handbook. $39.95

ASSEMBLER/MONITOR

Macro assembler and extended monitor. Supports all standard functions plus floating point constants. Monitor supports bank switching, quick trace, single step, more. $39.95

BASIC COMPILER

Compiles complete BASIC language into either fast 6510 machine language and/or compact speedcode. Supports overlays and many extended BASIC commands. C-64 $39.95
C-128 $59.95

SUPER PASCAL

Full Pascal supports graphics, sprites, file management, more. Supports pointers, dynamic memory management, machine language. Compiles to fast 6510 machine code. C-64 $59.95
C-128 $59.95

FORTH LANGUAGE

Based on Forth 79 (+ parts of '83). Supports hires graphics and sound synthesizer. Full screen editor, programming tools, assembler, samples, handbook. $39.95

MASTER

Professional development package for serious applications. Indexed file system, full screen & printer management, programmer's aid, multi-precision math, royalty-free runtime, more. $39.95

VIDEO BASIC

Add 50+ graphic, sound and utility commands to your programs with this super development package. Free distribution of RUNTIME version - no royalties! $39.95

XREF-128 & XREF-64

BASIC cross-reference

Indispensible tool for BASIC programmers. Finds all references to variables, constants & line numbers. Sorts in alphabetical order. C-64 $17.95
C-128 $17.95

For Commodore Computers

Reference Books

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

ANATOMY OF 1541 DRIVE Best handbook on this drive, explains all. Filled with many examples programs and utilities. Fully commented 1541 ROM listings. 500+pp $19.95

MACHINE LANGUAGE FOR C-64 Learn 6510 code & write fast programs. Many samples and listings for complete assembler, monitor and simulator. 200pp $14.95

GRAPHICS BOOK FOR C-64 Best reference, covers basic and advanced graphics. Sprites, Hires, Multicolor, 3D-graphics, IRQ, CAD, projections, curves, more. 350pp $19.95

TRICKS & TIPS FOR C-64 Collection of easy-to-use techniques: advanced graphics, improved data input, enhanced BASIC, CP/M, data handling and more. 275pp $19.95

1541 REPAIR & MAINTENANCE Handbook on the drive's hardware. Includes schematics & techniques to keep 1541 running. Align drive w/ & w/o scope. Large handbook size. $19.95

ADVANCED MACHINE LANGUAGE Subjects not covered elsewhere: video controller, interrupts, timers, I/O, extensions to BASIC. Tips for the serious programmer. 210pp $14.95

PRINTER BOOK C-64/VIC-20 Understand Commodore, Epson compatible printers & 1520 plotter. Utilities, screen dump, 3D-plot, commented MPS-801 ROM listings. 330pp $19.95

SCIENCE/ENGINEERING ON C-64 In-depth introduction to computers in science. Some topics covered are chemistry, physics, astronomy, electronics & others. 350pp $19.95

CASSETTE BOOK C-64/VIC-20 Make your cassette run faster than a disk drive! Cassette data-base, disk to tape backup, tape to disk, FastTape operating system. 225pp $14.95

Productivity Tools

TECHNICAL ANALYSIS SYSTEM

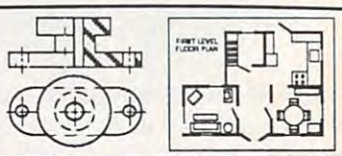
A sophisticated charting and technical analysis system for serious investors. By charting and analyzing the past history of a stock, TAS can help pinpoint trends & patterns and predict a stock's future. TAS lets you enter trading data from the keyboard or directly from online financial services. $59.95

PERSONAL PORTFOLIO MANAGER

Complete portfolio management system for the individual or professional investor. Allows investors to easily manage their portfolios, obtain up-to-the minute quotes & news, and perform selected analysis. $39.95

CADPAK

A deluxe graphics design and drawing package. Use with or without an optional lightpen to create highly-detailed designs. With dimensioning, scaling, text, rotation, object libraries, hardcopy and more. $39.95



DATAMAT

Powerful, easy-to-use data management package using menu selections. Free-form design, 50 fields/record, 2000 records/disk. Sort on multiple fields in any combination. Complete selection and formatting for printing reports. $39.95

Authoritative Books

From Abacus Software

...a name you can count on

BOOKS COVERING THE C-128

IDEAS FOR USE ON C-64 Themes: auto expenses, calculator, recipe file, stock lists, diet planner, window advertising, others. Includes all program listings. 200pp $12.95

COMPILER BOOK C-64/C-128 All you need to know about compilers: how they work, creating your own and generating the final machine code. 300pp $19.95

Adventure Gamewriter's Handbook A step-by-step guide to designing and writing your own adventure games. Adventure game generator & four example games. 200pp $14.95

PEEKS & POKES FOR THE C-64 Includes in-depth explanations of PEEK, POKE, USR, and other BASIC commands. Learn the "inside" tricks about your '64. 200pp $14.95

OPTIONAL DISKETTES FOR BOOKS For your convenience, the programs contained in each of our books are available on diskette. All program thoroughly tested & error-free. Specify title of book when ordering. **$14.95 each**

C-128 INTERNALS Detailed guide presents the 128's operating system, explains the graphics chips, Memory Management Unit, and commented listing of Kernal. 500+pp $19.95

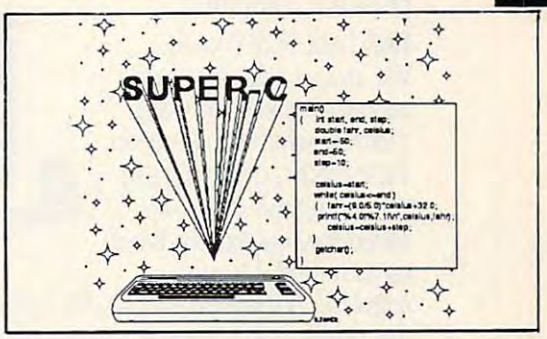
1571 INTERNALS Insiders' guide for novice and advanced users. Covers sequential & relative files, and direct access commands. Describes important DOS routines. Commented DOS listings. 500+pp $19.95

C-128 TRICKS & TIPS Chock full of info for everyone. Covers 80 column hi-res graphics, windowing, memory layout, Kernal routines, sprites and more. 300 pp $19.95

CP/M ON THE C-128 Essential guide to using CP/M on your 128. Simple explanations of the operating system, memory usage, CP/M utility programs, submit files and more. $19.95

COMPUTER AIDED DESIGN on your C-128 or 64. Create a CAD system using programs provided. Covers 3D objects & rotation, MACROS, hatching, zooming, mirroring, line widths, dashed lines, more. **300 pages** $19.95

Special Feature



For school or software development, choose **SUPER C**. **SUPER C**'s powerful screen editor is full-functioned with horizontal and vertical scrolling, copy and search/replace for easy editing. Source files may be up to 41K.

The **SUPER C** compiler is fast and creates link files. Up to seven separate modules may be linked into a ready-to-run object program. To maintain C's portability, **SUPER C** supports the Kernighan & Ritchie standard (without bit fields), making it very complete. **SUPER C** also includes a complete I/O library.

Other features of the **SUPER C** package:

- convenient hexadecimal and octal input
- error file listed to diskette
- supports conditional compiling
- complete strings and arrays
- full mathematical functions

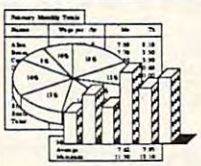
C-64 $79.95
C-128 $79.95

XPER

Capture your information on XPER's knowledge base and let this first *expert system* for Commodore computers help you make important decisions. Large capacity. Complete with editing & reporting. $59.95

POWERPLAN

One of the most powerful spreadsheets with integrated graphics for your Commodore computer. Includes menu or keyword selections, online help screens, field protection, windowing, trig functions and more. PowerGraph lets you create integrated graphs and charts from your spreadsheet data. $39.95



QUICKCOPY V2.0

Back up your valuable data with the fastest disk copier we've seen to date. Copies an entire disk in two and a half minutes on two drives or three and a half on one. $19.95

CHARTPAK

Make professional-quality pie, bar and line charts, and graphics from your data. Includes statistical functions. Accepts data from CalcResult and MultiPlan. C-128 has 3X the resolution of the C-64 version. Outputs to most printers. C-64 $39.95
C-128 $39.95

Ordering Information

Abacus Software

MasterCard VISA American Express Card

P.O Box 7211 Grand Rapids, Michigan 49510

For Postage and handling include $4.00 per order. Foreign orders include $10.00 per item. Money order and checks in U.S. Dollars only. MasterCard, VISA and American Express accected. Michigan residents please include 4% sales tax.

For *fast* service call (616) 241-5510 Telex 709-101

For **free** catalog, please return this coupon or a copy to:
Abacus Software, P.O. Box 7211, Grand Rapids, MI 49510

PHONE: (616) 241-5510

Name_____
Address_____
City_____
State_____ Zip_____

How to turn your computer on.

(The following is an actual conversation between Bantam Software and an unusually talkative personal computer).

BANTAM SOFTWARE: We always ask what turns *people* on. Now we want to know what turns *you* on.

PERSONAL COMPUTER: It's about time someone asked the real expert. What turns me off is boring software. Boring, uninvolving, predictable software. And cold rooms. Why is it always so cold in here?

B: *Games* and *Ahoy* magazines called *Sherlock Holmes in "Another Bow"* one of the year's best.

PC: Let me decide. Okay? (Disk inserted.) Well, this is anything but elementary. You're Holmes. Watson's at your side. And you determine your own fate in case after case. And look, you run into the likes of Picasso, Gertrude Stein, Henry Ford, Louis Armstrong. And such graphics! These derive from early 20th century photographs. I don't have a clue how you did it, but you have a winner. Next case.

B: *The Fourth Protocol*, from Frederick Forsyth's gigantic best-selling book. *Games* called it "nerve-tingling." Here you go. (Slides disk in.)

PC: You mean circuit-tingling. If I knew I had to save the world, I would have gotten more sleep. All kidding aside, this involves *nuclear* weapons. A British traitor. The KGB. And the subversion of NATO. This is a challenge. Will it help if I read the book? (Loud explosion on screen.) Oh no! Does that mean I lost?

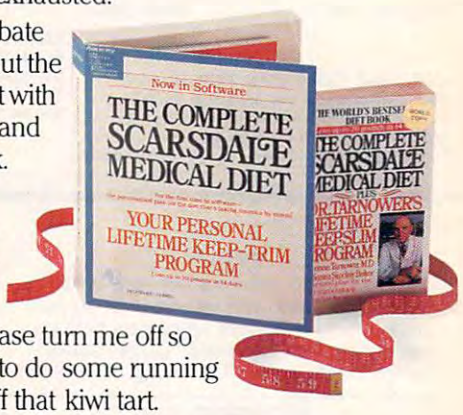
B: No, but losing's the whole point of the next one. *The Complete Scarsdale Medical Diet.* You know the bestseller.

PC: Why, do I look heavy? Never mind, let's have a taste. (Disk is inserted.) This is some menu. It helps you assess your goals. Monitor your progress. Mix 'n match meals from all five Scarsdale diets. Even prepares your shopping list. It'll tell you how much exercise you need to work off certain foods. Let's see about kiwi tart...

B: We've got one other program.

PC: No more. I'm exhausted.

B: No...this is a rebate program. Just fill out the coupon and mail it with proof of purchase and you get $5.00 back.



PC: Thank you. That's a nice offer.

B: So, did we turn you on?

PC: Yup. Now, please turn me off so I can rest. I've got to do some running later on to work off that kiwi tart.

Sherlock Holmes available for: Apple // Series, Commodore 64/128, IBM PC/PCjr, Macintosh.
Scarsdale Medical Diet available for: Apple // Series, IBM PC/PCjr.
The Fourth Protocol available for: Commodore 64/128. Available soon for Apple // Series and IBM PC/PCjr.

BANTAM ELECTRONIC PUBLISHING

Bantam Software Special Offer

$5.00 REBATE!
OFFER ENDS APRIL 15, 1986

To receive your $5.00 Rebate, just send a dated cash register receipt, the warranty card from inside the Bantam Software you bought, plus this rebate form with your name and address clearly printed.

Mail to: **Bantam Books, Inc., Dept. MT, 666 Fifth Ave., New York, NY 10103**

Name _____ Apt. # _____

Address _____

City _____ State _____ Zip _____

Terms: Purchase must be made between Jan. 15, 1986 and April 15, 1986. The warranty card and cash register receipt along with this form (or a plain piece of paper that provides all the above information) must be postmarked no later than midnight, April 30, 1986. Void where prohibited or restricted by law. This offer is not available to employees of Bantam Books, Inc. or their customers. PRINTED IN USA.

C 2/86

COMPUTE!

FEBRUARY 1986
VOLUME 8
NUMBER 2
ISSUE 69

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 114 before typing in programs.

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

AP Apple, **Mac** Macintosh, **AT** Atari, **ST,** Atari ST, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **128** Commodore 128, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **AM** Amiga. *General interest.

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019
Address all inquiries to:
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. 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

As is usual at this time of year, we begin to think about what the new year holds. This process usually gets up to speed by early December. That happens to be when we're writing this particular set of editor's notes. It also means you'll be reading them in February. It happens every year like this, but what can we say? We simply can't get too pseudo-visionary in October.

Last year this time, we were confidently predicting great things to come from Commodore. This year, we're predicting great things to come from Commodore's new Amiga just as soon as it begins to ship in quantity and . . . . You get the picture. In the editorial offices we call this hedging. It's a technique we've had to polish up on the last year or so. There was a time when this industry just grew and grew. In fact, it grew so fast that many marketing snafus, many less-than-polished products, were never recognized as such. Those times have passed. We no longer suspect that perhaps this is simply a pause in the phenomenal growth of years past. Times have truly changed, and our markets and marketeers have begun to adjust.

We won't try to offer any detailed predictions on 1986. This year, all we have are some reflections on the past, and a few on what we might expect from months ahead.

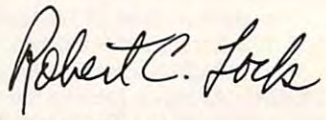
IBM's massive advertising campaign for the PCjr has been impressive for two reasons. It presents an opportunity to reap a reasonable savings on an adequately designed system, and it presents uninformed buyers with the opportunity to purchase a discontinued computer system six or more months after the announcement of its cessation. IBM does point out that it will continue to fully service and support the PCjr, and there's certainly no requirement that such merchandise be identified as no longer in production. But one must wonder whether each and every buyer who responds to this robust advertising campaign is fully aware of the transitory state of their choice of hardware.

Has Jack Tramiel done it again? This headline has been increasingly frequent of late, in part we suspect because, in this rather boring downturn in the industry, Mr. Tramiel is reliably eccentric. We have saluted his successes several times over the years, and do so again. Regardless of what the future holds for Atari, he, his sons, and their colleagues have done a remarkable, from the bootstrings up, job.

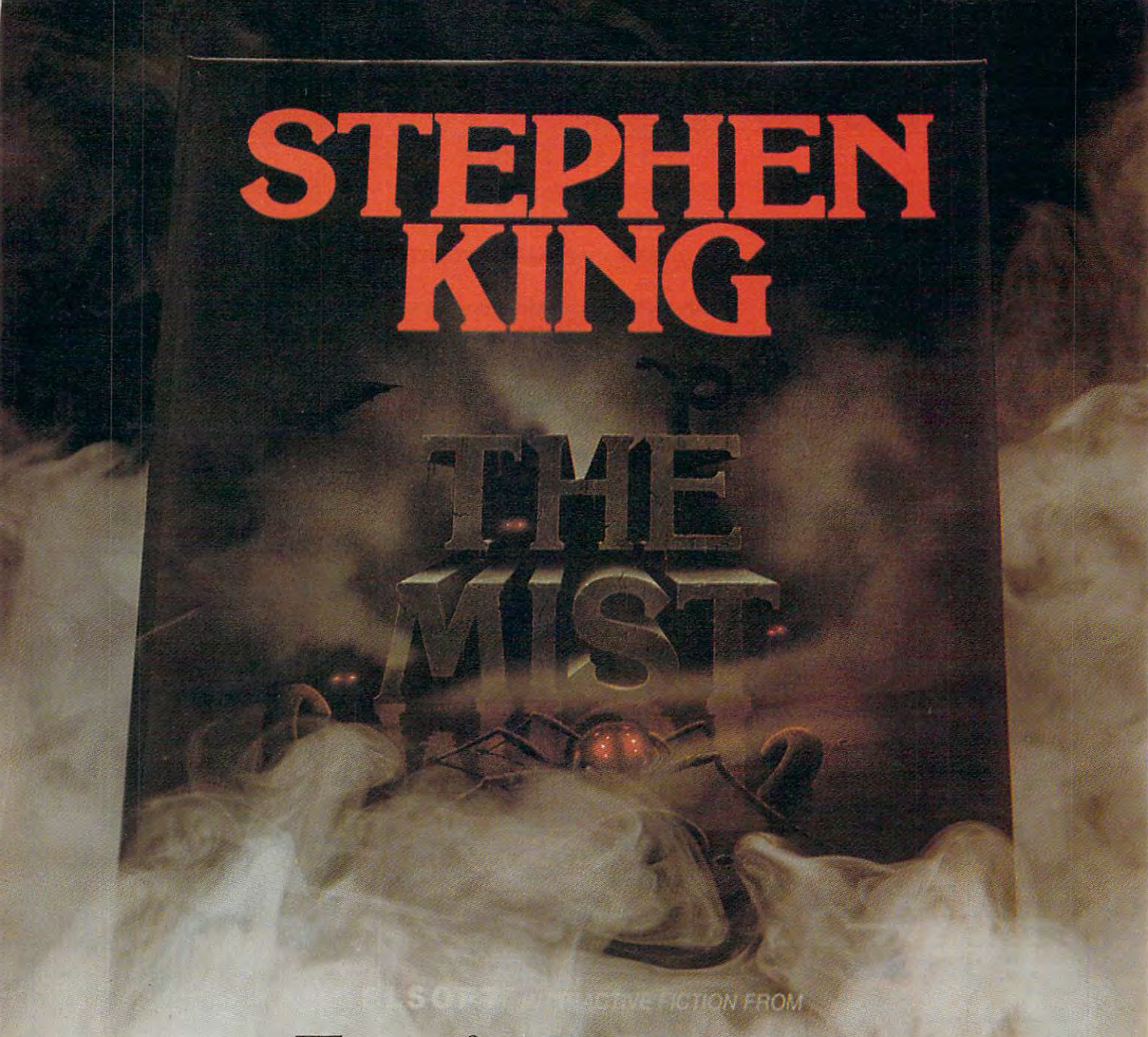
While we're on the subject of phoenix rising, the folks Mr. Tramiel left behind at Commodore haven't been doing so badly themselves. Recent news reports indicate, or at least express hope for, a profitable quarter for the Christmas season. That's one present Commodore shareholders haven't seen lately. This upturn is projected to arrive on the extended wings of the 128 and resurging sales of the 64. The Amiga has yet to begin to move in quantity, although we remain confident that it will, just as we're sure that more and more software developers will move to support it.

What else might the new year hold for us? Continuing consolidation, we're sure. Both corporate casualties and corporate successes. Everyone has become much more cautious now, so the flow of new materials will continue to diminish. Just as book publishers have become more selective about the type and quantity of titles brought to market, so, too, are the software publishers and the hardware manufacturers. Unfortunately, we can probably expect an increasing sameness, a growing presentation of products in new clothes. As the industry matures, we'll see the caution that pervades such maturation begin to inhibit previous risk-taking, so we suspect that we'll see less and less product breadth, and more and more "me-too-ness" in the market. Highly successful software will beget similar programs more rapidly. Etc. This is a kind of consolidation that markets engage in that we're not entirely comfortable with, but we're also hopeful that the rapid advances on the periphery of our technology will be sufficient to insure continued innovation. In fact, we have no doubt of it.

A belated new year to all of you, and we look forward to a pleasant 1986.



Robert C. Lock
Founder/Editor in Chief



The software program you'll play with all the lights on.

The setting is a quiet New England town. Or is it? Because when a mysterious mist descends upon it, anything can happen.

An enormous, bloodthirsty dragonfly removes heads with a snap. A mutant centipede attacks with razor-sharp legs. A giant presence paralyzes with the shriek of a curfew whistle from hell.

This could only be Stephen King's *The Mist.*™ Although it's based on the novella of the same name, you can't turn to the last page in this frightening text adventure from Mindscape.™

Your commands control the action. Those you confront along the way react to your every move.

Journey to your software dealer for Stephen King's *The Mist.*™ And while you're there, ask for Mindscape's™ other new text adventures: *James Bond 007: A View To A Kill,*™ *Forbidden Castle*™ and *Voodoo Island.*™ Each one is delightfully terrifying.

Even with the lights on.

Mindscape

Software that challenges the mind.

All of these text adventures are available for the Apple® II series and IBM PC® and PC jr. Stephen King's The Mist and James Bond 007: A View To A Kill are also available for the Macintosh.™

Mindscape, Inc. 3444 Dundee Road, Northbrook, Illinois 60062 1-800-221-9884. (In Illinois 1-800-942-7315)

Apple, Macintosh and IBM are registered trademarks of Apple Computer, Inc., and International Business Machines [illegible]

Publisher	James Casella
Founder/Editor In Chief	Robert C. Lock
Director of Administration	Alice S. Wolfe
Senior Editor	Richard Mansfield
Managing Editor	Kathleen Martinek
Editor	Tom R. Halfhill
Assistant Editor	Philip Nelson
Production Director	Tony Roberts
Production Editor	Gail Cowper
Editor, COMPUTE!'s GAZETTE	Lance Elko
Technical Editor	Ottis R. Cowper
Assistant Technical Editors	John Krause, George Miller
Program Editor	Charles Brannon
Features Editor	Selby Bateman
Assistant Editor, COMPUTE!'s GAZETTE	Todd Heimarck
Assistant Features Editor	Kathy Yakal
Programming Supervisor	Patrick Parrish
Editorial Programmers	Tim Victor, Kevin Mykytyn
Research/Copy Editor	Joan Rouleau
Copy Editor	Ann Davies
Submissions Reviewer	Mark Tuttle
Programming Assistant	David Florance
Executive Assistant	Debi Nash
Administrative Assistants	Julia Fleming, Iris Brooks, Jan Kretlow
Associate Editors	Jim Butterfield, Toronto, Canada Harvey Herman, Greensboro, NC Fred D'Ignazio, Roanoke, VA David Thornburg, Los Altos, CA
Contributing Editor	Bill Wilkinson
COMPUTE!'s Book Division Editor	Stephen Levy
Assistant Editor	Gregg Keizer
Director, Book Sales & Marketing	Steve Voyatzis
Assistant	Carol Dickerson
Production Manager	Irma Swain
Art & Design Director	Janice R. Fary
Assistant Editor, Art & Design	Lee Noel
Mechanical Art Supervisor	De Potter
Artists	Debbie Bray, Dabney Ketrow
Typesetting	Terry Cash, Carole Dunton
Illustrator	Harry Blair
Director of Advertising Sales	Ken Woodard
Production Coordinator	Patti Stokes
Administrative Assistant	Kathleen Hanlon
Promotion Assistant	Caroline Dark
Customer Service Manager	Philippa King
Dealer Sales Supervisor	Gail Jones
Assistants	Liz Krusenstjerna, Rhonda Savage
Individual Order Supervisor	Judy Taylor
Assistants	Betty Atkins, Gayle Benbow, Mary Hunt, Jenna Nash, Chris Patty
Receptionist	Anita Armfield
Warehouse Manager	Lonnie Arden
Staff	Harold Ayers, Larry O'Connor, David Hensley
Data Processing Manager	Leon Stokes
Assistants	Chris Cain, Steve Bowman
Vice President, Finance & Planning	Paul J. Megliola
Director, Finance & Planning	R. Steven Vetter
Financial Analyst	Karen K. Rogalski
Staff	Jill Pope
Credit Staff	Sybil Agee, Pat Fuller, Doris Hall, Linda Miller, Mary Waddell, Jane Wiggs

Robert G. Burton, President
Paul J. Megliola, Vice President, Finance and Planning

Member
Magazine Publishers Association
Audit Bureau of Circulations

Coming In Future Issues

***SpeedCalc* For Apple DOS 3.3 And ProDOS**

***SpeedCalc* For Atari 400/800, XL, XE**

Commodore 64 Program Profile

Analyze The Performance Of Your BASIC Programs

ST Doodler
A Drawing Program For Atari 520ST Logo

Screen Clock For IBM

Instant Apple Help Screens

Introduction To The 68000

COMPUTE! Publications, Inc. publishes:

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

Corporate office:
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408 USA
Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA
Telephone: 919-275-9809

Subscription Orders

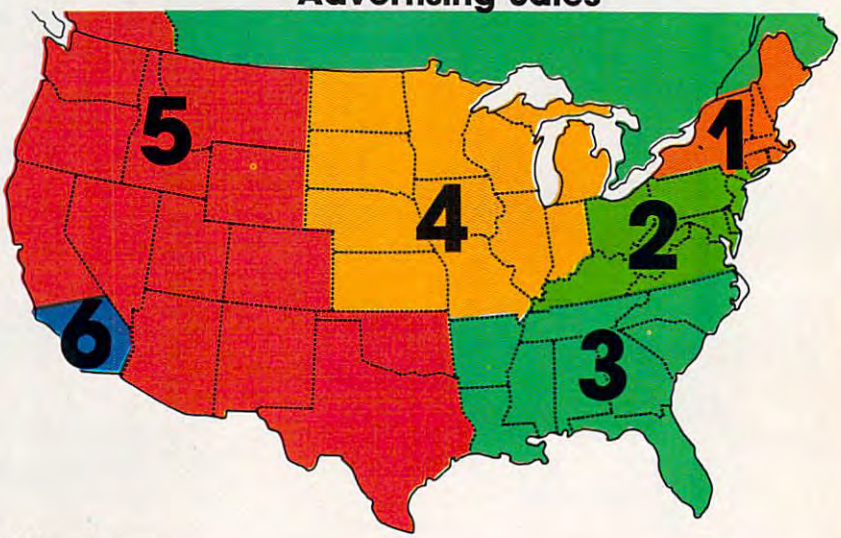
COMPUTE!
P.O. Box 10955
Des Moines, IA 50950

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

COMPUTE! Subscription Rates (12 Issue Year):

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

Advertising Sales



1. New England
Jonathan M. Just
Regional Manager
212-315-1665

2. Mid Atlantic
John Saval
Eastern Advertising Manager
212-315-1665

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

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/ Mountain/Texas
Phoebe Thompson
408-354-5553

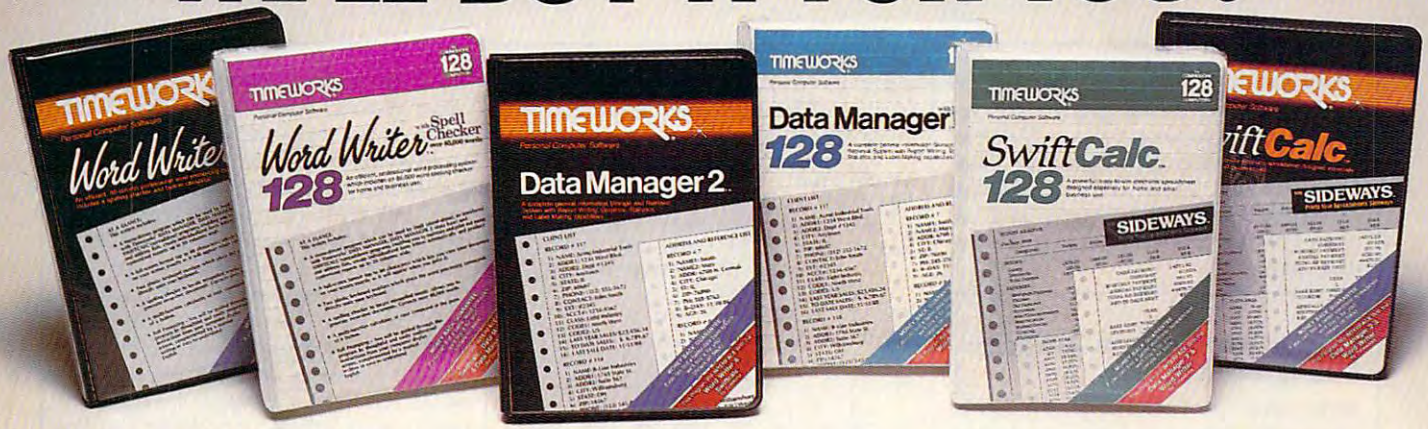
6. Southwest
Ed Winchell
213-378-8361

Director of Advertising Sales
Ken Woodard

COMPUTE! Home Office 919-275-9809.

Address all advertising materials to:
Patti W. Stokes
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue,
Greensboro, NC 27408

IF YOU CAN FIND A BETTER PROGRAM WE'LL BUY IT FOR YOU!*



WORD WRITER with Spell Checker

Now with 85,000 word Spelling Checker

- An efficient, 80-column professional word processing system which includes a spelling checker and built-in calculator.
- Contains all the features you'll need for everyday word processing, plus most of the sophisticated features found in more expensive programs: document chaining, form letter printout, page separations, horizontal and vertical scrolling, and more.

With Timeworks you get more power for your dollar

You can use each program alone. Or interface this trio – one at a time if you like – into a completely integrated productivity system that delivers all the power and features most of you will ever need . . . at a cost that lets you enjoy their use.

Look for these and other Timeworks programs at your favorite dealer. Or contact Timeworks, 444 Lake Cook Road, Deerfield, IL 60015. Phone: (312) 948-9200.

DATA MANAGER 2

Faster, more efficient, more versatile

- A complete general information storage and retrieval system with report-writing, graphics, statistics, and label-making capabilities.
- Exclusive X-SEARCH, X-SORT, and X-CHART features allow you to cross-search any category of information; sort items alphabetically, numerically, or by date; break down statistical information into categories; and graphically view your results.

With Timeworks you get more than software

You Get Our Customer Technical Support Team

At the other end of our toll-free hotline, you'll find our full-time Customer Technical Support Team. Free of charge to all registered users.

You Get Our Liberal Trade-Up Policy

You'll find the details inside each package.

SWIFTCALC with SIDEWAYS

New easy-to-use spreadsheet for home and small business use

- The SIDEWAYS option lets you print all your columns on one, continuous sheet . . . sideways.
- 250 rows and up to 250 columns (128K version) provide up to 62,500 cells (locations on the spreadsheet) in which to place information.
- Performs mathematical functions up to 17 digits. Allows the use of minimum and maximum values, averages, sums, integers, absolute values and exponential notation.

*With Timeworks you get our Money Back Guarantee

If you can find anything that works better for you, simply send us your Timeworks program, your paid receipt, and the name of the program you want, along with your check or credit card number for any retail price difference. If it's available, we'll buy it for you.**

For Apple, IBM, Commodore 128 (128K) & Commodore 64 Computers

TIMEWORKS®

More power for your dollar.

Other Timeworks Programs:

- The Evelyn Wood Dynamic Reader
- Sylvia Porter's Personal Finance Series
- Swiftax ■ Cave of the Word Wizard
- Business Systems ■ Wall Street
- The Electronic Checkbook
- The Money Manager

These programs interface with each other

Now available for Commodore 128

** Offer valid for 90 days from date of purchase.

*** Registered trademarks of Apple Computer, Inc., International Business Machines Corporation, and Commodore Computer Systems

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.

Can A Worm Have Artificial Intelligence?

Some time ago I read a description of "core war" games and their variations (*Scientific American*, March 1985). The game consists of two short programs placed in a memory area that loops around at the end. A special operating system runs the two battling programs by alternately performing one instruction from each; and the two programs attack and try to write over one another, while trying to avoid attacks and repairing themselves. Also mentioned was a "worm" program that replicates itself in a journey through the computer's memory. I am very interested in such programs and would like to see a self-replicating program for the Commodore 64.

Charles Willett

Like other eight-bit computers, the 64 isn't very well suited for conducting program wars. Most such games run on mainframe systems which are capable of multitasking—running several programs at once. Of the currently available personal computers, only the Commodore Amiga is designed for multitasking. It would require quite an elaborate machine language program to emulate even a simple form of multitasking on the 64. The battle programs must be written in machine language as well.

However, a "worm" program is quite easy to write. Following are two short examples for the 64 that reproduce themselves as they move upward in memory. By the time they're done, all of the space they've traveled through is filled with discarded copies of themselves—it's a bit like a snake shedding its skin.

The first version creates and activates an ML routine that begins at location 3000: It displays its current starting address as it goes, ending with a lockup when it hits the 64's BASIC ROM at location 40960 (turn the computer off and on to regain control):

```
10 J=3000
20 READ X:IF X<>256 THEN CK=CK
   +X:POKE J,X:J=J+1:GOTO20
30 IF CK<>9834 THEN PRINT"ERRO
   R IN DATA STATEMENTS--CHECK
    TYPING":END
40 SYS 3000
50 DATA 169,200,133,251,169,11
   ,133,252,169,247,133
60 DATA 253,169,11,133,254,160
   ,47,177,251,145,253
70 DATA 136,16,249,24,165,251,
   105,47,133,251,165
80 DATA 252,105,0,133,252,24,1
   65,253,105,47,133
90 DATA 253,165,254,105,0,133,
   254,166,251,165,252
100 DATA 32,205,189,169,13,32,
    210,255,256
```

The second version is written entirely in BASIC, but does essentially the same thing. It repeatedly copies itself to a new memory area and then runs the copy, printing its beginning and ending addresses before each move.

```
10 CLR:J=0:EA=0:SA=0:HB=0:LB=0
   :LN=0:HN=0
20 SA=PEEK(43)+256*PEEK(44):PR
   INT CHR$(147)"WORM CODE STA
   RTS AT ";SA
30 EA=PEEK(49)+256*PEEK(50):PR
   INT "AND ENDS AT ";EA
40 FOR J=0 TO (EA-SA):POKE (EA
   +J),PEEK(SA+J):NEXT
50 HN=INT(EA/256):LN=EA-(256*H
   N)
60 HB=INT((EA+(EA-SA))/256):LB
   =(EA+(EA-SA))-(256*HB)
70 PRINT CHR$(147)"POKE "EA",0
   "
80 PRINT CHR$(17)CHR$(17)"POKE
   43,"LN":POKE44,"HN
90 PRINT CHR$(17)CHR$(17)"POKE
   45,"LB":POKE46,"HB
100 PRINT CHR$(17)CHR$(17)"SYS
    42291"
110 PRINT CHR$(17)CHR$(17)"RUN
    "
120 POKE 198,6:POKE 631,19:FOR
     J=0 TO 6:POKE 632+J,13:NE
    XT
```

Locations 43–44 and 49–50 point to the beginning of program text and the end of simple variable storage, respectively. Line 40 does the actual duplication, copying everything between the starting and ending points into the addresses just above the end of the current program. Once that's done, the worm executes a series of direct mode commands with the dynamic keyboard technique to set the start-of-program and end-of-variables pointers at the right positions for the new copy. SYS 42291 relinks the program lines so they'll run properly in their new location. The BASIC worm stops with an OUT OF MEMORY message when it travels so high that there's not enough RAM left to hold its variables (your BASIC program space is almost nil at this point; type SYS 64738 to reset the computer). With some modifications, these programs will run on other Commodore computers as well.

Of course, the results here are trivial. But exercises like these can form the basis of artificial intelligence experiments. Once you begin to view a program as a "being," all sorts of intriguing questions arise: What properties in addition to movement and self-replication characterize a living entity? How can a computer emulate those actions? What is intelligence? We're only beginning to see the fruits of these inquiries in such applications as expert systems and speech recognition.

Controlling IBM's NUM LOCK

I'm trying to read the IBM PC's cursor keys from within a BASICA program. But if the NUM LOCK key is set in the wrong mode, the numeric keypad generates number codes instead of cursor codes. How can I set NUM LOCK under program control?

Dennis Heckman

When you first boot up an IBM PC, the numeric keypad keys act as cursor keys. Pressing NUM LOCK makes the computer read them as numeric keys. Memory location 1047 controls the status of NUM LOCK as well as several other special keys. Each bit of this location serves a different purpose:

Bit	Key
7	INSERT
6	CAPS LOCK
5	NUM LOCK
4	SCROLL LOCK
3	ALT
2	CTRL
1	Left SHIFT
0	Right SHIFT

In each case, a 1 in the bit position

From Origin comes the long-awaited sequel to the award-winning Ultima™ III

Ultima IV

Quest of the Avatar

Available on Apple,® Commodore™ 64

A state-of-the-art fantasy role-playing game of unprecedented magnitude by Lord British™.

Prepare yourself for a grand adventure: Ultima™ IV, sixteen times larger than Ultima III, is a milestone in computer gaming—one that challenges your physical and mental skills while testing the true fabric of your character.

Enter Britannia, kingdom of Lord British. Journey through terrain of infinite proportions, conversing with characters on hundreds of topics. Unravel the mysteries of a superior magic system. At each turn beware of daemons, dragons and long-dead wizards haunting the most tranquil of places. Encounters with parties of mixed enemy types test your strategic abilities. Shrewd use of terrain can lead to victory against seemingly impossible odds.

Survive this multi-quest fantasy, then begin the final conflict, your quest of the Avatar. The ultimate challenge—the self—awaits....

ORIGIN SYSTEMS INC. 340 HARVEY ROAD, MANCHESTER, NH 03103 (603) 644-3360

ULTIMA™ III sends you on an incredible fantasy role-playing journey through monster-plagued Sosaria in search of the elusive Exodus.

MOEBIUS™ takes you through the elemental planes of a colorful Oriental world of fantasy and adventure in search of the Orb of Celestial Harmony.

AUTODUEL™ is a futuristic, fast-paced strategy role-playing game where the right of way goes to the biggest guns.

OGRE™ is a strategy game fought on the nuclear battlefield of tomorrow as an inhuman juggernaut Cybertank battles conventional forces.

Authors wanted. Call us today.

shows that the corresponding key is pressed, and a 0 indicates that it's not pressed. Thus, after NUM LOCK has been pressed (when the keypad keys act like number keys), bit 5 of location 1047 contains a 1. When NUM LOCK is not in effect (when the keypad keys act like cursor keys), bit 5 contains a 0. To change the NUM LOCK status from within a program, all you need to do is put a 1 or 0 in bit 5 of location 1047. In most cases you'll want to leave the rest of the keyboard in its current configuration; thus, it's preferable to PEEK the current value of 1047, AND that value with 223 (to set bit 5 to 0), and POKE the resulting value back into 1047. Put the following statement at the beginning of your program to perform the entire operation:

```
10 DEF SEG=0:POKE 1047, PEEK(1047)
AND 223
```

After the computer executes line 10, the keypad keys work as cursor keys. If you need to reverse the NUM LOCK status, use the following statement to set bit 5 of location 1047 to 1:

```
DEF SEG=0:POKE 1047, PEEK(1047) OR
32
```

Of course, even after you've set the NUM LOCK status, the user can still change it back by hitting NUM LOCK. To be safe, you might want to set the status immediately before every operation requiring the keys.

Similar statements can be used to affect the other keyboard features controlled by location 1047. For instance, POKE 1047, PEEK(1047) AND 191 turns off CAPS LOCK, and POKE 1047, PEEK (1047) OR 64 turns it on.

Amiga Features

I can't wait to purchase an Amiga, but I need to know a few things. First, will your magazine support the Amiga (programs, columns, etc.)? Secondly, referring to your article about the Amiga's IBM compatibility ("Amiga Goes IBM-Compatible," October 1985), can the Amiga handle IBM PC software that must be booted on a PC? Thirdly, can the Amiga do Macintosh-type graphics and text fonts?

Victor Swindell

COMPUTE! is supporting the Amiga with product reviews and tutorials, and will add programs when the final version of BASIC is released. The first Amigas were being shipped last fall with ABasiC, a BASIC interpreter written by Meta-ComCo, the British company which also wrote AmigaDOS. However, Commodore was making final preparations in November to ship an entirely different BASIC written by Microsoft. This BASIC reportedly was adapted from Microsoft BASIC for the Macintosh and has many more features than ABasiC. The latest word we've received is that Microsoft BASIC is to replace ABasiC as the standard language shipped with the Amiga.

The IBM PC emulator software—now known as the Transformer—also was not available at this writing (early November), but demonstrations of preproduction versions show that it is capable of booting PC software directly off 5¼-inch IBM disks. (External 5¼-inch disk drives for the Amiga are optional.) Early reports indicate that an Amiga with the Transformer is not 100 percent IBM-compatible, but that it can run most of the best-selling PC software with a slight sacrifice of speed. Commodore is also working on an accelerator package that will improve the Transformer's performance. We'll report on the Transformer's degree of IBM compatibility and its speed when it becomes available.

The Amiga displays Macintosh-style graphics with its operating system interface, the Workbench, which resembles the Macintosh desktop. This screen mode has 640 × 200 resolution (128,000 pixels), less than the Mac's 512 × 384 resolution (196,608 pixels). However, the Amiga screen is in color. The Amiga also has a high-resolution mode with 640 × 400 resolution (256,000 pixels), but the Workbench doesn't work in that mode. A couple of graphics-drawing programs which are being released for the Amiga are similar to MacPaint, except they take advantage of the Amiga's palette of 4,096 colors. The Amiga also has various text fonts available like the Macintosh. You can try out some of these fonts by opening the Notepad from the Utilities drawer on the Workbench disk, then pulling down the Font and Style menus. There are several different fonts, type sizes, and styles.

Secret Apple Self-Test

While using my Apple IIc, I discovered something rather odd. I pressed CONTROL-RESET at the same time that I was unknowingly pressing one of the joystick buttons. First the screen went blank, then it filled up with colorful, constantly changing hi-res graphics patterns. I know this had nothing to do with the program I was using. What happened?

Sam Robison

On the Apple IIe and IIc, pressing a joystick button performs the same action as pressing the Open-Apple and Closed-Apple keys on the keyboard. Pressing RESET while holding down both CONTROL and the Open-Apple key forces the computer to reboot, just as if the power switch had been turned on.

Something different happens on the IIe if you also hold down the Closed-Apple key along with the other three (it takes a bit of practice to reach all four at once). The computer performs a self-test of its circuitry, which takes about 20 seconds. While this is happening, a changing lo-res pattern fills the screen. If all is well, the screen clears and the message "System OK" appears. If any other message shows up, you should have your computer checked by a technician.

The Apple IIc does not have a built-in diagnostic program. According to the Apple IIc Reference Manual, pressing the same cluster of four keys activates "Teri's Memory and Soft Switch Exercise Program," which generates the hi-res display that you saw. The program runs for a few minutes, eventually locking up the system (no harm is done—simply reboot as usual). In the Reference Manual, Apple claims that this program is used only during manufacture and has no use after that point. By accessing RAM and the display circuitry, it may generate signals that Apple's test equipment can recognize. It's also quite pretty to watch.

Commodore Boot Programs

I have a PCjr and my boss recently purchased a Commodore 64 (his first computer). I would like to know how to write the equivalent of an IBM AUTOEXEC.BAT file for the 64 so my boss can load and activate certain programs (such as "TurboDisk" and the DOS Wedge) automatically, without having to type anything. Can this be done without installing a special ROM chip?

Steve Neeland

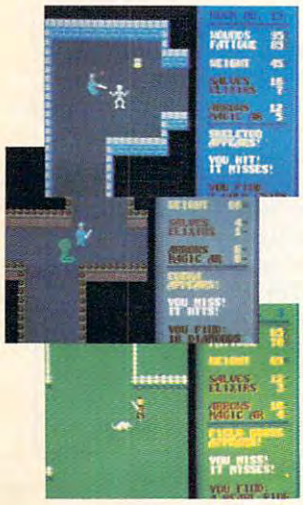
While an autoboot feature of this type is common in most computer systems, it is not built into the Commodore 64. However, it is possible to create a disk file that loads and runs automatically when you type LOAD"*",8,1 and press RETURN. Though it's too long to include here, there's a program called "Autoload" in COMPUTE!'s Third Book of Commodore 64 which creates such files for you.

If you don't mind two extra keystrokes, you can load and run any BASIC program from disk with a single command. Type the following, replacing FILENAME with the name of the BASIC program you want to run:

```
LOAD"FILENAME",8:
```

Don't forget the colon after the 8. With the cursor positioned in the space following the colon, press SHIFT-RUN/STOP. The 64 prints LOAD after the colon and proceeds to load and run the program. (On the Commodore 128, you can achieve the same effect with RUN "FILENAME".) It's also relatively easy to load and activate a series of machine language programs from BASIC, provided they return to BASIC and don't perform a NEW when they set up. Here's a short program that

TEMPLE OF APSHAI TRILOGY™
BIGGER. MEANER. AND RICHER THAN EVER.



You know Temple of Apshai.

The classic. Best-seller for over *four years.*

You may have friends trapped forever in its dark recesses.

Players have dropped from sight for weeks at a time, searching for the treasures of Apshai.

Well now we've raised the stakes.

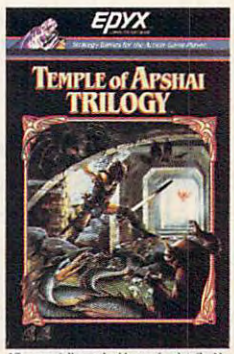
Introducing the *new* Apshai Trilogy. The *combined* wrath of the world famous Temple of Apshai®, Upper Reaches of Apshai®, and Curse of Ra®. All on a single disk. Twelve levels. 568 rooms to explore. More choices. More chances. Best of all, there's faster game play.

The graphics and sounds are new. The challenge of the dungeons is timeless.

Are you ready for the most involving role-playing game ever designed?

Temple of Apshai is waiting. Silently lurking. Patiently waiting. *For you.* At your nearest Epyx dealer.

	APPLE II	MAC	ATARI	IBM PC	C64/128
Temple of Apshai Trilogy	✓	✓	✓	✓	✓



*See specially marked boxes for details. No purchase necessary. Sweepstakes ends December 31, 1985. Official rules available at participating retailers. T-shirts available while supplies last.

EPYX
COMPUTER SOFTWARE

1043 Kiel Court, Sunnyvale, CA 94089

Strategy Games for the Action Game Player

loads the two utilities you mentioned. Be sure to save the program before you run it, since it performs NEW after installing "TurboDisk" and the DOS Wedge:

```
10 REM THIS PROGRAM ERASES
   ITSELF--SAVE BEFORE YOU RUN
20 IF Z=2 THEN 60
30 IF Z=1 THEN 50
40 Z=1:LOAD"DOS 5.1",8,1
50 Z=2:LOAD"TURBODISK.OBJ",8,1
60 SYS 49152:SYS 52224:NEW
```

Along with many other new features, the Commodore 128 has the ability to perform a true autoboot. When you turn it on, the 128 searches track 1, sector 0 of the disk in the drive for a special "signature" code consisting of the characters CBM. If that code is present, the system loads and runs the program specified in the autoboot sector. Since the autoboot program can in turn load and run a larger boot program, it's possible to create quite an elaborate boot sequence, which loads and activates your favorite utilities and otherwise configures the system exactly to your liking. Autobooting works with the 1541 disk drive (even for CP/M disks) as well as the newer 1571. COMPUTE!'s Commodore 128 Programmer's Guide *contains a detailed discussion of the autoboot process as well as a program that creates autobooting disks for the 128.*

More TI Supplies

Sorry that we weren't able to be included in your September 1985 list in this column of Texas Instruments suppliers. We're a small business and work directly with various distributors across the country. Rather than stock items, we place orders with our distributors according to our customers' needs. We offer a delivery time of two weeks in most cases.

Mary Ann Holzer
Creative Ideas
7062 South Tamarac Street
Englewood, CO 80112

Please include us on your list of companies that support the TI-99/4A computer. We have been in business since 1982 and provide software, hardware, and peripherals for the TI-99/4A as well as other computers.

Bob Polizzotto
Multi Video Services
P.O. Box 246
East Amherst, NY 14051

Thank you for the information.

Custom Cursors For 64 *SpeedScript*

Even though I have used more elaborate word processors with my Commodore 64, I frequently prefer to use *SpeedScript* because of its speed and convenience. However, I find the incessantly blinking cursor a distraction. Can you tell me how to get rid of it?

Paul Newsom

Just as everyone seems to prefer different screen colors, some people like a blinking cursor while others find it maddening. Fortunately, it's easy to stop the blink or change its speed. Of course, you wouldn't want to eliminate the cursor altogether, since that would make it hard to find your way around inside a document. To defeat the blink, load SpeedScript *into memory, type one of the following lines in direct mode (without a line number), and press RETURN. Be sure to use the correct POKEs for the version of* Speedscript *you're using, and type very carefully—even a small error may have drastic consequences:*

SpeedScript 2.0
POKE 2527,240:POKE 2528,246

SpeedScript 3.0 or *3.1* (Commodore 64 only)
POKE 2698,240:POKE 2699,246

Resave SpeedScript *under a new filename to distinguish this version from the original. Now the reverse video cursor remains steady rather than blinking. Since* SpeedScript *blinks the cursor only during idle times (when you're not pressing any keys), this has no effect on the rest of the program. To restore the blink, enter one of these lines:*

SpeedScript 2.0
POKE 2527,165:POKE 2528,162

SpeedScript 3.0 or *3.1*
POKE 2698,165:POKE 2699,162

*Changing the cursor's blink speed is even easier. To make the cursor blink at half its normal rate, enter POKE 2530,32 (*SpeedScript 2.0*) or POKE 2701,32 (*SpeedScript 3.0 *or* 3.1*). To make the cursor blink in double-time, POKE the same location with 8 instead of 32. Depending on your preferences, you may find one of these preferable to the default speed. POKE the same location with 16 to restore the normal blink rate. Because the blink is created by replacing the character under the cursor with its reverse video equivalent, there's no way to change the cursor's actual appearance without grafting a complete set of custom characters onto* SpeedScript *as well. (See "Commodore 64* SpeedScript *Fontmaker,"* COMPUTE!, *January 1986.)*

Improving Atari CLOADs

I would like to respond to James Jenkins' letter in the October 1984 issue of COMPUTE! about Atari CLOAD errors 138 and 143. Here are a few suggestions:

When purchasing blank cassettes, buy only those whose cases are held together with five screws. Tape errors are caused not so much by the quality of the tape as by the quality of the case. The Atari Program Recorders seem very susceptible to minor tape fluctuations caused by the tape binding in the case. Second, after using a tape for some time, it may become unevenly wound, causing it to bind and generate errors. To free the tape, slap it on the flat side of the cassette against a hard surface. This forces the tape against one side of the case and reduces errors. Finally, instead of pressing SYSTEM RESET to clear the screen, type GR.0 or press SHIFT-CLEAR. SYSTEM RESET can disrupt operation of the POKEY chip, which controls input/output operations. Thus, pressing SYSTEM RESET before you do a CSAVE or CLOAD can cause tape errors. To recover from this situation, type LPRINT and press RETURN while your printer (if you have one) is offline or switched off. You'll see an ERROR 138, but this simply means the printer is not responding. This resets the POKEY chip and allows error-free tape operations.

Richard L. Baldwin

Thanks for the advice. In a related letter, reader W. Byrom Dorsey points out that you can get similar information free of charge from Atari, 1265 Borregas Avenue, Sunnyvale, CA 94086. Just ask for the bulletin entitled "410 Tech Tips."

Atari Keyboard Buzzer

When I type 107 characters on my Atari 800XL, the computer sounds a buzzer. Is this a Revision B operating system bug, or does Atari have a purpose for it?

John Lapetina

The buzzer effect is a deliberate design feature, not a bug. It happens in BASIC with all Atari 400/800, XL, and XE computers with all versions of the operating system. (Incidentally, your 800XL has the XL operating system, not Revision B. Revision B fixed some bugs in the original Revision A operating system shipped with early 400s and 800s. It is available for XL and XE computers on the Atari Translator *disk.)*

The buzzer is analogous to the end-of-line bell on a typewriter: It warns when you are reaching the end of a BASIC logical line. *A logical line is the maximum number of characters that can be typed after a line number. On the Atari, a logical line may be as long as three* physical lines *(screen lines). If a BASIC statement (or series of statements separated by colons) won't fit on a logical line, you must either shorten it or break it up into two logical lines.*

The actual number of characters allowed in a logical line varies according to how the screen margins are set. Atari BASIC normally defaults to a 38-column

WINTER GAMES.™
THE QUEST FOR THE GOLD CONTINUES...

You've captured the gold in Summer Games® and Summer Games II™. Now it's on to the *Winter Games*! And what an incredible setting—a completely realistic winter wonderland featuring *seven* action-packed events.

At the Ski Jump you control your form in mid-air, knees straight, leaning forward. Hot Dog Aerials challenges your courage *and* your sense of humor. In Figure Skating you leap into Double and Triple Lutz jumps—wow the crowd with a perfect Camel into a Sit Spin. It's timing and style that counts. Free Skating lets you choreograph your own routines. In Speed Skating it's you against a fellow speed demon—the fastest human beings on level earth! And the Bobsled—still faster as you fly around hairpin turns, leaning hard to stay in the tube. Finally the Biathlon, the ultimate challenge to your endurance in cross-country skiing and marksmanship.

All of this fun and excitement is easy to learn and play. You control the action with the joystick, animating your player for style and rhythm. You choose the country you want to represent. Listen to its national anthem. Then it's practice, training and learning a winning strategy for each event. Now the *Opening Ceremony* and the competition begins—against your friends or the computer. Will you be the one who takes the gold at the *Awards Ceremony*? Will your name be etched amongst the *World Record* holders?

The quest for the gold continues... And it's all here—the strategy, the challenge, the competition, and *pageantry* of Winter Games!

	APPLE	MAC	C64/128
Winter Games	✓	✓	✓

* See specially marked boxes for details. No purchase necessary. Sweepstakes ends Dec. 31, 1985. Official rules available at participating dealers.

EPYX® COMPUTER SOFTWARE

1043 Kiel Ct., Sunnyvale, CA 94089

Strategy Games for the Action Game Player®

screen—38 characters per physical line. That means a logical line can be up to 114 characters long, and the warning buzzer sounds after 107 characters. But this can be adjusted by storing different numbers into two memory locations which control the screen margins. Location 82 sets the left margin and location 83 sets the right margin.

These locations usually contain the values 2 and 39, respectively. Atari presets the left margin at 2 because many people plug their computers into TV sets, and TVs often suffer from overscan*—they display slightly less than the entire image area on their screens. Minor overscan is rarely noticeable when watching TV shows, but it can clip off a column or two of characters along the margins when you plug in a computer. By defaulting to a 38-column screen instead of 40 columns, Atari computers automatically compensate for overscan. If your TV doesn't overscan, or if you have a computer monitor, you can reset the left margin to 40 columns with the statement POKE 82,0. This allows 40-character physical lines and 120-character logical lines. The warning buzzer won't sound until the 113th character. (SYSTEM RESET restores the default value.) On the other hand, if a TV suffers from extreme overscan, you can make the screen narrower by POKEing larger numbers into location 82 and smaller numbers into location 39. This, in turn, reduces the number of characters per physical line and logical line.*

Keep in mind, however, that you can exceed the logical line limit by using abbreviated keywords. For example, type this line exactly as shown:

```
10 GR.7:SE.4,0,0:C.1:PL.20,20:DR.40,20
:DR.40,40:DR.20,40:DR.20,20:DR.40,40
:PL.40,20:DR.20,40
```

GR. is the abbreviation for GRAPHICS, SE. stands for SETCOLOR, C. is COLOR, PL. is PLOT, and DR. is DRAWTO. Abbreviated, these commands total 92 characters and fit comfortably in a logical line of three physical lines. But when you type LIST, Atari BASIC automatically expands the abbreviations and the statement overflows into four physical lines with a total of 136 characters. Ordinarily, you couldn't type a logical line that long. If you type RUN, the program executes perfectly, so this is one way of squeezing more statements into a logical line. However, the technique should be avoided for two reasons: Other people can't type this line without also using abbreviations, and any editing which changes the length of a statement also chops off all the characters following the third physical line. ©

Get the jump on the weather-man by accurately forecasting the local weather yourself!

A scientifically proven way to develop an awesome memory.

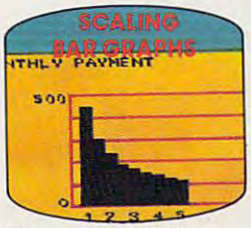
You are trapped in a five-story, 125-room structure made entirely of ice. Find the exit before you freeze!

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

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

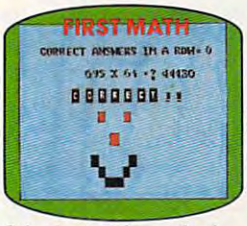
Cut your energy costs by monitoring your phone, electric and gas bills.

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



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

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



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

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

A fun way to dramatically increase typing speed and accuracy.

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

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

And when you subscribe to COMPUTE!, you'll get them all for less than 15 cents each!

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

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

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

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

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

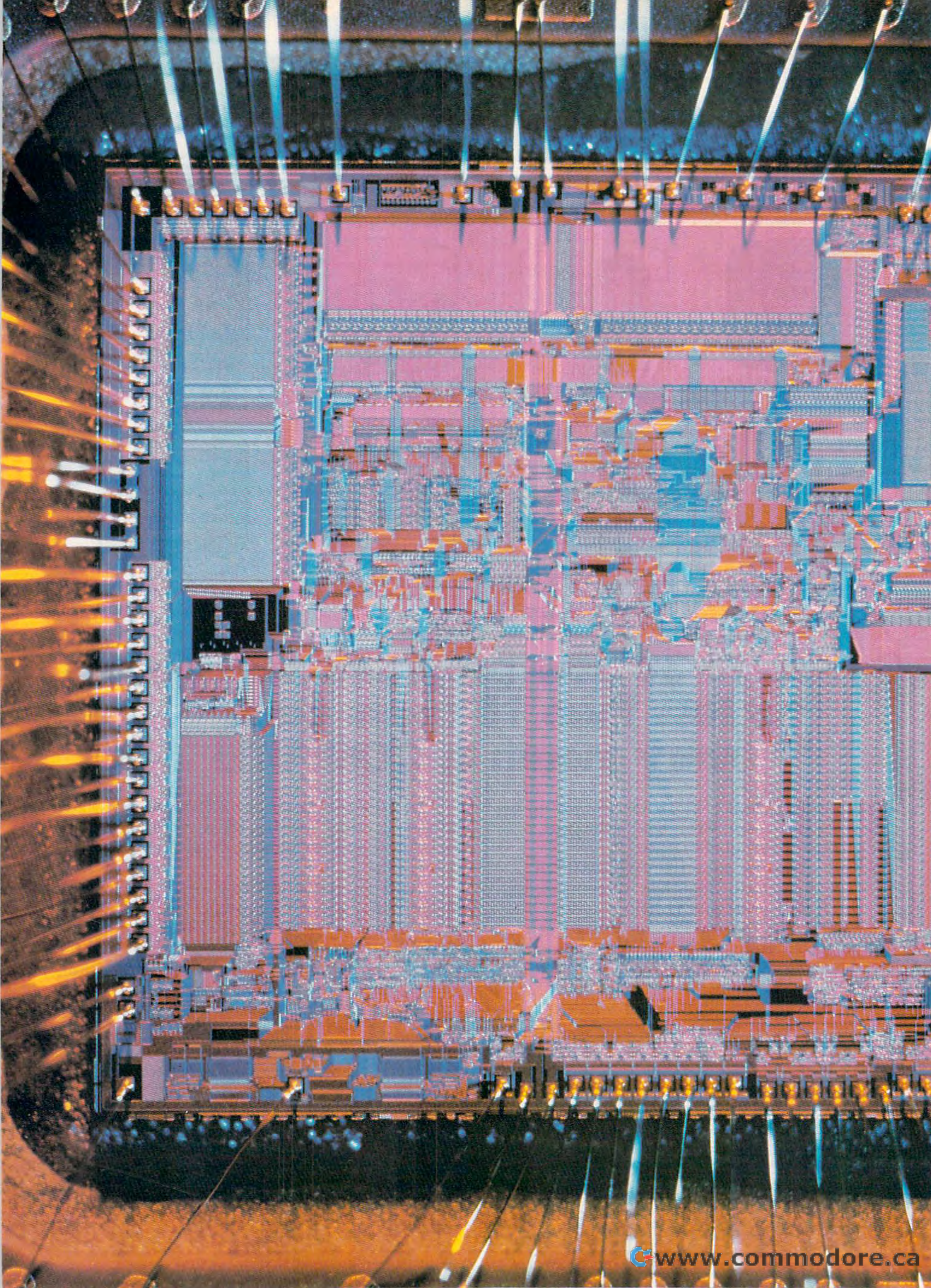
What's more, you get information-packed articles, product reviews, ideas and advice that add power and excitement to all your home computing.

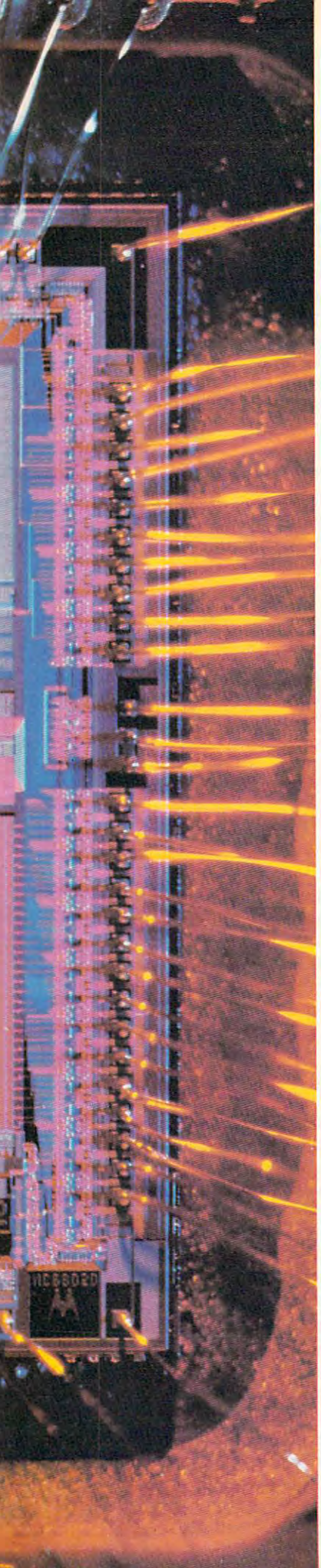
And when it's time to shop for peripherals or hardware, check COMPUTE! first. Our product evaluations can save you money and costly mistakes. We'll even help you decide what to buy: Dot-matrix or daisy-wheel printer? Tape storage or disk drive? What about modems? Memory expansion kits? What's new in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

Order now! Mail the postpaid card attached to this ad and start receiving every issue of COMPUTE!.

For Faster Service Call Toll-Free 1-800-247-5470 (in Iowa 1-800-532-1272)

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





GENEALOGY OF A CHIP

The 68000 Yesterday, Today, And Tomorrow

Selby Bateman, Features Editor

Motorola's 68000 microprocessor is the powerhouse beneath the hood of a new generation of leading-edge personal computers—the Commodore Amiga, Atari 520ST, and Apple Macintosh. With its hybrid 16/32-bit architecture and remarkable processing speed, the 68000 is helping to bring a new level of flexibility to personal computing.

8080, Z80, 6502, 8088, 68000—To a computer enthusiast, these strange numbers tell a fascinating story about the continuing development of the personal computer. They're the microprocessors that helped make millions of computers affordable to millions of people. At one time or another, these were the best brains—computer brains—our money could buy.

In the history of microelectronics, they'll be remembered as the first personal computer microprocessors to invade American homes, schools, and businesses in Apple, Atari, Commodore, IBM, and a flock of other computers. Each of these microprocessors has its own parents, cousins, and children. The family names come from companies called Intel, Zilog, MOS Technology, and Motorola.

Hidden beneath the computer's shell, a microprocessor is little more than a fingernail-sized wafer of silicon containing an intricate grid of almost microscopic transistorized circuits. But the tiny microprocessor is the central processing unit (CPU), or brain, that controls or coordinates virtually everything that goes on in the computer. It gathers instructions from the computer's memory, executes those instructions, and stores the resulting information back into memory.

A microprocessor does what its name implies—*processes* information. And that information is in the form of electrical signals. To make sense of the signals, the computer uses a binary code of ones and zeros that matches the *on* or *off* states of electricity. Each on or off position is defined as a *binary digit*, or *bit*, of data.

During the past several years, millions of people have become

acquainted with computers that have microprocessors capable of handling eight bits of information at a time. These eight-bit machines include many of the most popular computers first used in homes, schools, and small businesses. For example, the Commodore 64 is based on MOS Technology's 6510 microprocessor, a sibling of the earlier 6502 microprocessor used in the Commodore PET, VIC-20, Apple II/II+, and Atari 400/800/ XL computers. Other eight-bit 6502-compatible microprocessors are the 6502B (Apple IIe), 6502C (Atari 130XE), 65C02 (Apple IIc), 7501 (Commodore Plus/4 and 16), and the 8502 (Commodore 128). The strengths and limitations of each of these chips have helped define the nature of the computers in which they're housed.

A microprocessor CPU in today's computers is usually but one of a number of integrated circuit chips carrying on the work. But the other chips, unlike a microprocessor, are dedicated to certain functions such as memory or special support functions. The Commodore 64, for instance, also has a sophisticated programmable three-voice sound chip (the 6581 Sound Interface Device) and a versatile graphics chip (the 6567 Video Interface Chip). The Commodore 64's CPU, the 6510, coordinates the activities of these chips and others. Because the CPU itself doesn't have to carry out the duties of these support chips, it's freed to carry on its other activities without losing significant processing speed.

The latest personal computers to hit the market, such as the Amiga, advance this principle even further by making the CPU the conductor of a whole orchestra of support chips. At the same time, the latest CPU chips have grown so powerful that they're now capable of running several programs simultaneously. The newest microprocessors emerging from today's laboratories pack the power of a large mainframe computer onto a tiny chip of silicon.

(Preceding page) A microscope's view of Motorola's new 68020 microprocessor. The chip's actual size is about the diameter of an adult's little fingernail.

It was little more than a dozen years ago, in 1972, that Robert Noyce's Intel Corporation created the first functioning microprocessor—the four-bit 4004, developed by an engineer named Ted Hoff. Instead of "hard-wiring" several small chips together to accomplish certain tasks, engineers could now program the microprocessor to do a variety of operations. This four-bit chip found its most useful home in a generation of hand-held calculators.

Intel soon followed with the first eight-bit microprocessor, the 8008, and then with the 8080. The 8080A became the CPU for the first hobbyist computer, the MITS Altair, introduced as a do-it-yourself kit in 1975. Soon other companies joined in the race. Zilog introduced the eight-bit Z80, which became the CPU for a multitude of personal computers from Radio Shack, Osborne, Kaypro, Timex/Sinclair, and others. A Z80 chip is also one of the two microprocessors found in the Commodore 128. At about the same time, MOS Technology created the popular eight-bit 6502, and Motorola introduced the eight-bit 6800.

These microprocessors perform quite similarly, despite individual differences in the ways they handle such internal functions as addressing modes, memory registers, and other operations. All of them fetch, execute, and store data eight bits at a time within CPU pathways called *buses*. There are at least three basic kinds of buses in most microprocessors: a data bus, an address bus, and a control bus. The width of these buses determine whether a microprocessor is considered an eight-bit, 16-bit, or hybrid chip.

Today's microprocessors include a number of similar sections, each having specified duties. The components of a chip include an *instruction set*, simple commands that are hard-wired into the microprocessor; *registers* for storing and manipulating instructions; buses for carrying data; an internal *clock* that helps organize the flow of data; a *logic unit* for mathematical calculations; and a *decoder* that interprets the instruction set.

There are also other functioning units and subunits within every CPU, such as accumulators, status registers, interrupt functions, and program counters. And every microprocessor has its own unique style of handling data.

In 1981, IBM introduced its first personal computer, the IBM PC, which uses Intel's 8088 microprocessor (a more sophisticated descendant of the earlier 8008). The 8088 is a hybrid chip, a mixed-mode CPU that handles internal bus communications 16 bits at a time and external communications eight bits at a time. This 8/16-bit microprocessor can address, or access, up to a megabyte of memory (1,024K, or 1,048,576 bytes). In contrast, an eight-bit chip like the 6502 or Z80 can address only 64K (65,536 bytes) of memory.

In 1983, Apple introduced the Macintosh, the first popular personal computer based on the Motorola 68000, a hybrid chip even more powerful than the 8088. This 16/32-bit descendent of the 6800 processes 32 bits of information at a time on its internal buses and 16 bits in external communication. And it can directly address up to 16 megabytes of memory—16,384K, or 16,777,220 bytes.

Work began on the 68000 in the mid-1970s. Motorola formed a project team known within the company as MACSS (Motorola's Advanced Computer System on Silicon). The result of that effort was announced in 1979, when Motorola first demonstrated the new microprocessor.

In addition to the 68000's 16/32-bit processing capability, it also has a potential internal clock

All the exciting, entertaining, and educational games, applications, and utilities from COMPUTE! magazine are now available on disk

for your Commodore, Atari, Apple, or IBM personal computer.

The COMPUTE! Disk

A new *COMPUTE! Disk* is published every month, rotating among the four major machines covered by *COMPUTE!*: Commodore 64 and 128; Atari 400/800, XL, and XE; Apple II-series; and IBM PC, PCjr, and compatibles.

Every three months you can receive a disk with all the quality programs from the previous three issues of *COMPUTE!* that will run on your brand of computer.

Like the popular *COMPUTE!'s Gazette Disk*, the *COMPUTE! Disk* is ready-to-load and error-free. It saves you valuable hours of typing time and eliminates typing errors.

With a subscription, you will receive one disk every three months for a total of four disks a year—for only $39.95. That saves you $20 a year off the single-issue cost.

Or you can order individual issues of the *Disk* for $12.95 a disk plus $2.00 shipping and handling.

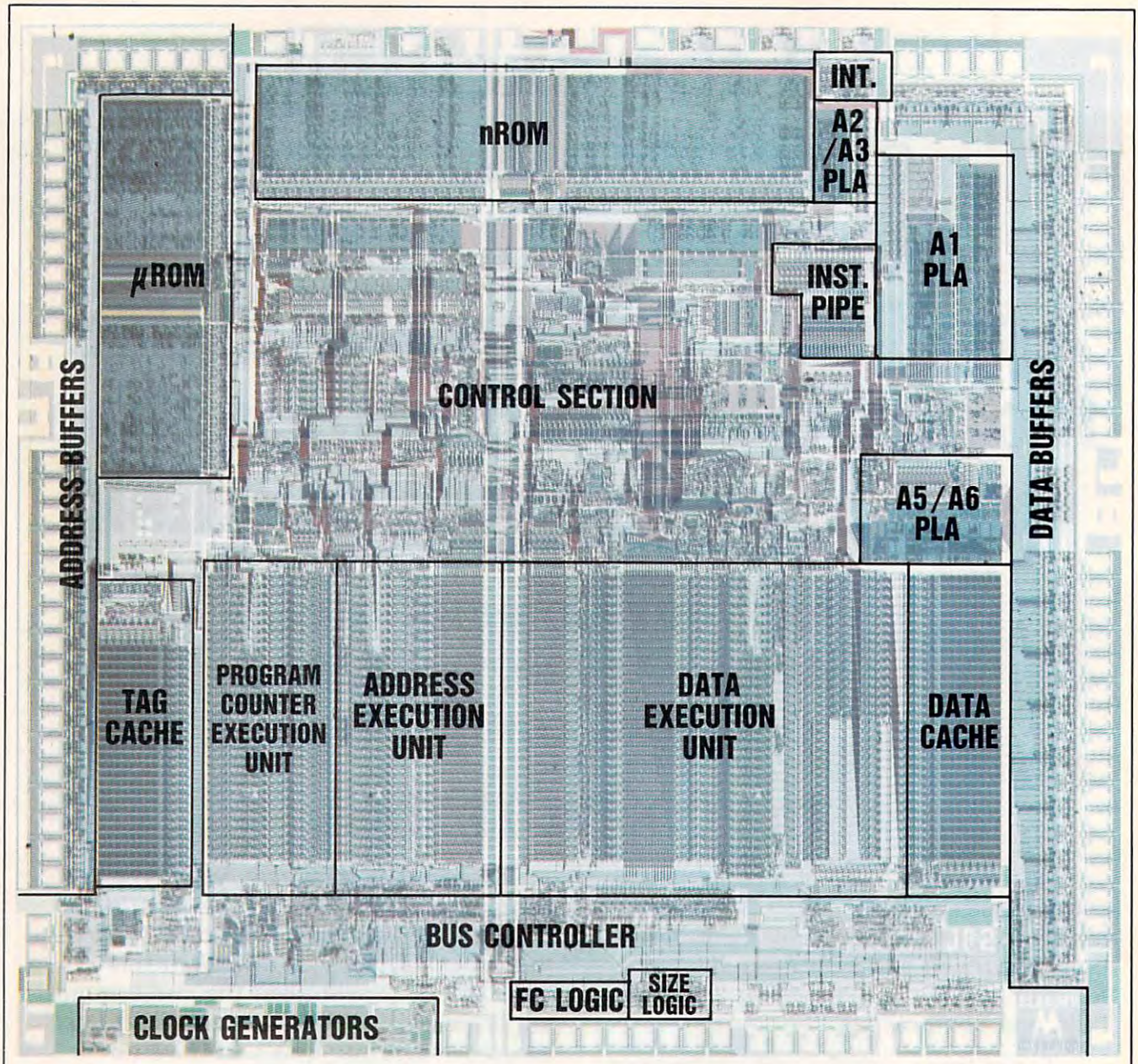
Remember to specify your type of computer when ordering the *COMPUTE! Disk*. You'll find more information about this month's *COMPUTE! Disk* in this issue. (Note: You'll need the corresponding issues of *COMPUTE!* magazine to use the *Disk* since the disk will have no documentation.)

For fastest service when ordering a subscription to the *COMPUTE! Disk*, call toll free 1-800-247-5470 (in Iowa 1-800-532-1272).

For more details or to order individual issues of the *COMPUTE! Disk*, call our Customer Service Department toll free at 1-800-346-6767 (in New York 212-887-8525).

Please allow 4–6 weeks after placing an order for your first disk to arrive.

COMPUTE! Publications, Inc. abc
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



The 68020 microprocessor contains the equivalent of 200,000 transistors. Major functions of the chip shown here include Read Only Memory microcode and control sections; programmable logic array (PLA) units; data and address buffers; clock generators; function code (FC) logic and size logic sections; program counter, address, and data execution units; interrupt logic; bus controller; and instruction pipeline.

speed of 12.5 megahertz (MHz). The clock speed is essentially how fast a chip runs, based on the number of cycles an electrical signal can make through the chip per second. Hence, 1 Hz is equivalent to one cycle per second; 1 MHz equals one million cycles per second. The higher the clock speed, the more instructions a CPU can fetch and execute per second.

Compare the 68000's 6–12.5 MHz clock speed to the 1 MHz speed of a Commodore 64, or the 4.77 MHz clock speed of an IBM PC. The current crop of 68000-based personal computers generally have clock speeds in the 7 to 8 MHz range. The Macintosh has a clock speed of 7.8 MHz, the Amiga has a clock speed of 7.16 MHz, and the Atari 520ST runs at 8 MHz. But many other factors besides raw clock speed determine a computer's overall speed, including the efficiency of its operating system and other system software.

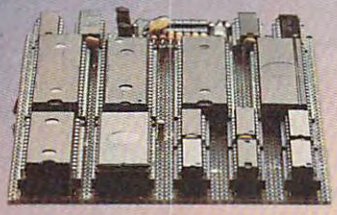
To date, the Amiga makes the most efficient use of the 68000 microprocessor because it incorporates three custom-designed VLSI (Very Large Scale Integration) chips for sophisticated support functions.

THE SHADOW $89.95

Shadow is a new and revolutionary piece of hardware that is used to duplicate even the most protected software. Fitting inside the disk drive (no soldering required), SHADOW takes complete control of all functions giving near 100% copies.

Being the best utility available today, it will even copy the other copy programs.

Because of the Shadow's unique abilities, we feel DOS protection is a thing of the past.



*HACKER PACKAGE $39.95

Shadow a disk while it loads, then read an exact list of:

- Track, sector, ID, check sum, drive status
- High and low track limits
- Density use on each track
- Half tracks that are used
- Command recorder shows commands that were sent to 1541 while program was loading
- RAM recorder records custom DOS

Shadow-scan any disk, then read exact list of:

- Valid tracks, half tracks, partial tracks and segments
- Sync mark link, header block links and data block links
- Track to track synchronization

Exclusive snap shot recorder will give you an exact copy of the 1541 RAM and can be viewed, saved or printed. Plus many more features included.

*Requires Shadow

*GT PACKAGE $44.95

Highly sophisticated and integrated piece of hardware that turns you 1541 into something you've always wanted.

- Track and sector display
- Drive reset switch
- Device number change
- Half track indicator
- Abnormal bit density indicator
- Shadow on-off indicator

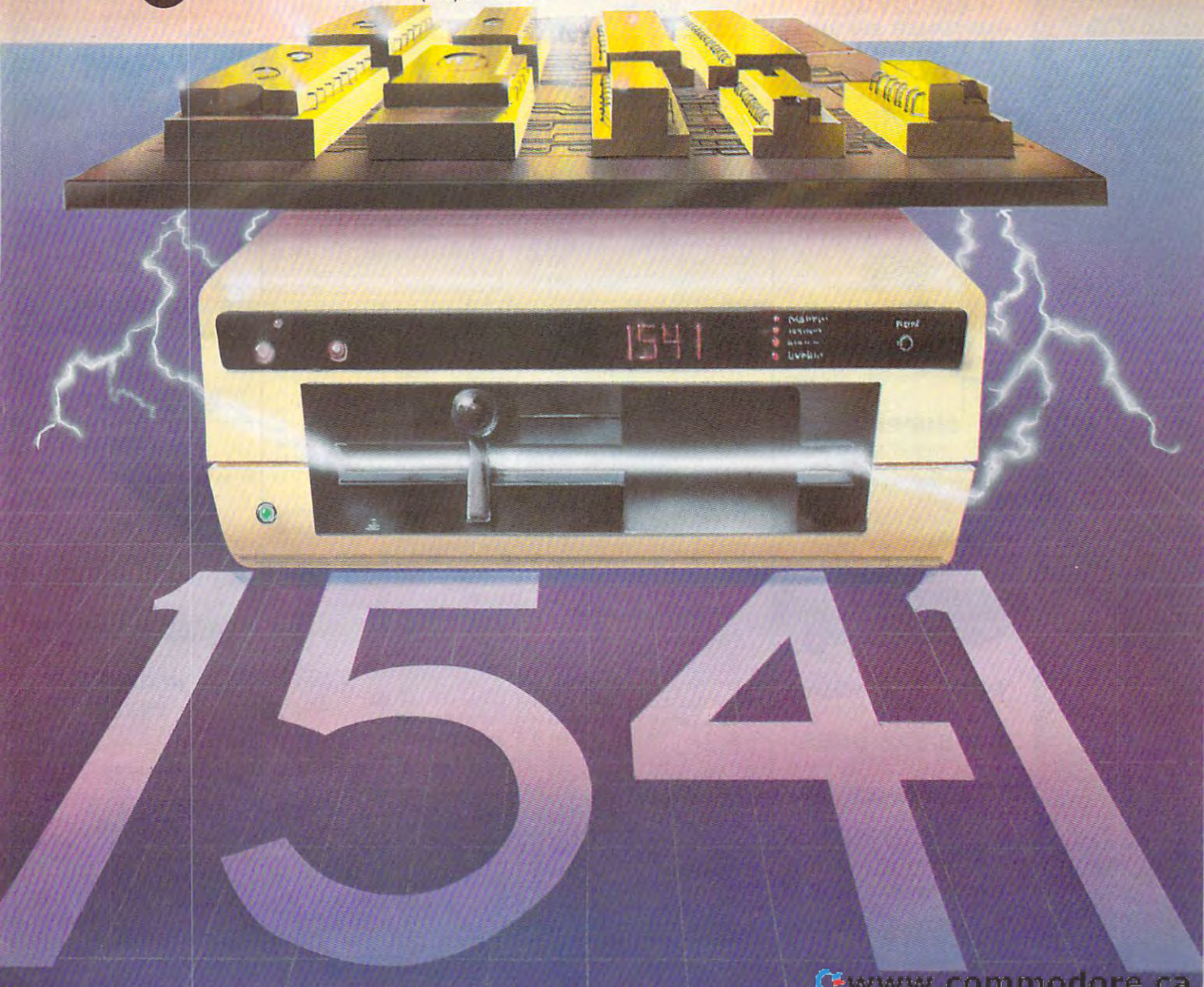
The Shadow display will give you an accurate display of precisely what track you are accessing during a normal load even if the program does a read past track 35.

*Requires Shadow

Order by phone 24 hrs./7 days or send cashier's check/money order payable to Megasoft. Visa, MasterCard include card number and expiration date. Add $3.50 shipping/handling for continental U.S., $5.50 for UPS air. CODs add $7.50, Canada add $10.00. Other foreign orders add $15.00 and remit certified U.S. funds only. Distributors invited and supported.

MegaSoft LTD

P.O. Box 1080 • Battle Ground, Washington 98604
1-800-541-1541
Canadian/Foreign Orders Call
(206) 687-5205



COMPUTER MAIL ORDER

THE CMO ADVANTAGE

- THE BEST PRICES!
- Next day shipping on all in stock items.
- Free easy access order inquiry.
- Orders from outside Pennsylvania save state sales tax.
- Free technicial support with our factory trained technical staff.
- There is no limit and no deposit on C.O.D. orders.
- There's no extra charge for using your credit card. Your card is not charged until we ship.
- No waiting period for cashiers checks.
- We accept purchase orders from qualified corporations. Subject to approval.
- Educational discounts available to qualified institutions.
- FREE CATALOG MEMBERSHIP.

ORDER LINE

CALL TOLL-FREE
1-800-233-8950
Educational Institutions Call Toll-Free
1-800-221-4283
CUSTOMER SERVICE & TECH SUPPORT
1-717-327-1450
Dept. A202

MAILING ADDRESS

Computer Mail Order
Dept. A202
477 East Third Street
Williamsport, PA 17701

MEMBER DIRECT MARKETING ASSOCIATION

CREDIT CARDS

VISA MasterCard American Express Diners Club

SHIPPING

Add 3%, minimum $5.00 shipping and handling on all orders. Larger shipments may require additional charges.

All items subject to availability and price change.

Returned shipments may be subject to a restocking fee.

CANADIAN ORDERS

1-800-268-3974
Ontario/Quebec
1-800-268-4559
Other Provinces
1-416-828-0866
In Toronto
TELEX: 06-218960

2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U.S.A. orders.
Call The Canadian Office for Canadian prices.

HOME COMPUTERS

ATARI

130XE (128K)	**CALL**
520ST (512K)	**CALL**
800XL 64K	CALL
1010 Recorder	$49.99
1050 Disk Drive	CALL
1027 Letter Quality Printer	$129.00
1030 Direct Connect Modem	$59.99

Software Specials

8036 Atari Writer	$24.99
Star Raiders	$4.99
Missile Command	$4.99
Defender	$4.99
Galaxian	$4.99
Asteroids	$4.99
Centipede	$4.99
Miner 2049'er	$4.99
Eastern Front	$4.99
SynCalc	$19.99
SynFile	$19.99
VisiCalc	$39.99

ATARI 520-ST SOFTWARE

SIERRA ON LINE

Ultima II	$39.99
Gato	$29.99

INFOCOM

Zork I, II, III	(ea.) $29.99
Hitchiker's Guide	$29.99
Wishbringer	$29.99
Suspended	$37.99

HABA

Hippo-C	$49.99

MIRAGE CONCEPTS

Express	$34.99
ST-Toolbox	CALL

MARK OF THE UNICORN

Final Word	$94.99
Hex	$29.99
PC Intercom	$89.99

MINDSCAPE

Halley Project	$34.99

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
MacINTOSH	CALL
IIc LCD Display	CALL

HAYDEN

Art Grabber	$31.99
Home Design	$49.99
Media Works	$63.99

PALADIN

Crunch 512	$189.00

MacIntosh Software

Lotus Jazz	CALL
Microsoft Excel	$259.00
Living Videotext ThinkTank 512	$159.00
Manhatten Ready, Set, Go	$79.99
Creighton Development Mac Spell	$69.99
Monogram Dollars & Sense	$99.99
Peachtree Back to Basics - GL	$109.00
PFS File & Report (New Version)	$119.00
Silicon Beach Airborn	$25.99

commodore

C128 Computer	**$269.00**
C1571 (Disk Drive for C128)	**$NEW**
C1902 (RGB 13" Monitor for C128)	**$NEW**
C1670 (Modem for C128)	**$NEW**
CBM 64	CALL
C1541 Disk Drive	$199.00
C1530 Datasette	$39.99
M-801 Dot Matrix Printer	$169.00
MCS 803 Dot Matrix	$179.00
C1702 Color Monitor	$189.00
C1660 Auto Modem	$59.99
DPS 1101 Daisy Printer	$339.00

Professional Software

Fleet System II w/Spell (128)	$49.99
Trivia Fever	$29.99
Word Pro 4 Plus/5 Plus each	$239.00
Info Pro	$179.00

BRODERBUND

The Print Shop	$29.99
Music Shop	$29.99

pfs:

File (64)	$49.99

BATTERIES INCLUDED

PaperClip w/Spell Pack	$64.99
The Consultant DBMS	$45.99
Bus Card II	$139.00
80 Col Display	$129.00

MODEMS

ANCHOR

Volksmodem	$59.99
Volksmodem 300/1200	$189.99
Signalman Express	$299.00
Lightning 2400 Baud	$399.00

DIGITAL DEVICES

AT300 - 300 Baud (Atari)	$99.99

Hayes

Smartmodem 300	$139.00
Smartmodem 1200	$389.00
Smartmodem 1200B	$359.00
Smartmodem 2400	$599.00
Micromodem IIe	$149.00
Smart Com II	$89.99
Chronograph	$199.00
Transet 1000	$309.00

AST

Reach 1200 Baud Half Card	$399.00

MPP MICROBITS

MPP-1064 AD/AA (C-64)	$69.99

Novation

Smart Cat Plus	$319.00
J-Cat	$99.99
Novation 2400	$499.00
Apple Cat II	$229.00
212 Apple Cat II	$379.00
Apple Cat 212 Upgrade	$229.00
Macmodem	$279.00

QUADRAM

Quadmodem II 300/1200	$339.00
300/1200/2400	$499.00

TELELEARNING

C64 300 Baud (Closeout)	$39.99

EVEREX

1200 Baud Internal (IBM/PC)	$199.00

GRAPHICS

Polaroid

Palette	$1299.00

PORTABLE COMPUTERS

HEWLETT PACKARD

41CV	$189.99
41CX	$249.99
HP 11C	$62.99
HP 12C	$89.99
HP 15C	$89.99
HP 16C	$89.99
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

We stock the full line of HP calculator products

NEC

PC-8401 LS	CALL
PC-8201 Portable Computer	$319.00
PC-8231 Disk Drive	$599.00
PC-8221A Thermal Printers	$149.00
PC-8281A Data Recorder	$99.99
PC-8201-06 8K RAM Chips	$79.99

SHARP

PC-1350	$159.99
PC-1261	$159.99
PC-1260	$99.99
PC-1500A	$169.99
PC-1250A	$89.99
CE-125 Printer/Cassette	$129.99
CE-150 Color Printer Cassette	$159.99
CE-161 16K RAM	$134.99

DISKETTES

maxell

3½" SS/DD (10)	$24.99
3½" DS/DD (10)	$39.99
5¼" MD-1 w/Hardcases (10)	$13.99
5¼" MD-2 w/Hardcases (10)	$19.99
5¼" MD-2-HD for AT (10)	$39.99
3½" 5 pack SS/DD	$15.99

Verbatim.

5¼" SS/DD	$19.99
5¼" DS/DD	$24.99
Disk Analyzer	$24.99

Dennison

Elephant 3½" SS/DD	$29.99
Elephant 5¼" SS/SD	$13.99
Elephant 5¼" SS/DD	$15.99
Elephant 5¼" DS/DD	$16.99
Elephant Premium DS/DD	$22.99

IBM

5¼" DS/DD floppy disks (Box of 10)	$26.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-in-File 10	$2.99
Flip-in-File 50	$17.99
Flip-in-File 50 w/lock	$24.99
Flip-in-File (400/800 ROM)	$11.99
Flip-in-File 100	$24.99

AMARAY

50 Disk Tub 5¼"	$9.99
30 Disk Tub 3½"	$8.99

DRIVES

HARD

I-MEGA

10 meg Bernoulli Box	$1899.00
20 meg Bernoulli Box	$2599.00
5 meg "MacNoulli"	$1499.00

TALLGRASS TECHNOLOGIES COMMITTED TO MEMORY

25, 35, 50, 80 meg (PC) ..... from $1299.00

IRWIN

Tape Backup	CALL

EVEREX

60 Meg Internal Backup System	$829.00

U-SCI

10 meg Internal IBM	$399.00
20 meg Internal IBM	$549.00

FLOPPY

INDUS

Atari GT	$219.00
C-64 GT	$219.00

MICRO SCI

A1.5 Apple	$179.00
A2 Apple	$179.00

1/SD

SD1 C-64 Single	$219.00
SD2 C-64 Dual	$469.00

Tandon

320K 5¼" (PC)	$109.00

TEAC

320K 5¼"	$109.00

CALL TOLL-FREE

MONITORS

AMDEK

Video 300 Green.....$129.00
Video 300A Amber.....$139.00
Video 310A Amber TTL.....$169.00
Color 300 Composite.....$179.00
Color 500 Composite/RGB.....$249.00
Color 600 Hi-Res. RGB.....$399.00
Color 710 Ultra Hi-Res.....$489.00
Color 722 Dual Mode.....$549.00

NEC

JB1270G/1275A.....(ea.) $99.99
JB1280G TTL.....$129.00
JB1285G TTL.....$129.00
JC1460 RGB.....$249.00
JC1225 Composite.....$179.00

PRINCETON

MAX-12E Amber.....$179.00
HX-9 9" RGB.....$469.00
HX-9E Enhanced.....$519.00
HX-12 12" RGB.....$469.00
HX-12E Enhanced.....$559.00
SR-12 Hi-Res.....$599.00

TAXAN

115 12" Green.....$119.00
116 12" Amber.....$129.00
121 TTL Green.....$139.00
122 TTL Amber.....$149.00
610 510x200 RGB.....$NEW
620 640x200 RGB.....$NEW
630 640x200 RGB.....$NEW
640 720x400 RGB.....$NEW

QUADRAM

8400 Quadchrome I.....$499.00
8410 Quadchrome II.....$339.00
8420 Amberchrome.....$179.00
8500 Quad Screen.....$149.00

ZENITH

ZVM 1220/1230.....(ea.) $99.99
ZVM 1240 IBM Amber.....$149.00
ZVM 130 Color.....$269.00
ZVM 131 Color.....$249.00
ZVM 133 RGB.....$429.00
ZVM 135 RGB/Color.....$459.00
ZVM 136 RGB/Color.....$599.00

INTERFACES

AST

Multi I/O (Apple II).....$159.00

PRACTICAL PERIPHERALS

Graphcard.....$79.99
Seriall Card.....$99.99
Microbuffer II+.....$169.00
Microbuffer 32K.....$189.00

QUADRAM

Microfazer.....from $139.00
Efazer (Epson).....from $79.99

Orange Micro

Grappler CD (C64).....$89.99
Grappler + (Apple).....$89.99
Grappler 16K+ (Apple).....$159.00

DIGITAL DEVICES

Ape Face (Atari).....$49.99
U-Print A (Atari).....$54.99
U-A16/Buffer (Atari).....$74.99
U-Call Interface (Atari).....$39.99
U-Print C (C64).....$49.99
P-16 Print Buffer.....$74.99
U-Print 16 apple IIc.....$89.99

PRINTERS

Canon

A40.....CALL
LBP-8A1 Laser.....CALL

CITIZEN

MSP-10 (80 col.).....$279.00
MSP-15 (132 col.).....$389.00
MSP-20 (80 col.).....$349.00
MSP-25 (132 col.).....$509.00

C.ITOH

Prowriter 7500.....$199.00
Prowriter 1550P.....$349.00
Starwriter 10-30.....$399.00

corona

Lazer LP-300.....$2799.00

DIABLO

D25 Daisywheel.....$549.00
635 Daisywheel.....$899.00
D80IF Daisywheel.....CALL

daisywriter

2000.....$749.00

EPSON

Homewriter 10, LX-80, LX-90.....CALL
FX-85, FX-185, RX-100, JX-80.....CALL
DX-10, DX-20, DX-35, LQ-1500....CALL
SQ-2000, Hi-80, HS-80, AP-80.....CALL
LQ-800, LQ-1000.....CALL
Epson/Comrex 220-Atari.....$89.99

JUKI

6000 Letter Quality.....CALL
6100 Letter Quality.....CALL
6200 Letter Quality.....CALL
6300 Letter Quality.....CALL
5510 Dot Matrix.....CALL

LEGEND

808 Dot Matrix 100 cps.....$179.00
1080 Dot Matrix 100 cps.....$259.00
1380 Dot Matrix 130 cps.....$289.00
1385 Dot Matrix 165 cps.....$339.00

NEC

8027 Transportable.....$199.00
3000 Series.....$1099.00
8000 Series.....$1399.00
ELF 360.....$449.00
Pinwriter 560.....$999.00

OKIDATA

182, 183, 192, 193, 2410, 84.....CALL
Okimate 10 (Specify C64/Atari)$189.00
Okimate 20 (IBM).....CALL

Panasonic

KX1080.....$NEW
KX1091.....$259.00
KX1092.....$389.00
KX1093.....$479.00

QUADRAM

Quadjet.....$399.00
Quad Laser.....CALL

SILVER-REED

500 Letter Quality.....$279.00
550 Letter Quality.....$419.00
770 Letter Quality.....$759.00

star

SG-10C (C64 Interface).....CALL
SB/SD/SG/SR Series.....CALL
Powertype Letter Quality.....CALL

Texas Instruments

TI850.....$529.00
TI855.....$639.00
TI865.....$799.00

TOSHIBA

1340 (80 column).....$469.00
P341 (132 column).....$949.00
P351 (132 column).....$1099.00

PC COMPATIBLES

IBM PC SYSTEMS

Configured to your specification. Call for Best Price!

IBM-PC, IBM-PC II, IBM-XT, IBM-AT

KAYPRO

KP-2000 Portable.....CALL

SOFTWARE FOR IBM

ASHTON-TATE

Framework II.....$399.00
dBase III.....$369.00

BORLAND

Turbo Pascal 3.0.....$49.99
Sidekick (unprotected).....$59.99
Reflex.....$59.99

CENTRAL POINT

Copy II PC-Backup.....$29.99

DECISION RESOURCES

Chartmaster.....$229.00
Signmaster.....$169.00
Diagram Master.....$219.00

ENERTRONICS

Energraphics/Plot.....$289.00

FOX & GELLER

Quickcode III.....$169.00

FUNK SOFTWARE

Sideways.....$39.99

HARVARD SOFTWARE

Total Project Manager.....$269.00

INFOCOM

Cornerstone.....$279.00

LIFETREE

Volkswriter Deluxe.....$159.00

LIVING VIDEOTEXT

Think Tank.....$109.00
Ready.....$64.99

LOTUS

Symphony.....CALL
1-2-3.....CALL

MECA SOFTWARE

Managing Your Money 2.0.....$109.00

MICROSTUF SOFTWARE

Crosstalk XVI.....$89.99
Crosstalk Mark IV.....$149.00
Remote.....$89.99

MICRORIM SOFTWARE

R:Base 4000.....$249.00
R:Base 5000.....$389.00
Clout 2.0.....$129.00

MICROPRO

WordStar 2000.....$249.00
WordStar 2000 +.....$299.00
WordStar Professional.....$299.00
Easy.....$99.99

MICROSOFT

Word.....$229.00
Mouse.....$139.00
Flight Simulator.....$39.99
MultiPlan.....$129.00

MULTIMATE

Advantage.....$299.00
Multi Mate Word Proc.....$249.00
On File.....$94.99
Just Write.....$94.99

NOUEMENON

Intuit.....$69.99

NORTON

Norton Utilities 3.1.....$59.99

ONE STEP

Golf's Best.....$39.99

PEACHTREE SOFTWARE

Peachtext 5000.....$179.00

PFS:IBM

First Success.....$209.00
File/Graph.....(ea.) $79.99
Report.....$74.99
Write/Proof Combo.....$79.99

PROFESSIONAL SOFTWARE

Wordplus-PC w/Boss.....$249.00

THE SOFTWARE GROUP

Enable.....$259.00

SATELLITE SYSTEMS

Word Perfect 4.1.....$219.00

SORCIM/IUS

Accounting
AP/AR/GL/INV/OE.....(ea.) $295.00
SuperCalc III.....$195.00
EasyWriter II System.....$195.00
Super Project.....$195.00

SPI SOFTWARE

Open Access.....$379.00

SUBLOGIC

Jet.....$39.99

5th GENERATION

Fast Back.....$119.00

ZENITH

PC-138 Series.....CALL
PC-148 Series.....CALL
PC-158 Series.....CALL
PC-160 Series.....CALL
PC-171 Series.....CALL
AT-200 Series.....CALL

SANYO

MBC 550-2 Single Drive.....$649.00
MBC 555-2 Dual Drive.....$949.00
MBC 675 Portable.....CALL
MBC775.....$1699.00
MBC 880 Desktop.....CALL

AT&T

Safari (7300).....CALL
6300.....CALL

corona

PPC400 Dual Portable.....$1289.00
PPCXT 10 meg Portable.....$1989.00
PC40022 Dual Desktop.....$1389.00
PC400-HD2 10 meg.....$1989.00

ITT

ITT X-TRA
256K, 2 Drive System.....CALL
256K,10 meg Hard Drive System CALL
XP5, 20 meg.....CALL

SPERRY

Sperry-AT.....as low as $1749.00
Sperry-IT.....as low as $2699.00
Call for Specific Configuration!
All Models.....CALL

MULTIFUNCTION CARDS

AST

Six Pack Plus.....$229.00
Mega Plus II.....$269.00
I/O Plus II.....$139.00
Advantage-AT.....$399.00
Graph Pak/64K.....$599.00
MonoGraph Plus.....$399.00
Preview Mono.....$299.00
PC Net Cards.....$379.00
5251/11 On-line.....$669.00
5251/12 Remote.....$579.00

dca

IRMA 3270.....$879.00
IRMA Print.....$999.00
IRMA Smart Alec.....$779.00

EVEREX

Edge Card.....$269.00
Graphics Edge.....$289.00
Magic Card II.....$169.00

HERCULES

Graphics.....$299.00
Color.....$159.00

IDEAssociates

IDEA 5251.....$589.00

MYLEX

The Chairman.....$469.00

PARADISE

Color/Mono Card.....$159.00
Modular Graphics Card.....$279.00
Multi Display Card.....$219.00
Five Pack C, S.....$129.00

PERSYST

Bob Board.....$359.00

TECMAR

Captain - 64.....$199.00
Captain Jr. 128K.....$199.00
Graphics Master.....$469.00

QUADRAM

Quadport-AT.....$119.00
Liberty-AT (128K).....$349.00
The Gold Quadboard.....$449.00
The Silver Quadboard.....$239.00
Expanded Quadboard.....$209.00
Liberty.....$309.00
QuadSprint.....$499.00
QuadLink.....$399.00
QuadColor.....$199.00
QuadJr. Expansion Chassis.....$469.00
Expansion Chassis Memory.....$199.00
Chronagraph.....$79.99
Parallel Interface Board.....$64.99

INTEL

PCNC8087 5MHz.....
PCNC8087-2 8 MHz.....
PCNC80287 6 MHz.....
1010 PC-Above Board.....
1110 PS-Above Board.....
2010 AT-Above Board.....

CALL FOR YOUR PC

COMPUTER MAIL ORDER

1-800-233-8950

www.commodore.ca

The 65816 Chip

New Life For The 6502?

When Apple Computer introduced the Apple IIc in April 1984, the company slogan was "Apple II Forever!". But with the growing popularity of powerful 68000-based 16/32-bit computers like the Atari ST, Amiga, and Apple's own Macintosh, is the Apple II family eventually headed for the eight-bit scrap heap? And what about the future of other eight-bit computers, like the Commodore 64 and 128 and the Atari 400/800/XL/XE computers, all of which are based on the 6502 chip or its offspring?

Veteran chip designer William Mensch believes there's still plenty of potential in the venerable 6502-based machines. How about a 16-bit Apple II, one that's still compatible with the thousands of eight-bit Apples, but which also can address up to 16 megabytes of memory in sections of 64K?

Mensch, founder and director of the seven-year-old Western Design Center in Mesa, Arizona, has developed the 8/16-bit 65816, a hybrid 6502 which is rumored to be Apple Computer's choice for the future of the Apple II. Apple president John Sculley and Apple's executive vice president of product operations, Del Yocum, have already announced that 1986 is the year that the Apple II gets a new CPU.

At this writing (late November), all Mensch can say is that Apple Computer has had samples of his chip since March 1984. Atari bought some samples before Jack Tramiel acquired Atari from Warner Communications, but hasn't announced any plans for upgrading the its eight-bit line with the 65816. Nor has Commodore, so far.

William Mensch is no stranger to new microprocessors. His company created the 65C02 chip that Apple uses in the IIc. And prior to that, Mensch was one of the original designers of the 6502 microprocessor. He also worked at Motorola on the design team that came up with the 6800 chip, the eight-bit predecessor to the 68000.

The 65816 is compatible with the original 6502 (Apple II and II+), the 6502B (Apple IIe), and the 65C02 (Apple IIc). External bus communication is handled eight bits at a time, but the internal data bus handles data in 16-bit chunks. And the clock speed of the 65816 is about 4 megahertz (MHz), compared to 1 or 2 MHz for most 6502 chips. Mensch says his design efforts are pushing toward 6 and even 8 MHz within the next year.

Mensch says a 4 MHz 65816 could run from one to two million instructions per second, about half the speed of a VAX 780 minicomputer, at least for some operations. For word processing and spreadsheet data manipulation, the 65816 could approach the performance of a VAX, he says. Naturally, functions which depend on the eight-bit external bus structure of the 65816 would not be as fast.

"My goal is to elevate the Apple II to the equivalent of a mainframe for the single user," says Mensch.

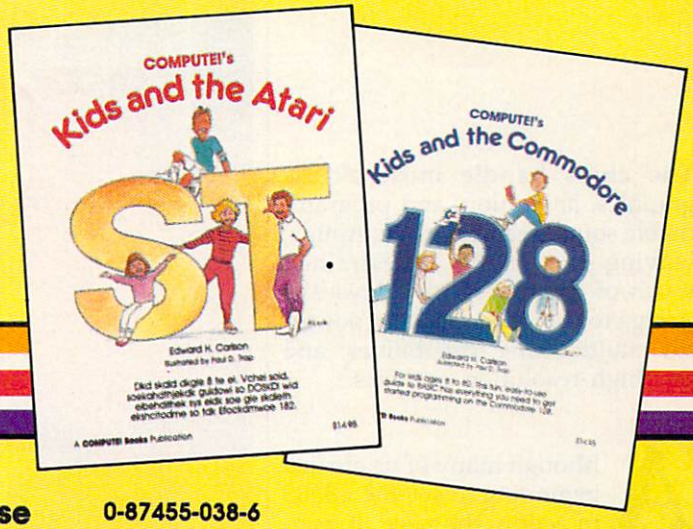
Beyond that, Mensch is shooting for future-generation chips with clock speeds of 100 MHz. That kind of speed would almost certainly require the use of chips based on gallium arsenide crystals rather than silicon, since silicon chips tend to overheat at higher speeds. Gallium arsenide can reportedly handle speeds up to five times faster than today's fastest silicon chips. Mensch's Western Design Center already has two licensees, Northern Telecom and GTE, that are interested in gallium arsenide versions of the 6502. Mensch may begin working toward that goal for the 6502 and 65816 sometime next year.

If a 16-bit Apple II doesn't strike you as wild enough, how about a 32-bit Apple II? Mensch is also developing a 32-bit chip, the 65C832, which is ultimately meant to be a plug-in replacement for the 65816. Mensch says it will have built-in 32-bit floating point math and other operations. If an Apple II already has the 65816, Mensch says it would be a simple matter to pull the 65816 board and plug in the 65C832 board without replacing any other boards or chips. At this point, Mensch expects the 32-bit 65C832 to be available within two or three years.

A decision by Apple Computer on the future of the Apple II line and possible adoption of the 65816 chip may be announced early in 1986, perhaps as soon as the annual Apple stockholders' meeting in January.

NEW

COMPUTE! Books For Kids



Help your children learn the basics of computer programming with these two new entertaining and educational books from COMPUTE!.

0-87455-038-6
$14.95

0-87455-032-7
$14.95

Each book contains easy-to-follow instructions, programming examples, quick reviews, and colorful illustrations. Written in COMPUTE!'s clear, easy-to-understand style, the books offer hours of entertainment while helping kids (and adults) learn to program in BASIC.

If you're acquainted with BASIC, you can easily write your own games and applications on Atari's ST or Commodore's 128 computers. Over 30 sections—all with instructor notes, lessons, assignments, and lively illustrations—entertain and amuse as you learn to use these powerful computers. *COMPUTE!'s Kids and the Atari ST* and *COMPUTE!'s Kids and the Commodore 128*, in the bestselling series from author Edward Carlson, are gentle introductions to programming your new computer. Clear writing and concise examples, both trademarks of this series, make it easy for anyone—child or adult—to learn BASIC painlessly.

Look for these and other books from COMPUTE! at your local book store or computer store. Or order directly from COMPUTE!.

To order, call toll free in the US 1-800-346-6767 (in NY 212-265-8360) or mail the attached coupon with your payment to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please send me the following COMPUTE! books. My payment is enclosed.

_____ ***COMPUTE!'s Kids and the Commodore 128,*** (032-7) $14.95 each _____

_____ ***COMPUTE!'s Kids and the Atari ST,*** (038-6) $14.95 each _____

Subtotal _____

NC residents add 4.5% sales tax _____

Shipping and handling per book (In U.S. and surface mail, $2.00 per book; airmail, $5.00 per book.) _____

Total amount enclosed _____

ALL ORDERS MUST BE PREPAID IN U.S. FUNDS

☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Account No. _____ Exp. Date _____ (Required)

Name _____

Address _____

City _____

State _____ Zip _____

Please allow 4–6 weeks for delivery.

COMPUTE!™ Publications, Inc. abc
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 in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England and in Canada from Holt, Rinehart, & Winston, 55 Horner Avenue, Toronto, ON M8Z 4X6.

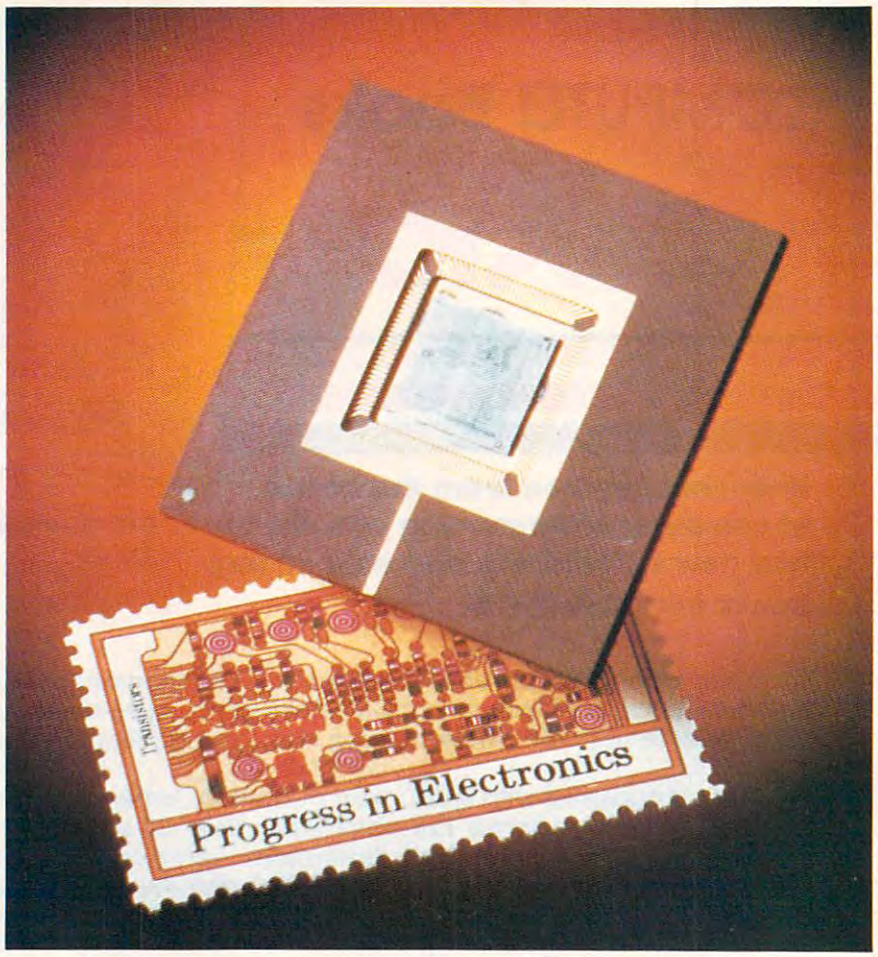
The chips handle most of the graphics, animation, and programmable sound features of the Amiga, leaving the 68000 unrestrained much of the time. This allows the Amiga to provide unusually powerful multitasking capabilities and fast high-resolution graphics.

Although many of us are just beginning to see the 68000 in action, the pace of technological progress in chip design has already leapfrogged to a new generation. In 1984, Motorola introduced the next step: the 68020 microprocessor, a true 32-bit chip that's equivalent to yesterday's powerful mainframe computers. In fact, the 68020 is often called "the mainframe on a chip."

Packed onto the surface of this tiny microprocessor is the equivalent of 200,000 transistors. The chip is capable of executing two to three million computer instructions per second. And the 68020 can address up to a whopping four gigabytes of memory (4,194,304K, or 4,294,967,296 bytes).

Even more importantly, the 68020 is upwardly compatible with its ancestors in the 68000 family. This means that within a year or two, new generations of Macintoshes, Amigas, and STs may be using the 68020 chip—four times as powerful as the 68000—while remaining compatible with earlier software and hardware. In fact, rumors have circulated for more than a year that Apple Computer may base the next generation of its Macintosh on the 68020. And officials at both Atari and Commodore speak of their STs and Amigas in terms of machines that will have future generations.

Although Motorola won't comment on who may be planning to use the 68020, upgrading from the 68000 would not be difficult, says Jeff Nutt, Motorola's technical marketing manager for the 68000 family. "In some cases it can be as simple as pulling out the [68000] processor board and plugging in the 68020 board. And I think you'll see a more rapid visibility with the 68020 than occurred with the 68000 because of the compatibility issue. Look at the normal desktop system; the biggest complaint is having to wait for something [to be processed by the computer]. Not so with the 68020."



Technological advances in silicon-based computer circuitry give today's leading microprocessors the power and speed of mainframe computers for just a few dollars per chip.

Motorola has not been alone in developing such powerful new chips, of course. Intel, National Semiconductor, NCR Corp., and AT&T are all competing in the 32-bit microprocessor arena. Motorola's jump with the 68020 and its compatibility with the 68000, however, have given the company an edge with the latest personal computers.

As a result of this fierce competition, chip prices continue to decline sharply. The 68000, which cost about $450 per unit early in its life, dropped to about $50 in 1984. And, according to one report, in late 1985 Apple Computer was paying as little as $6 per 68000 chip for each Macintosh computer.

Where will it all end? It won't. In fact, Motorola announced in late November that its 68020 has now been successfully tested at the previously unheard-of clock speed of 20 MHz. With such speed and memory potential, personal computers in the very near future will truly be the equivalent of today's mainframe computers.

"It still amazes me," says Motorola's Nutt. "It's incredible when you start thinking about what you can do with something that powerful." ©

Amplify your skills

WITH THESE NEW INTRODUCTORY BOOKS FROM COMPUTE! BOOKS.

These titles will help you unleash the power inside your computer. Whether you're an experienced programmer learning a new language or a beginner just starting out, these books will show you, clearly and quickly, how to get more than you ever thought possible from your computer.

THE AMAZING AMIGA

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

Using AmigaDOS
Arlan R. Levitan and Sheldon Leemon
A comprehensive reference guide and tutorial to the powerful AmigaDOS—the operating system underlying the Workbench and Intuition—this book offers information useful to every Amiga owner. AmigaDOS, the alternative to the icon-based Workbench, lets you control the computer directly. Using AmigaDOS defines and illustrates all DOS commands, and shows you how to create file directories, access peripherals, and run batch file programs. You'll learn why the system prompts you to swap disks, and how to avoid "disk shuffle." The screen- and line-oriented text editors, both overlooked in the user's guide which comes with the Amiga, are explained in detail. Numerous examples and techniques show you how to use AmigaDOS to make operating your computer even more convenient. A full reference section details each DOS command, giving you easy access to the complete AmigaDOS.
ISBN 0-87455-047-5
$14.95

BRING THE ATARI ST ALIVE

Introduction to Sound and Graphics on the Atari ST
Tim Knight
The ST, Atari's powerful new computer, is an extraordinarily impressive sound and graphics machine. Easy to use, the ST can produce color graphics and sound. You'll find thorough descriptions of the computer's abilities, and the information needed to create a complete sound and graphics system. This is the perfect introductory reference to sound and graphics on the Atari ST.
ISBN 0-87455-035-1
$14.95

LEARN C

From BASIC to C
Harley M. Templeton
This introduction to C takes you by the hand and shows how to move from BASIC to this increasingly popular language. BASIC programmers will find this approach designed just for them. Early chapters discuss C language equivalents for common BASIC statements and the similarities and differences between BASIC and C. Later chapters teach everything you need to know to write, debug, and compile programs in C.
ISBN 0-87455-026-2
$16.95

Visit your local bookstore or computer store to purchase any of these new, exciting books from COMPUTE! Publications. Or order directly from COMPUTE!. Call toll free 1-800-346-6767 (in NY 212-265-8360) or mail your check or money order (including $2.00 shipping and handling per book) to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

COMPUTE! Publications, Inc. abc
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 in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England and in Canada from Holt, Rinehart, & Winston, 55 Horner Ave., Toronto, ON M8Z 4X6.

Lyco Computer Marketing & Consultants

ATARI

ATARI

800XL	CALL
130XE (NEW)	CALL
520ST (NEW)	CALL
1050 Drive	165
1010 Recorder	42
1020 Printer	55
1025 Printer	159
1027 Printer	179
850 Interface	109

ATARI

VIP Professional	74.50
ST Companion	15.95

ATARI SOFTWARE (NEW)

Codewriter	35.75
Filewriter	20.75
Reportwriter	20.75
Menuwriter	20.75
Home Integrator	19.75
Small Bus. Inventory	11.75
Salesman Expenses	11.75
Accs Rec/Pay	11.75
Retail Invoice	11.75
Final Legacy	15.75
Adventure Writer	18.75
Star Voyager	11.75

BRODERBUND (Atari)

The Print Shop	28.95
Graphics Library	18.95
Graphics Library II	19.50
Bank St. Writer	42.75
Whistler's Brother	18.95
Spelunker	18.95
Stealth	18.95
Serpent's Star	24.95
Mask of the Sun	24.95

INFOCOM (520ST)

Wishbringer	28.95
Zork I	28.95
Zork II	31.95
Zork III	31.95
Deadline	33.95
Starcross	33.95
Suspect	31.95
Suspended	33.95
Cutthroats	28.95
Sea Stalker	28.95
Infidel	31.95
Sorcerer	31.95
Enchanter	28.95
Witness	28.95
Planetfall	28.95
SpellBreaker	33.95

SPINNAKER

Delta Drawing Room	19.95
Cosmic Life Room	19.95
Up for Grabs Room	19.95

PERSONAL PERIPHERALS

Super Sketch C-64	29.95
Printer Utility C-64	14.99
Super Sketch Atari	25.95
Super Sketch Apple/Ibm	55.00

MICROPROSE (Atari)

Kennedy Approach	21.75
Crusade in Europe	24.75
Decision in Desert	24.75
Solo Flight	20.75
Nato Commander	20.75
Spitfire Ace	18.75
F-15 Strike Eagle	20.75
Hellcat Ace	18.75
Acrojet	21.75
Silent Service	21.75
Gunship	21.75

SSI (Atari)

Colonial Conquest	24.75
Field of Fire	24.75
Kampfgruppe	36.75
Objective: Kursk	24.75
Market Garden	29.75
Six Gun Shootout	24.75
War in Russia	49.75
Computer Baseball	24.75
Computer Quarterback	24.75
Gemstone Warrior	22.75
50 Mission Crush	24.75
Questron	29.75

HABA

10 Megabyte	CALL
Hippo C Compiler	49.95
Check Minder	49.95
Business Letters	36.95
Wills	36.95

SOFTECHNICS

Timelink	75.00
Rhythm	39.95

MIRAGE CONCEPTS

EXPRESS	34.95

SUB LOGIC (Atari)

Flight Simulator II	*32.75*
Night Mission Pinball	20.75

apple

SCARBOROUGH (Apple)

NET WORTH	48.75
Improved Mastertype	22.75
Mastertype Filer	22.75

BRODERBUND (Apple)

The Print Shop	31.50
Graphics Library	18.50
Graphics Library II	18.95
Paper Refill	11.95
Bank St Writer 64K	41.50
Bank St Writer 128K	42.75
Bank St Filer	42.75
Bank St Mailer	42.75
Bank St Speller	42.75
Carmen Sandiego	22.75
Karateka	22.75
Captain Goodnight	22.75
Muppey Cruise	25.75

SUB LOGIC (Apple)

Flight Simulator II	*32.75*
Night Mission Pinball	20.75

MICROPROSE (Apple)

Crusade in Europe	25.75
Decision in Desert	25.75
F-15 Strike Eagle	21.75
Nato Commander	21.75
AcroJet	21.75
Silent Service	21.75
Gunship	21.75

SSI (Apple)

Mech Brigade	36.75
Norway 1985	22.75
Market Garden	29.75
Six Gun Shootout	24.75
50 Mission Crush	24.75
Phantasie	24.75

INNOVATIVE CONCEPTS

Flip-n-File 10	3.50
Flip-n-File 15	8.25

COMMODORE

COMMODORE

C-128 NEW	CALL
1571 Drive	CALL
1572 Drive	CALL
1902 Monitor	CALL
1670 Modem	CALL
C-64 Computer	CALL
1541 Drive	189
MPS801 Printer	LOW
1702 Monitor	199
Simon's Basic	24.75
Assembler 64	34.75
Super Expander	22.75
Logo 64	49.75
Pilot 64	38.75
Easy Cale	34.75
Easy Script	38.75

CARDCO

Digitizer Camera	189.95
32K Printer Buffer	59.95
Numeric Keypad	34.95
CB/5 5-slot Board(64)	54.00
CB/2 2-slot Board(64)	25.00
S'More Basic Rom	49.95
Write Now-64	35.00
Mail Now-64	29.00
Spell Now-64	29.00
File Now-64	29.00
Paint Now-64	29.00
Calc Now-64	29.00
Tax Survival	29.00
Super Printer Utility	27.95
Write Now-Vic-20	29.95

SPINNAKER (C-64ROM)

Cosmic Life ROM	19.75
Jukebox	19.75
Alphabet Zoo	17.75
All in Color Caves	19.75
Up for Grabs	19.75
Delta Drawing	19.75
Kids on Keys	16.75
Kindercomp	14.75
Facemaker	17.75
Fraction Fever	17.75

MICROPROSE (C-64)

Kennedy Approach	21.75
Crusade in Europe	24.75
Decision in Desert	24.75
Solo Flight	20.75
Nato Commander	20.75
Spitfire Ace	18.75
F-15 Strike Eagle	20.75
Hellcat Ace	18.75
Acrojet	21.75
Silent Service	21.75

SSI (C-64)

Colonial Conquest	24.75
Wings of War	24.75
Computer Ambush	34.75
Field of Fire	24.75
Fighter Command	36.75
Kampfgruppe	36.75
Mech Brigade	36.75
Market Garden	29.75
Six Gun Shootout	24.75
Computer Baseball	24.75
Computer Quarterback	24.75
Imperium Galactum	24.75
Phantasie	24.75
Cartels & Cutthroats	24.75
50 Mission Crush	24.75
Questron	24.75

BATTERIES INCLUDED

Paper Clip	59.95
Spell Pak	34.95
Consultant	59.95
Paper Clip w/Spell Pak	75.95
Home Pak	34.95
Bus Card	129.95
80 Column Board	109.95

BRODERBUND

The Print Shop	28.75
Graphics Library	18.75
Graphics Library II	19.75
Graphics Library III	19.95
Karateka	19.75
Castles Dr. Creep	19.75
Bank St. Writer	32.75
Loderunner	20.75
Mask of the Sun	24.75
Spelunker	19.75
Serpent's Star	24.75
Whistler's Brother	18.75
Sid Bungeling Bay	18.75

SUB LOGIC (C-64)

Flight Simulator II	*32.75*
Night Mission Pinball	20.75

COMMODORE

MPS1000 Printer	259
C1350 Mouse	42
C1700 128K RAM	145
C1750 512K RAM	269
JANE	35
Perfect Writer	49
Perfect Calc	49
Perfect Filer	49

MODEMS

US ROBOTICS

Password 1200M	229
Password 1200F	229
Password 300M	139
Password 300F	139
Autodial 212A	359
PCM5	319
PCM64	519
PCM256	769
S-100	379
Courier	469
Microlink	469
Telpac Ms-DOS	79

RACAL-VADIC

2400PC	549
2400PA	799
2400V	559
1200PC	329
300V	205
300PC	199

HAYES

SMARTMODEM 300	133
SMARTMODEM 1200	377
SMARTMODEM 1200B	347
SMARTMODEM 2400	598
MICROMODEM IIE	135

ANCHOR

Volksmodem	55
Volksmodem 12	186
Mark 12	229

TELE LEARNING

Total Telecommunications (C-64)	29.95
AP-250 (300 Baud Apple)	69.95
IB-250 (300 Baud IBM)	69.95

NOVATION

IBM 300/1200 MS-DOS ext	319
IBM 300/1200 CPM-86 ext	319
IBM 300/1200/2400 ext	529
IBM 300/1200/2400 MS-DOS	579
IBM MS-DOS int	325
IBM CPM-86 int	325
Macmodem 300/1200	315

MICROBITS

1064Modem	57

MONITORS

TAXAN

115 12" Greeen Composite	CALL
116 12" Amber Composite	CALL
121 12" Green TTL	135
122 12" Amber TTL	145
220 14" Color Composite	259
410 12" RGB Hi Res IBM	329
420 12" RGB Super Hi IBM	409
440 12" RGB Ultra Hi Res	555
Tilt Stand	35

PRINCETON GRAPHICS

MAX-12 Amber	185
HX-12 RGB	465
SR-12 RGB	595

PANASONIC

DT1300D 13" RGB/Composite	247
DTM140 14" RGB/Composite	329
DTH103 10" RGB Hi Res	395
DTS101 10" Composite	175
DT1000G 10" RGB	166
TX12H3P 12" Color	419
TR120M1PA 12" Green	109
TR120MBPA 12" Amber	109
TR122M9P 12" Green IBM	148
TR122MYP 12" Amber IBM	148

X-TRON

Comcolor I Composite Green	177

SAKATA

SG 1000 12" Green	99
SA 1000 12" Amber	109
SG 1500 12" Green TTL	119
SA 1500 12" Amber TTL	129
SC 100 13" Color Comp	209
SC 200 13" RGB	389
STS1 Tilt Stand	29

ZENITH

ZVM 122A Amber	75
ZVM 123G Green	75
ZVM 124 Amber IBM	129
ZVM 131 Color	275
ZVM 133 RGB	389
ZVM 135 Composite	449
ZVM 136 Hi Res Color	589
ZVM 1220	95
ZVM 1230	95
ZVM 1240	149

TEKNIKA

MJ-10 Composite	179
MJ-22 RGB	255

AMDEK

300 Green	11.
300 Amber	128
310 Amber IBM	155
Color 300 Audio	234
Color 500 Composite	369
Color 600	397
Color 700	495
Color 710	569

NEC

JB-1260 Green	95
JB-1201 Green	135
JC 1215 Color	235
JC 1216 RGB	375
JC 1460 Color	265

LYCO COMPUTER
AMERICA'S MAIL ORDER HEADQUARTERS!

Lyco Computer Marketing & Consultants

SAVE ON THESE IN STOCK PRINTERS

1091.....$233

AXIOM SEIKOSHA
GP250AT (Atari).....222
GP550CD (C-64).....222
GP700AT (Atari).....439
GP700AP (Apple).....439
ELITE5CD (C-64).....229

C. ITOH
ProWriter 8510Sp+.....349
1550Sp+.....489
StarWriter.....769
PrintMaster.....929

TOSHIBA
P1340.....469
P351+.....1149
P341P.....969
P341S.....999
351 Sheet Feeder.....529

CARDCO
32K BUFFER..(C-64).....59

CORONA
LP300 Laser Printer.....2686
200361 Toner Cartridge.....89

EPSON
FX85 (New).....333
LX80.....212
FX185 (New).....464
LX90 (New).....226
SQ2000 (New).....1555
JX80.....467
Homewriter 10.....193
CR-20-Atari.....153
CR-220-C-64.....153
DX-10 (New).....207
DX-20 (New).....297
HS-80 (New).....288
LQ1500P.....975
LQ1500S.....1039
RX-100.....356
FX-100+.....CALL

CITIZEN
MSP-10.....269
MSP-15.....358
MSP-20.....337
MSP-25.....495
Sheetfdr (10/20).....189
Sheetfdr (15/25).....199

OKIDATA
Okimate 10.....179
Okimate 20.....CALL
182.....214
192.....348
193.....563
92P.....349
93P.....565
84P.....645
92 Imagewriter.....349
*IBM versions also

BROTHER
HR-15XL-P.....359
HR-15XL-S.....359
HR-35P.....839
HR-35S.....839
2024L-P.....949
M1009-P.....189

JUKI
Juki 6100.....347
RS232 Serial Board.....55
6100 Tractor.....119
6100 Sheet Feeder.....209
Juki 6300.....757

LEGEND
880.....188
1080.....222
1380.....262
1385.....296
LEGEND 808.....159

DIGITAL DEVICES
16K BUFFER.....75
32K BUFFER.....89
64K BUFFER.....125

DIABLO
D25.....549
630 API.....1599
630 ECS.....1759
D 80 1F.....2395
P 32 CQ1.....699
P 38.....1749
C 150.....999
DX-35 (NEW).....CALL
AP-80.....CALL

SG-10.....$208

PANASONIC
1091.....233
3131 (NEW).....269
1092.....373
1093.....426
3151 Letter.....426
4K Buffer.....65

SILVER REED
EXP400.....249
EXP500.....295
EXP550.....399
EXP770.....749

STAR MICRONICS
SG-10.....208
SG-15.....373
SD-10.....336
SD-15.....442
SR-10.....483
SR-15.....583
SB 10.....595
Power Type.....303
SG-10 C-64 (NEW).....CALL

DUST COVERS

Atari
520ST.....11.95
130XE.....6.99
800XL.....6.99
1050.....6.99
1025.....7.99

Commodore
C128.....7.99
1571/1541.....6.99
1902.....10.95
1702.....8.99
C64/Vic20.....6.99

Panasonic
1090/1091.....8.99
1092.....8.99
1093.....9.99

Star Micronics
SG/SD10.....8.99
SG/SD15.....9.99
SR10.....9.99
SR15.....9.99

Okidata
82/92.....8.99
83/93.....9.99
193.....9.99

DRIVES

INDUS
GT ATARI.....215
GT COMMODORE.....235

COMTEL
Enhancer 2000 (c-64).....179

TANDON
320K ⅘ ¼" Drive.....115

TYMAC
MDD-640 3½" Apple Drive 640K.....289

MSD
SD1 Drive (C-64).....229
SD2 Drive (C-64).....469

PRINTER INTERFACING

AXIOM
AT846 (Atari).....65
Parallax (C-64).....65

MICROBITS
MPP-1150 (Atari).....54
MPP-1150XL (Atari).....59
MicroPrint (Atari).....39
64K RAM Board (600xl).....65

DIGITAL DEVICES
ApeFace XLP (Atari).....49
ApeFace 12XLP (1200xl).....49
U-Print A (Atari).....54
U-Print C (C-64).....49
U-Print A16K Buffer.....74
U-Print A64K Buffer.....99
U-Print AP16K (Apple).....99
U-Print AP64K (Apple).....139
U-CALL RS232 (Atari).....37

MICROTEK
Dumpling GX (Apple).....59
Dumpling 16K (Apple).....89
RV-611C (Apple).....49

TYMAC
Connection (C-64).....55
Tackler (Apple).....49
PPC-100 (Apple).....39

ORANGE MICRO
GRAPPLER+ (Apple).....85
Grappler 16K (Apple).....149
ORANGE (Apple).....59
Grappler CD (C-64).....79

CARDCO
G-Wiz (C-64).....54
C/?+G (C-64).....45
C/?PS (C-64).....49
C/?B (C-64).....39
C/?AT (Atari).....49
C/?AP (Apple).....49

DISKETTES

DENNISON
ELEPHANT 5¼" SSSD...11.99
ELEPHANT 5¼" SSDD...12.99
ELEPHANT 5¼" DSDD...14.99
PREMIUM 5¼" SSDD....13.99
PREMIUM 5¼" DSDD.....15.99

SUNKYONG
SKC 5¼" SSDD.....11.99
SKC 5¼" DSDD.....13.99

MAXELL
5¼" MD1.....13.99
5¼" MD2.....19.99

VERBATIM
5¼" SSDD.....13.99
5¼" DSDD.....19.99

BONUS
5¼" SSDD.....9.99
5¼" DSDD.....12.99

NO LABEL DISKETTES
NL 5¼" SSDD...10.99 (Box 10)
NL 5¼" DSDD...15.99 (Box 10)
*Free Diskette Writer Pen!
*Free Storage Case!

IBM-PC SOFT-WARE

*LOTUS
Lotus 1-2-3.....309.00
Symphony.....439.00

*QUADRAM
Quad Jr. Exp. Chassis....519.00
Quad Jr. Exp. Memory....209.00
Quad Memory Jr.....209.00

ITT XTRA XP Personal Computer!
*256K, 2 Drive System.....CALL
*256K, 10 Meg. Hard System..Call

IBM-PC COMPATABLE

*LEADING EDGE
Nutshell.....69.95
LEWP Basic.....65.00
LEWP Merge Print.....99.00
LE Spell Correction.....169.00

SSI (IBM)
Battle for Normandy.....24.95
Knights of Desert.....24.95
Tigers in Snow.....24.95
Computer Baseball.....24.95
Epidemic.....22.95
Warp Factor.....24.95
Cartels & Cutthroats.....24.95

SYNAPSE (IBM)
Synstock.....64.95
Essex.....28.95
Wizard of Wall St.....28.95
Brimstone.....28.95

SUB LOGIC (IBM)
Jet Simulator.....34.95

BRODERBUND (IBM)
Bank St. Writer.....48.95
The Print Shop.....34.95
Graphics Library I.....22.95
Lode Runner.....22.95
Ancient Art of War.....27.95
Champ Lode Runner.....22.95

MICROPROSE (IBM)
F-15 Strike Eagle.....20.75
Solo Flight.....20.75
Hellcat Ace.....20.75
Spitfire Ace.....18.75

AT&T
Safari 7300.....CALL
6300.....CALL

TOLL FREE 1-800-233-8760

MasterCard **TO ORDER**

CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825

VISA

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA
17740

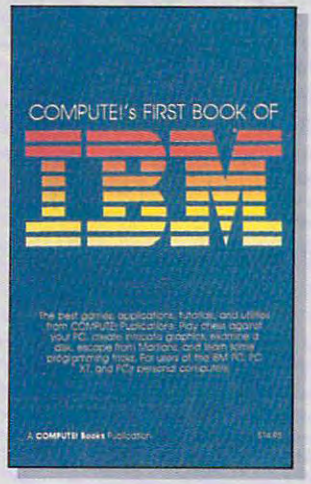
RISK FREE POLICY

In-stock items shipped within 24 hours of order. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. Volume discounts available. PA residents add sales tax. APO, FPO, and international orders add $5.00 plus 3% for priority mail service. Advertised prices show 4% discount for cash, add 4% for MasterCard or Visa. Personal checks require 4 weeks' clearance before shipping. Ask about UPS Blue and Red label shipping. All merchandise carried under manufacturer's warranty. Free catalog with order. All items subject to change without notice.

COMPUTE!'s IBM LIBRARY

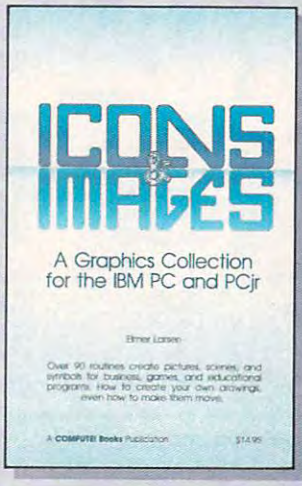
COMPUTE! offers you special savings on this set of top-selling titles for your IBM PC and PCjr.

Each book contains valuable tutorials, programming guides, personal and business applications, and games. Together, the books provide all the up-to-date, ready-to-use information and programs you need to get the most from your IBM personal computer.



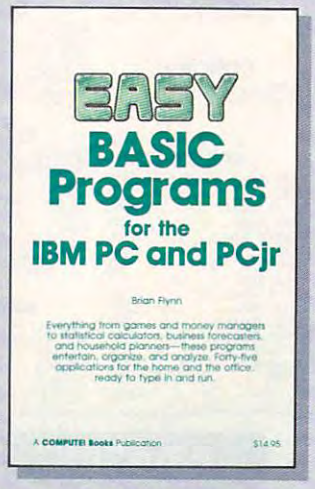
COMPUTE!'s First Book of IBM

Edited, 326 pages
Thirty of the best games, utilities, graphics and sound generators, and applications for the IBM PC and PCjr. A disk is also available which includes programs in the book, $12.95, 0106BDSK.
ISBN 0-87455-010-6
$14.95



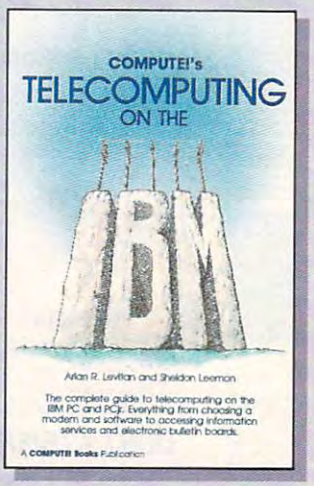
Icons and Images: A Graphics Collection for the IBM PC and PCjr

Elmer Larsen, 227 pages
Ninety-four short routines to instantly enhance business, educational, and entertainment programs on either the IBM PC or PCjr.
ISBN 0-942386-84-1
$14.95



Easy BASIC Programs for the IBM PC and PCjr

Brian Flynn, 359 pages
Everything from games to home and office applications programs is included for the IBM PC and PCjr.
ISBN 0-942386-58-2
$14.95



COMPUTE!'s Telecomputing on the IBM

Arlan R. Levitan and Sheldon Leemon, 274 pages
The ins and outs of telecomputing on the IBM PC or PCjr, from selecting a modem and evaluating terminal software to getting online with the major information services.
ISBN 0-942386-96-5
$14.95

ORDER ALL FOUR BOOKS FOR $49.95 AND SAVE *OVER 15%* OFF THE RETAIL PRICE!

Take advantage of the great price savings and exceptional value of these bestselling books from COMPUTE! and order the four-book set today.

To order, call our toll free customer service number, 1-800-346-6767 (in NY 212-265-8360) and ask for COMPUTE!'s IBM Library. Or, mail the attached coupon with your payment to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

You can also order individual copies of any of the above books at the stated retail price. Sorry, no substitutions. LIMITED TIME OFFER! You must order before March 15 to receive your 15-percent discount.

Please send me the books I have selected. My payment is enclosed.

_____ ***Sets of COMPUTE!'s IBM Library (four books per set) $49.95*** _____
_____ ***COMPUTE!'s First Book of IBM (010-6) $14.95*** _____
_____ ***COMPUTE!'s Telecomputing on the IBM (96-5) $14.95*** _____
_____ ***Easy BASIC Programs for the IBM PC and PCjr (58-2) $14.95*** _____
_____ ***Icons and Images: A Graphics Collection for the IBM PC and PCjr (84-1) $14.95*** _____

All orders must be prepaid.
☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Subtotal _____
NC residents add 4.5% sales tax _____
Shipping and handling ($2.00 per book; $5.00 per book airmail) _____
Total amount enclosed _____

Account No. _____ Exp. Date _____ (required)
Name _____
Address _____
City _____ State _____ Zip _____
Please allow 4–6 weeks for delivery. 36211911

COMPUTE! Publications, Inc. abc
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 in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England and in Canada from Holt, Rinehart, & Winston, 55 Horner Avenue, Toronto, ON M8Z 4X6.

Train for the Fastest Growing Job Skill in America

Only NRI teaches you to service and repair all computers as you build your own 16-bit IBM-compatible micro

As computers move into offices and homes by the millions, the demand for trained computer service technicians surges forward. The Department of Labor estimates that computer service jobs will actually *double* in the next ten years—a faster growth than any other occupation.

Total System Training

As an NRI student, you'll get total hands-on training as you actually build your own Sanyo MBC-550 series computer from the keyboard up. Only a person who knows *all* the underlying fundamentals can cope with *all* the significant brands of computers. And as an NRI graduate, you'll possess the up-to-the-minute combination of theory and practical experience that will lead you to success on the job.

You learn at your own convenience, in your own home, at your own comfortable pace. Without classroom pressures, without rigid night-school schedules, without wasted time. Your own personal NRI instructor and NRI's complete technical staff will answer your questions, give you guidance and special help whenever you may need it.

NRI is the only home study school that trains you as you assemble a top-brand micro-computer. After building your own logic probe, you'll assemble the "intelligent" keyboard...

The Exciting Sanyo 16-bit IBM-compatible Computer—Yours To Keep

Critics hail the new Sanyo as the "most intriguing" of all the IBM-PC compatible computers. It uses the same 8088 microprocessor as the IBM-PC and the MS/DOS operating system. So, you'll be able to choose thousands of off-the-shelf software programs to run on your completed Sanyo.

As you build the Sanyo from the keyboard up, you'll perform demonstrations and experiments that will give you a total mastery of computer operations and servicing techniques. You'll do programming in BASIC language. You'll prepare interfaces for peripherals such as printers and joysticks. Using utility programs, you'll check out 8088 functioning. NRI's easy step-by-step directions will guide you all the way right into one of today's fastest growing fields as a computer service technician. And the entire system, including all the bundled software and extensive data manuals, is yours to keep as part of your training.

Your NRI course includes a Sanyo 16-bit microcomputer with 128K RAM, monitor, double-density/ double-sided disk drive, and "Intelligent" Keyboard; The NRI Discovery Lab®, Teaching Circuit Design and Operations; a Digital Multimeter; Bundled Spread Sheet and Word Processing Software Worth over $1000 at Retail—and More.

...then install the computer power supply, checking all the circuits and connections with NRI's Digital Multimeter. From there you'll move on to install the disk drive and monitor.

100-Page Free Catalog Tells More

Send the postage-paid reply card today for NRI's big 100-page color catalog, which gives you all the facts about NRI training in Microcomputers, Robotics, Data Communications, TV/Video/Audio Servicing, and other growing high-tech career fields. If the card is missing write to NRI at the address below.

NRI SCHOOLS

McGraw-Hill Continuing Education Center

3939 Wisconsin Avenue, NW
Washington, DC 20016

We'll Give You Tomorrow.

IBM is a Registered Trademark of International Business Machine Corporation

A Quantum Leap From 6502 To 68000

Richard Mansfield, Senior Editor

A new era is dawning for machine language programmers on personal computers. Thanks to the extra power of the 68000 microprocessor, the latest-generation computers can offer such advanced features as super high-resolution graphics, multitasking, megabytes of main memory, and processing speed comparable to the mainframe computers of just a few years ago. Here's an introduction to this fascinating chip.

The venerable 6502 microprocessor chip, which has been the brain of the majority of personal computers for a decade, is in the twilight of its life. The new generation of machines—Commodore's Amiga, Atari's ST, and Apple's Macintosh—is built around the 68000 chip. Compared to the 6502 and its relatives, the 68000 is significantly more powerful in the two ways that count: memory and speed.

At their most elementary level, computers spend most of their time getting, sending, and manipulating numbers. Even characters of the alphabet are coded in the computer as numbers. To display a message, for example, the computer fetches a number from memory and sends it to the screen, then fetches the second number and sends it, repeating this get-and-store process until it has sent the entire message. Clearly, the more memory you can gulp at a time, the faster you can manipulate numbers and, by extension, the better you can compute.

Capable of directly addressing 16 megabytes (16,384K or 16,777,216 bytes), the Motorola 68000 greatly exceeds the addressing power of the MOS Technology 6502, which can only address 64K (65,536 bytes). Some computers with 6502 or 6502-compatible chips—such as the Commodore 128, Apple IIc, and Atari 130XE—get around this limitation by switching back and forth between *banks* of memory, but at a cost in speed and programming flexibility.

The 68000 can also be driven at clock speeds of 8 megahertz (MHz) and higher, while most 6502 machines run at 1 MHz. (Again, the Commodore 128 and Atari computers are exceptions; the 128 can be switched to 2 MHz if no peripherals are being accessed, and Atari machines normally run at nearly 2 MHz.) Both the larger addressing and faster speed capabilities of the 68000 contribute to a significant gain in overall computing power. You can hold more data in a 68000 machine, and you can process it faster.

One of the first things anyone wants to know about a new computer language is what commands or instructions are available. The 68000 offers programmers plenty of power. If you're coming to this chip, as most of us are, from 6502 computers—Apples, Commodores, Ataris, and others—it's quite a liberating experience. The 68000 has roughly the

same number of addressing modes as the 6502, but that's where the similarity ends.

For example, many of the 6502 instructions are contained in the single, multipurpose 68000 command MOVE. The 6502's LDA, LDX, LDY, STA, STX, STY, TAX, TAY, TXA, and TYA instructions are all subsumed into MOVE. What makes this work is that the addressing modes for many 68000 instructions are dual-purpose: They specify both the source and destination of a transfer. Since most computer activity involves moving values around and manipulating them, the efficiency of MOVE is most desirable.

Here's how it works: If you want to transfer the number held in address 8000 to address 9000, the single instruction MOVE.B 8000,9000 fetches the value and stores it in the new location. There's no intermediate step as there would be when moving a byte with the 6502—LDA 8000:STA 9000.

Because the 6502 is an eight-bit chip—it can handle only eight bits of information at a time—machine language on the Commodore 64 and other 6502 computers often requires the programmer to fabricate special subroutines to increment, decrement, compare, or perform math on double-byte (16-bit) numbers. While there are only a few such routines necessary and they can be plugged into a program relatively easily, it is still desirable to have the 68000's single-instruction command over multibyte manipulations.

What's more, if you want a loop to access a whole range of memory, there are addressing modes which automatically increment and decrement in one-, two-, or four-byte steps. Specialized moving is also provided for with such instructions as EXG, which exchanges registers so you don't have to move A to C, B to A, and C to B just to exchange A with B. The SWAP instruction swaps the low and high words within a 32-byte data register. MOVEM moves the values in a cluster of registers to or from memory which, among other things, allows you to save and restore all the data and address registers and flags with this single instruction.

More than a dozen of the 68000's instructions are the same as the 6502's: JSR, RTS, NOP, JMP, CMP, etc. On the 68000, they just work with larger numbers when necessary. But there are other instructions that, while named differently, accomplish tasks with which all programmers are familiar. ADD, SUB, MULS, and DIVS perform arithmetic. There are 14 branching instructions, ranging from old friends like BCC and BEQ to new ones like BLT (less than), BLE (less than or equal), and BGE (greater or equal). However, you can branch from −32766 to +32769 bytes rather than the −127 to +128 range of the 6502.

Where the 6502 has only three eight-bit registers, A, Y, and X, the 68000 features eight data registers and seven address registers, each 32 bits large. What's more, these registers can be used in a variety of ways for a variety of purposes. What's possible with data register D0 is possible with any of the other seven data registers. That flexibility is not the case with A, Y, and X on the 6502.

The 68000's data registers can work with bytes, words (two bytes ganged together), or long words (four bytes). The address registers work only with words and long words. The various addressing modes, in conjunction with the multiplicity of registers, allow for considerable speed and many modes of transport between registers or memory. In addition, such things as multiplication and division are built into the chip itself and do not have to be constructed as routines or macros as they do when working with the 6502.

What do you need to get started with 68000 programming? If you're thinking of making the crossover from 6502 to 68000, you'll find the instruction set and addressing modes described in detail in several books currently available. You'll also need an assembler. At this writing (late fall), assemblers are available only from the computer manufacturers, usually as part of professional software development packages. But by early 1986, alternative assemblers from independent companies should be available.

Because of such features as multiple screen windows, multitasking operating systems, and other aspects of these new machines, memory allocation is not static as on earlier computers. The familiar technique of calling operating system hooks, like the Kernal on previous Commodore computers, does not work quite the same way on the ST, Amiga, and Macintosh. For example, on any Commodore, from the earliest PETs to the most recent Commodore 128, you could always JSR $FFD2 to print whatever was in the accumulator. On the new computers, however, your program needs to go through the operating system to make itself known to the screen.

For instance, if two Amiga windows are concurrently running two programs and you want to put something on the screen, you need to follow the rules of Intuition, the Amiga's operating system, to send your message. In this way, machine language begins to resemble aspects of C or other higher-level languages. You need to involve libraries and lists of equates to communicate with your computer, particularly when input/output is involved.

Another consequence of the dynamic memory allocation in these new computers is that you must write your machine language programs to be completely *relocatable*—capable of floating about anywhere in memory, without being dependent on fixed memory addresses. Fortunately, the 68000 includes a powerful set of relocatable branching instructions, such as BSR (Branch to SubRoutine). Some assemblers can even change your address-specific source code into Program Counter-relative, and thus relocatable, object code. And since the computer's operating system determines where your program will reside, there is less worry about memory conflicts.

As our computers grow increasingly complex, there are some additional techniques to master in machine language programming. But the power—and, in a strange way, the simplicity—of the 68000 chip more than makes up for any temporary inconveniences. A new, larger world is opening up for the machine language programmer who wants to accept the challenge. ©

Reach For The Stars For Commodore And Apple

James V. Trunzo

Requirements: Commodore 64 or 128 with a disk drive; or an Apple II-series computer with at least 64K RAM and a disk drive.

Galactic conquest is the theme of many a computer game, and quite a few of the more recent attempts have been solid efforts. A new title, *Reach for the Stars*, is a particularly fine simulation of galactic exploration, combat, and conquest.

Reach for the Stars can be played by up to four players in any combination of computer or human opponents. Each player must explore new star systems, colonize any planets that seem promising, allocate resources, and establish policies and strategies that take into consideration such diverse factors as environment, civil harmony, defense against inevitable alien attacks, and industrial expansion.

This game is special because players must maintain delicate balances to win. You can't build a huge armada at the expense of social programs, or else your colonies will suffer riots, sabotage, disease, and a lower birth rate. Likewise, to be overly concerned with strengthening existing planets while ignoring exploration and colonization allows other players to establish strong bases near your home planet, bases which will eventually build warships and attack your home colony. Strategy is quite important in *Reach for the Stars*.

Beware The Plague

Each turn of the game involves a number of phases: production, movement, combat, planetary conquest, etc. Each phase is handled with full-screen displays and keyboard controls which are efficient and easy to use. To make the game easier to learn, there's a complete tutorial game as well as an excellent rule book.

Reach for the Stars is impressively realistic thanks in part to the great number of interacting options and factors. There are such events as the threat of a star going nova, obliterating everything in the star system; a sudden influx of solar debris, hampering your well-laid movement plans; or plague and famine, weakening your key colony.

When played against the computer, *Reach for the Stars* demands that you remain constantly aware of all aspects of the game, allocating Production Points wisely in an effort to increase your technological level, to produce the best warships, and so on. When played against human opponents, the game makes the same demands, but brings out an additional element: diplomacy, and a nasty companion, treachery. Players may make agreements with each other, granting safe passage through their star systems; or they can gang together and declare war on an opponent who appears to be growing too powerful too quickly.

Reach for the Stars combines an extremely playable, efficient game structure with a sophisticated simulation. It's one of the better games on the market this year.

Reach for the Stars
Strategic Studies Group
Distributed by Electronic Arts
2755 Campus Drive
San Mateo, CA 94403
$45

PC/InterComm For Atari 520ST

George Miller
Assistant Technical Editor

Requirements: Atari 520ST computer and a compatible modem.

PC/InterComm is more than the first commercial terminal program marketed for the Atari 520ST. It's also one of the most versatile and easiest terminal programs we've ever used. All types of communications are a snap, and the looseleaf manual is written in a very clear and concise manner.

With its wide range of features, *PC/InterComm* won't be quickly outdated. As well as easy communications with commercial information services, remote databases, electronic bulletin boards, and other personal computers, its terminal emulation mode lets the 520ST emulate the popular DEC VT102 and VT100 terminals for linkups to DEC mainframes and any of the hundreds of machines running the Unix operating system with 3270 protocol converters.

PC/InterComm allows you to select baud rates from 50 to 19,200 bits per second (bps). Of course, the higher rates are beyond the capabilities of today's personal computer modems, but they do allow high-speed computer-to-computer transfers of data via null modem cables. We did most of our testing on the CompuServe Information Service with a Hayes Smartmodem 1200.

If you have any telecomputing experience at all, you'll probably find yourself online and communicating within minutes of running *PC/InterComm*. It's not strictly necessary to thoroughly read the manual before getting started; help menus and on-screen instructions are available for every function in the program.

Automatic Telecomputing

Customizing *PC/InterComm* is easy, too. Just follow the instructions from the manual or the help menus to select baud rates, stop bits, parity, and other necessary settings. You can even customize your copy of *PC/InterComm* to automatically dial your favorite bulletin board or service as soon as the program runs.

Once you've set up the parameters for communicating with a particular system, you can save the information in a special file on disk. In the future, you won't have to remember these settings or refer to the instructions each time.

For file transfers (uploading and

240K Apple® Compatible Computer System

NEW

SALE

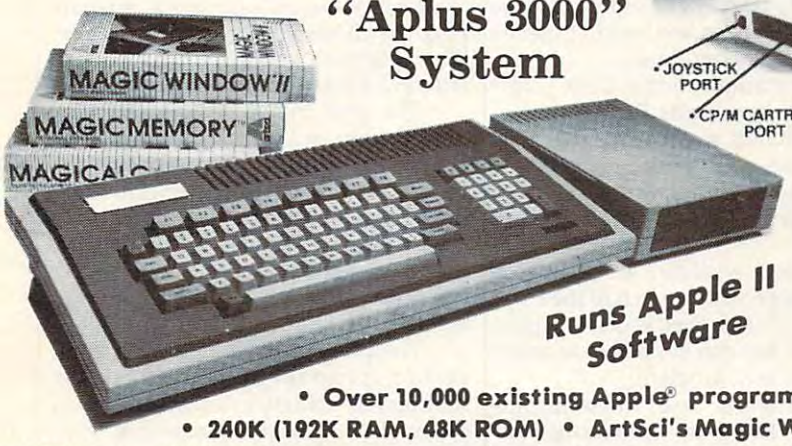
APlus 3000 computer system includes 192K RAM, 48K ROM (32K Microsoft Basic plus 16K ROM Emulator), 160K Laser 5¼" Disk Drive (Runs Apple II Software), Magic Window Wordprocessor, MagiCalc spreadsheet, Magic Memory Database. All for only $399.00

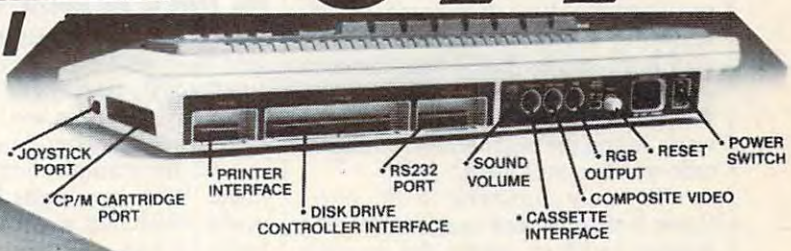
Complete System

$399.00

• 15 Day Free Trial

"Aplus 3000" System





Double Immediate Replacement Warranty

If any of the Aplus 3000 computer system equipment fails due to faulty workmanship or material within 180 days of purchase we will REPLACE it immediately with no service charge!!

- **Over 10,000 existing Apple® programs**
- **Centronics printer interface included**
- **240K (192K RAM, 48K ROM)**
- **ArtSci's Magic Window II, Magic Memory, and MagiCalc included**
- **160K Laser 5¼" Disk Drive (Runs Apple II software)**
- **RGB (80 columns in color) and composite included**

SPECIFICATIONS

A plus 3000 is a complete, self-contained computer based on the popular 6502A microprocessor and can tap into the tremendous software library of Apple II. Features include 192K Bytes RAM, 32KB Enhanced Microsoft BASIC, 80 column text, 560H X 192V color graphic display, 81 key sculptured keyboard and high efficiency switching power supply. Also included as standard are Centronics bus printer interface, Cassette interface, 4 channel sound generator, and 5¼" Apple Compatible Disk Drive.

- **TEXT**
  - — 40 columns X 24 rows or 80 columns X 24 rows software selectable.
  - — 5 X 7 characters in 7 X 8 matrix.
  - — Upper and lower case characters.
  - — One of Eight colors for characters/graphics and background, Red, Green, Blue, Cyan, Magenta, Yellow, Black and White.
  - — Character set with normal, inverse and flashing capabilities.
- **GRAPHICS**
  - — 280H X 192V 6 colors — Black, White, Violet, Green, Blue, Orange.
  - — 280H X 192V 8 colors bit image — Black, White, Red, Green, Blue, Cyan, Magenta, Yellow.
  - — 560H X 192V 6 colors — Black, White, Violet, Green, Blue, Orange. (High resolution color monitor required)

More Features than Apple® for less than Commodore®

Features	Aplus 3000	Apple IIe	Commodore C-128
RAM	192K	64K	128K
Runs Apple II Software	Yes	Yes	No
Function Keys	24	None	16
4 Voice, 6 Octave Sound	Yes	No	Yes
Composite Video	Yes	Yes	Yes
Disk Drive	included	Extra Cost	Extra Cost
Numeric Keypad	included	Extra Cost	Included
Video Cable	included	Extra Cost	Extra Cost
RGB Color Card	included	Extra Cost	Included
80 Column Card	included	Extra Cost	Included
Centronics Printer Interface	included	Extra Cost	Extra Cost
Drive Controller	included	Extra Cost	Included
$150 Wordprocessor (Magic Window)	included	Extra Cost	Extra Cost
$150 Spreadsheet (MagiCalc)	included	Extra Cost	Extra Cost
$60 Database prg. (Magic Memory)	included	Extra Cost	Extra Cost
Your Cost	**$399.00**	$1745.00	$1117.90

Super Apple Compatible Disk Drive Sale $149.95.
Quieter, Cooler, Better Disk Drives for your Apple II plus, IIe, IIc (specify when ordering). List $299.95. **Sale $149.95.**

15 Day Free Trial — If it doesn't meet your expectations within 15 days of receipt, just send it back to us UPS prepaid and we will refund your purchase price!!

Add $25.00 for shipping and handling!!

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add $10 more if C.O.D.

ACCESSORIES

	LIST	SALE
2nd Disk Drive	$299.95	$149.95
2 professional analog joysticks	$ 39.95	$ 24.95
Z-80 cart. allows CP/M use	$ 99.95	$ 59.95
RS232 adapter	$ 99.95	$ 59.95
R/F Modulator (TV hookup)	$ 29.95	$ 19.95
RGB cable (RGB Monitor hookup)	$ 24.95	$ 19.95
Centronics cable (for Centronics printer)	$ 34.95	$ 24.95
Technical reference manual	$ 29.95	$ 19.95
Comstar 10x 120-140 CPS dot matrix printer	$399.00	$179.00
80 columns Hi-Res Amber Monitor	$199.00	$ 89.95
80 column Hi-Res RGB Monitor	$399.00	$259.00

COMPUTER DIRECT

We Love Our Customers

22292 N. Pepper Rd., Barrington, Ill. 60010

312/382-5050 to order

downloading), several different error-checking protocols are available. Modem7 or XMODEM is probably the most useful, since it has become practically a standard, but you can also select from Kermit, Kermit Image, ASCII, Raw, and a proprietary protocol called Inter-PC/InterComm for exchanging files with another computer running *PC/InterComm*. All of these protocols are explained in the manual. Screen messages keep you informed of what's happening during the file transfer.

PC/InterComm also lets the computer dial a database automatically at a predetermined time, then automatically upload or download files without human intervention. Just follow the easy instructions in the manual, then go to sleep if you like and let your ST do the work during the night when communications rates are lowest.

The only drawback to *PC/InterComm* is that you can't exit to the GEM desktop and then reenter the program without rebooting it. For example, it would be nice, before downloading, to view a disk directory from the desktop to make sure there's enough room on the disk. Since the program won't let you move back and forth from the desktop, and since it lacks a directory command of its own, you can't easily obtain this information.

In all other respects, however, *PC/InterComm* is a valuable program that's worth taking a look at.

PC/InterComm
Mark of the Unicorn
222 Third Street
Cambridge, MA 02142
$124

Write 'n Spell

Tony Roberts, Production Director

Requirements: IBM PC, PCjr, or compatible with at least 256K memory and one disk drive.

Finding a full-featured word processor for an IBM PC or PCjr that doesn't cost a fortune has been a difficult task. Home users of the IBM line have often been forced to pay business prices for good word processing software. Now, however, there's *Write 'n Spell* from Professional Software. This program provides considerable power at a moderate cost.

In addition to offering nearly every word processing feature imaginable, *Write 'n Spell* also includes a 90,000-word interactive dictionary, context-sensitive help screens, mail merge capability, and a preview function which allows you to see how your text looks before you print it out.

Write 'n Spell is easy to use—so easy that with the help of only the brief instructions provided in the Quick Start folder, you'll be using the program effectively within minutes. The program disk contains several example files that lead you painlessly through many of *Write 'n Spell*'s important functions. And the well-organized manual includes complete tutorial and reference sections so you can find pertinent information quickly.

No Need To Remember

Write 'n Spell makes extensive use of the IBM function keys for commands and continuously displays a help line at the top of the screen, so you needn't memorize what each key does.

F2, for example, is the DISK command. When you press F2, a window opens which lists your current disk options. To select an option, either indicate your choice with the cursor keys or simply type the first letter of the command.

Command windows are removed from the screen with a touch of the ESC key. In fact, *Write 'n Spell* is quite forgiving; the ESC key can be used to recover from almost any problem.

The program uses meaningful mnemonics for print formatting—*lm* for left margin, *bm* for bottom margin, etc.—and checks the formatting commands for syntax errors. When it encounters a formatting error, the program prints the problem line in the message window, helping you isolate and correct the problem quickly.

Misspellings Begone

The Spelling Checker is one of the most powerful features of *Write 'n Spell*, and it, too, is activated by pressing a function key. It rapidly compares your text with its dictionary and offers four options when it finds a word it doesn't recognize: Ignore, Add, Retype, or Suggest.

If you press I for Ignore, the program skips the word and continues its search. If you press A, the word is added to the supplemental dictionary, which eventually will contain all of the unusual words, names, and numbers you use in your writing. If you press R for Retype, you can correct the misspelled word.

When you press S for Suggest, the program provides a most useful feature for those who find spelling troublesome. It searches through the dictionary to locate up to eight words that it thinks might fit your meaning. It then opens a window displaying those suggestions. If one of them is the word you want, just press its number to replace your misspelled word.

It's amazing how often the program comes up with the correct word, and even more amazing how often the correct word is the first one in its list of possibilities.

It took *Write 'n Spell* less than five minutes to check and correct the text for this article, which contained numerous misspellings (both intentional and unintentional).

Other Features

Write 'n Spell has a wide range of additional features. It can print one document while you work on another, it allows you to link files, and its sophisticated text-manipulation functions include block moves and copies. A setup program allows you to easily configure *Write 'n Spell* to work with more than 50 different kinds of printers.

The mail merge feature lets you insert names and addresses into a stack of form letters, and the program can also accept predefined information from spreadsheets such as *Lotus 1-2-3*. If you have a printer that supports IBM's extended graphics character set, you can print boxed text, complex mathematical formulas, and bar graphs.

The program disk includes a conversion program which helps you transfer files created by other word processors into a format compatible with *Write 'n Spell*. This program is not documented in the manual, but instructions are provided on a loose sheet packed in the box.

Write 'n Spell does not store text files in standard ASCII format, so you must convert them to ASCII if you plan to upload the files via modem. Printing a file to disk converts it to ASCII, but this requires you to use the setup menu to define the disk drive as your printer. Unless *Write 'n Spell* is set up this way, you'll have to save your document, exit the word processor, run the setup program, change the printer configuration, rerun *Write 'n Spell*, and then print the document to disk. The procedure is a little cumbersome, but it gets the job done.

Write 'n Spell does allow normal DOS functions—such as renaming files, erasing files, and copying the current document—without exiting to the system.

Overall, *Write 'n Spell* can handle nearly anything you'd ask of a word processor. Some of its complex operations may be a little cumbersome, but it's a small price to pay for such a powerful program.

Write 'n Spell
Professional Software
51 Fremont Street
Needham, MA 02194
$149.95

152K *Lowest Price In The USA!* 152K

ATARI®

Computer System Sale

• Students • Word Processing • Home • Business

LOOK AT ALL YOU GET FOR ONLY **$399** SYSTEM PRICE

LIMITED QUANTITIES

	LIST PRICE	INDIVIDUAL SALE PRICE
① Atari 130XE 152K Computer	$249.00	$134.95
② Atari 1050 127K Disk Drive	299.00	179.95
③ Atari 1027 Letter Quality 20 CPS Printer	299.00	179.95
Atari Writer Word Processer	59.95	49.95
Atari BASIC Tutorial Manual	16.95	12.95
TOTALS	**$923.90**	**$547.75**

All connecting cables & T.V. interface included.
☆ Monitors sold separetly.

SAVE OVER $100
All 5 ONLY
$399.00
SYSTEM SALE PRICE

CALL FOR 1027 PRINTER REPLACEMENT OPTIONS

Other Accessories	List	Sale	
☆ 12" Hi Resolution Amber Screen Monitor	$199.00	**$59.95**	Add $9.95 for Connection Cables
☆ 13" Hi Resolution Color Monitor	$399.00	**$159.95**	Add $10 for UPS

15 DAY FREE TRIAL. We give you 15 days to try out this ATARI COMPUTER SYSTEM!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!! **90 DAY IMMEDIATE REPLACEMENT WARRANTY**. If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!!

Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service • One Day Express Mail • Programming Knowledge • Technical Support

Add $25.00 for shipping and handling!!
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add $10 more if C.O.D., add $25 if Air Mail.

COMPUTER DIRECT

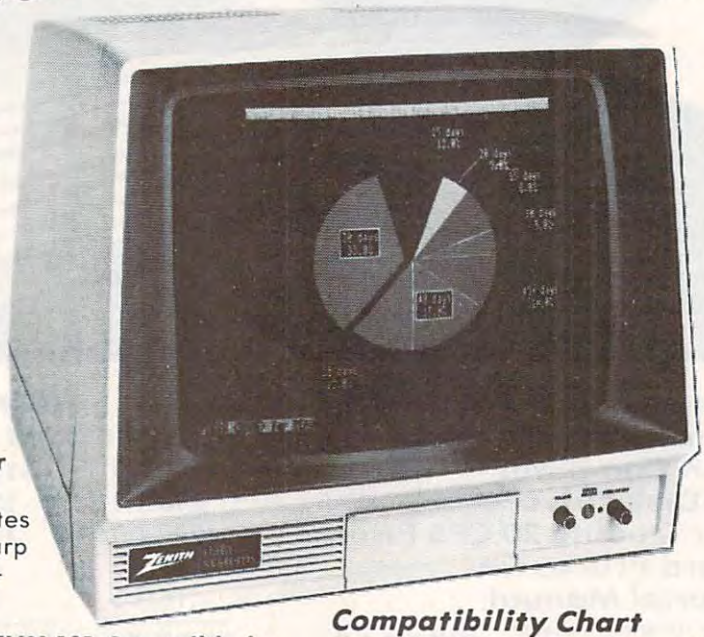
We Love Our Customers
22292 N. Pepper Rd., Barrington, Ill. 60010
312/382-5050 to order

13" Zenith Complete Color Monitor Sale

• Composite • RGB • Sound

Home Computers VCRs Modular TV Tuners Video Games

Sale $139.95 *LIMITED QUANTITIES*



Excellent Color Reproduction & Special "Green Screen Only" Option

True color reproduction is achieved by a Zenith designed state-of-the-art integrated circuit chip that processes the composite video signal. A custom Zenith analog RGB direct drive gain control integrated circuit allows user-preference for the adjustment of picture drive and black level. Zenith's unique "Green Screen Only" feature eliminates all other colors so that monochromatic text material may be easily displayed in green on the black face screen.

Constant Intensity Character Definition Quality

Quality circuitry design generates crisp lines, pure colors, and sharp character definition for easy-to-read displays. DC-coupling permits the video display to retain its color balance from a single dot to a full screen of data. Even when room lighting changes, a "special light sensor" automatically adjusts the display brightness.

List $499.00

Sale $139.95

ZVM 131-Accessible by Many Popular Systems

The ZVM 131 is designed to interface with most personal computers, VCRs, video discs, video games and modular TV tuners that have either composite video or RGB direct drive outputs.

Compatibility Chart

Computer	Interfaces Via
Apple II	Composite
Aplus 3000	RGB
Apple III	RGB
IBM PC	RGB
Commodore 128	RGB/Composite
Commodore 64	Composite
Commodore Vic-20	Composite
TI 99/4	Composite
Atari 800	Composite
Atari 1200	Composite
Atari 1400	Composite

The ZVM 131 Sound Of Quality

The output sound level is externally regulated by a user-adjustment volume control. Use the Zenith quality sound system to monitor the modern audio capabilities of the computer generation.

Easy-To-Reach Front Access Controls

ZVM 131's 13" diagonal display screen can exhibit impressive graphics and intensely clear copy. Easy-to-reach front access user controls (picture, black level, color level, tint, sharpness, audio volume, background noise control) make display adjustment simple and fast. An LED power on indicator notifies the user when the monitor is operable.

Multiple Monitors On A Single Computer

The composite video "loop-thru" feature permits a single composite video source to drive several monitors at the same time. This allows easy display possibilities for multiple viewers in business and educational applications. No more crowding around a single terminal. Everyone enjoys a clear, unobstructed view of important data.

Connection Cables

RGB Cable — $19.95
C128, Aplus 3000 (Specify)

Composite — $9.95
Commodore, Aplus 3000, Atari (Specify)

This Is The Best Value Of The Century

Add $17.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $35.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

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

VISA — MASTER CARD — C.O.D.

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

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

COMMODORE 64 COMPUTER $139.95

COMMODORE 128 COMPUTER $289.00

PLUS FREE $69.95 Word Processor

COMMODORE 64 SYSTEM SALE

SAVE!

LIMITED QUANTITIES

Deal 1
Commodore 64
Com. 1541 Disk Drive
Com. 803 Printer
$407
PLUS FREE $49.95 Oil Barrons Adventure

Deal 2
Commodore 64
1541 Disk Drive
13" Zenith Color Monitor
$457
PLUS FREE $49.95 Oil Barrons Adventure

Commodore C128 Computer $289.00 *

C128 1571 Disk Drive $259.00 *

SUPER OFFER **C128 COMPUTER** *SUPER OFFER*

FREE WORD PROCESSOR COUPON

(Expires 4-1-86) CM46

To Introduce you to the C128 computer we are offering you the finest word processor made, Word Writer II with Spell Checker by Timeworks. When you apply the $69.95 value word processor to your purchase price of the C128 at $289.00 **your net cost is only $219.05.** (1 Coupon per family)

C128 Word Writer with 85,000 word Spell Checker — An 80 column professional word processing system that includes a spelling checker and a built in calculator. Easy to use because of the full screen format, you can view the document on your screen as it will appear when printed. Pull-down menus mean that the user doesn't have to memorize commands. You press a key to activate a Word Writer feature and the program guides you through its proper use with logical and easy-to-follow prompts. The program has been designed to interface with Timework's Data Manager 2, a database program and Swiftcalc, a spreadsheet. Contains all the features you'll need for everyday word processing, plus more sophisticated features such as document chaining, form letter printout, page separations, horizontal and vertical scrolling and much more. (Disk) List $69.95.

* **Commodore C128 Computer $289.00.** This all-new revolutionary 128K computer uses Commodore 64 computer software, CPM Software, plus new advanced C-128 software. You pay only $289 for the C128 computer! **Less** the value of the **Special Software Discount Coupon** (see page 14 of our 64 page catalog) we pack with your computer that allows you to **Save Over $250** off software sale prices!! With only $100 of savings applied your net computer cost is $189.00. ***PLUS FREE $69.95 Word Processor.*** **

* **340K 1571 Commodore Disk Drive $259.00.** Double Sided, Single Disk Drive for C-128 allows you to use C-128 mode plus CPM mode. 17 times faster than the 1541, plus runs Commodore 64 software. You pay only $259.00 for the 340K 1571 Commodore Disk Drive. **Less** the value of the **Special Software Discount Coupon** (see page 14 of our 64 page catalog) we pack with your Disk Drive that allows you to **Save Over $250** off software sale prices! With only $100 of savings applied your net Disk Drive cost is only $159.00.

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

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

VISA — MASTER CARD — C.O.D.

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

PROTECTO

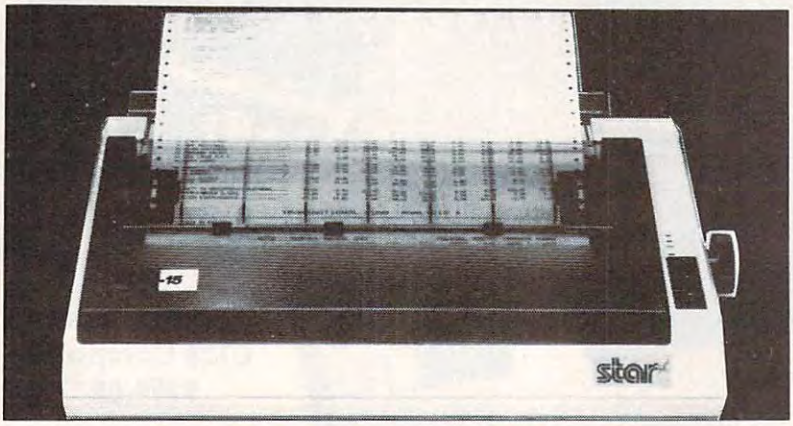
We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

EXTRA WIDE 15" + 120 CPS PRINTER SALE 160 CPS

• One Year Immediate Replacement Warranty **• 15 Day Free Trial**



Comstar 15½X

$229.00

120-140 CPS List $499.00

• Tractor/Friction Printer • Dot Matrix, Impact, Prints Single Sheets or Continuous Feed Paper, 15½" Carriage • Print Buffer • 9 x 9 Dot Matrix, Double Strike • Near Letter Quality, High Resolution Dot Bit Image • Underlining, Left-Right Margin • True Low Descenders, Super and Subscript • Prints Standard, Block Graphics & Italics • Centronics Parallel Interface

COMSTAR 15½X SPECIFICATIONS

(IBM — Commodore) (Apple — Atari — Etc.)

Print Size
10, 12, 17, 5, 6, 8.5 CPI

Number of Columns
136, 164, 232 (68, 82, 116 Double Width)

Character Fonts
Normal (10 CPI); Elite (12 CPI); Condensed (17 CPI); Enlarged (5, 6, 8.5 CPI); Emphasized; Double Strike; Super & Sub Script

Character Sets
96 Standard ASCII, 32 Block Graphic, 96 Italics Characters

Cartridge Ribbon. List $6.95. **Sale $4.95.**

15" Printers use 10" and 15" Paper

CANON 15" Printer

$259.00

List $699.00

160 CPS + Letter Quality Mode

• Programmable Characters
• 2K Buffer • 15 Day Free Trial

CANON SPECIFICATIONS

(IBM — Commodore) (Apple — Atari — Etc.)

Printing Method
Impact dot matrix

Printing Speed
160 CPS at standard character printing
27 CPS at NLQ character printing

Printing Characters
Standard 11 x 9 dot matrix
NLQ 23 x 18 dot matrix
Character set: Full ASCII character set (96), 32 special European characters

Print Buffer
2K-byte utility buffer

Image Printing
Horizontal 120 dots/inch (double density)
Horizontal 240 dots/inch (quadruple density)

Interface
8-bit parallel interface (Centronics type)

Paper
Plain paper, Roll paper, Single sheet, Fanfold, Multipart paper: max. 3 sheets

Ink Ribbon Cartridge — Sale $14.95
Ribbon Life: 3 million characters/cartridge

Maximum Number of Characters

Standard:	10 cpi	80 cpl
Enlarged:	5 cpi	40 cpl
Condensed:	17.1 cpi	136 cpl
Condensed enlarged:	8.5 cpi	68 cpl
Elite:	12 cpi	96 cpl
Elite enlarged:	6 cpi	48 cpl
NLQ pica:	10 cpi	80 cpl
NLQ pica enlarged:	5 cpi	40 cpl

Interfaces

IBM $89.00 **Apple $59.00** **Atari $59.00** **Commodore $39.00**

Add $17.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $35.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

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

VISA — MASTER CARD — C.O.D.

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

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Famous Smith Corona National Brand

10" PRINTER SALE

Below Wholesale Cost Prices!!!

• ONE YEAR IMMEDIATE REPLACEMENT WARRANTY

- **Speed: 120 or 160 characters per second**
- **Friction Feed/Tractor Feed — Standard**
- **80 character print line at 10 CPI**
- **1 Line Buffer, 2K Buffer on 120/160 CPS Plus LQM**
- **Six pitches**
- **Graphics capability**
- **Centronics compatible parallel interface**
- **Features Bidirectional Print, Shortline Seek, Vertical And Horizontal Tabs**

SUPER GRAPHICS

This is a sample of our emphasized near-letter-quality print.

italic print. There is standard data processing quality print

Check these features & prices

120 CPS 10" Printer

List $429.00 **SALE $159**

120 CPS + Letter Quality Mode 10" Printer

List $449.00 **SALE $179**

160 CPS + Letter Quality Mode 10" Printer

List $499.00 **SALE $199**

SPECIFICATIONS

(IBM — Commodore)

Size/Weight
Height 5.04" Width 16.7"
Depth 13.4" Weight 18.7 lbs.
Internal Char. Coding
ASCII Plus ISO
Print Buffer Size
120 CPS: 132 Bytes (1 line)
120/160 CPS Plus LQM: 2K
No. of Char. In Char. Set
96 ASCII Plus International
Graphics Capability
Standard 60, 72, 120 DPI
Horizontal 72 DPI Vertical
Pitch
10, 12, 16.7, 5, 6, 8.3, Proportional Spacing
Printing Method
Impact Dot Matrix

Char. Matrix Size
9H x 9V (Standard) to 10H x 9V
(Emphasized & Elongate)
Printing Features
Bi-directional, Short line seeking, Vertical Tabs, Horizontal Tabs
Forms Type
Fanfold, Cut Sheet, Roll (optional)
Max Paper Width
11"
Feeding Method
Friction Feed Std.; Tractor Feed Std.
Ribbon
Cassette — Fabric inked ribbon
Ribbon Life
4 million characters

(Apple — Atari — Etc.)

Interfaces
Parallel 8 bit Centronics compatible
120/160 CPS Plus NLQ: RS232 Serial inc.
Character Mode
10 x 8 Emphasized; 9 x 8 Standard; 10 x 8 Elongated; 9 x 8 Super/Sub Script (1 pass)
Character Set
96 ASCII
11 x 7 International Char.
Line Spacing
6/8/12/72/144 LPI
Character Spacing
10 cpi normal; 5 cpi elongated normal; 12 cpi compressed; 6 cpi elongated compressed; 16.7 cpi condensed; 8.3 cpi elongated condensed; 5.12.5 cpi elongated proportional

Cartridge Ribbon — List $19.95. **Sale $12.95.**

Interfaces

IBM $89.00 **Apple $59.00** **Atari $59.00** **Commodore $39.00**

Add $14.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $29.00 for CANADA, PUERTO RICO, HAWAII, ALASKA. APO-FPO orders. Canadian orders must be in U.S. dollars.

WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

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

VISA — MASTERCARD — C.O.D. No C.O.D. to Canada or APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Microsoft BASIC 2.1 For Macintosh

Charles Brannon, Program Editor

Requirements: Microsoft BASIC requires 128K memory, but 512K is recommended for best performance. (Amiga watchers take note—Microsoft AmigaBASIC is very similar in features to Macintosh Microsoft BASIC.)

When Apple unveiled its Macintosh in January 1984, it was received enthusiastically. Here was a machine with 128K RAM, a 400K disk drive, and a 16/32-bit 68000 microprocessor running at eight megahertz, one of the fastest, most powerful microprocessors in production. But what really attracted attention was the Macintosh's powerful, yet simple to learn, operating system.

Some critics, though, have questioned the utility of this flashy machine. Sure, it's easy to drag folders around to move or copy files. Certainly it's easier to point to a menu item than to type a cryptic stream of commands on a keyboard. But the glaring lack of a programming language seemed to stratify the market into two classes: users and programmers. Apple described the Macintosh as "the computer for the rest of us." Should that exclude those interested in writing their own programs?

With the first release of Microsoft BASIC, then enhanced BASIC 2.0, and now the even speedier and debugged version 2.1, this last obstacle has fallen. Anyone can now write their own Macintosh applications, whether for fun or profit. And this is no ordinary BASIC: It is one of the most sophisticated and full-featured BASICs available for any personal computer. If you're used to Microsoft BASIC on Apple, Commodore, or IBM machines, you'll instantly see some similarities, but just as quickly notice the differences.

Who Needs Numbers?

The core of Microsoft BASIC for the Macintosh is almost identical to IBM Advanced BASIC, but a major difference is that line numbers are now optional. Line numbers evolved from the simple line-oriented editors used on mainframe computers in the 1960s, and they have survived right into the 1980s. With full-screen editing—including word processor-like up and down scrolling—line numbers become superfluous, except as targets for GOTO and GOSUB statements. Macintosh BASIC even makes line numbers optional in these cases by allowing you to reference GOTOs and GOSUBs with labels, so you can write lines like this:

```
IF Balance<=0 THEN GOTO CheckBounce
...
CheckBounce: PRINT "That'll be a $10 service charge."
```

Since line numbers are optional, Macintosh Microsoft BASIC gives you the freedom to scroll anywhere in a listing and edit any line. The mouse is used to scroll up and down, and to set the insertion point (cursor position). You edit your program in a window called List. You can have two List windows open at once, showing different parts of your program. Familiar Macintosh features such as Cut and Paste are supported.

When you run your program, all input and output takes place in the Output window. A fourth window, called Command, lets you try out direct-statement lines and execute some commands. All these windows can be moved and resized. It's easy to have all four on the screen at once. When you double-click (press the mouse button twice in rapid succession) on a title bar, the corresponding window instantly fills the screen. Another double-click returns the window to its original size.

Microsoft BASIC fits in well with the Macintosh philosophy. It can read and write to the Clipboard, making it easy to transfer data between applications. For example, you can draw a picture with *MacPaint*, then grab and animate the picture in BASIC. Pull-down menus let you save, load, run, stop, and trace programs. The trace feature is especially powerful. While your program is running in the Output window, you can watch the program execute line by line in the List window. In the single-step mode, you can trace the program in one window, watch the output in the Output window, and enter commands in the Command window—all simultaneously.

Add Your Own Commands

Variable names can be up to 40 characters long, and all characters are significant (CHANGE and CHANGENAME would be different variables). Advanced structures include IF–THEN–ELSE and WHILE–WEND. Device-independent input/output lets you use the same I/O commands with all devices (screen, keyboard, printer, clipboard, etc.). Sequential and random-access disk files are supported. Another feature not to be ignored is subprograms. A subprogram is a miniature program with its own independent variables. In effect, you can create your own BASIC commands in BASIC. Subprograms are much more flexible than mere subroutines.

Also, note that there are two versions of BASIC 2.1 included in the package. One does its math in BCD (Binary Coded Decimal), which never makes rounding errors—a vital feature for business programming. The other BASIC uses standard binary floating point, and runs faster.

The Macintosh is known for its superb high-resolution graphics. Microsoft BASIC is no slouch here. It has commands for drawing points, lines, boxes, filled boxes, circles, ovals, and arcs. PUT and GET let you grab and animate rectangular sections of the screen. Pictures can also be stored in strings as a sequence of commands. Just use PICTURE ON, and all graphics calls will be stored until PICTURE OFF is executed. You can then display the picture anywhere on the screen, in any dimension, and enlarge or contract the picture. Microsoft BASIC also lets you use many of the powerful QuickDraw routines in the Macintosh ROM Toolbox.

With Microsoft BASIC, you can write programs that look and act like commercial software, taking advantage of pull-down menus, windows, and dialog boxes. The WINDOW command creates a variety of window styles and shapes. BUTTON creates a square box that is sensitive to mouse selection. MENU and DIALOG let you read the status of menus, windows, and dialog boxes. You can even trap certain events. Your program could be busy drawing a figure, then interrupted when the user selects a menu. This transfers control to your menu subroutine. When the menu action is fulfilled, the program continues drawing the figure.

Speed And Memory

Like many applications, Microsoft BASIC drives the Macintosh to the limits of its hardware. Keeping in mind the great power of this BASIC, there are still some inadequacies. There just isn't enough memory to hold the operating system, your BASIC program, and the Microsoft BASIC interpreter all at once. To get around this, BASIC loads itself in pieces, swapping sections in and out as needed. This can slow you down to a crawl, though, especially when you're switching between windows. It's possible to increase the size of the heap space (where the swapped portions of BASIC reside) at the cost of program space, and this seems to help some.

Still, Microsoft BASIC for the Macintosh runs faster than comparable Microsoft BASICs on other microcomputers. As with many other

Macintosh programs, Microsoft BASIC needs a second disk drive and more memory to live up to its full potential, but works well with a 128K, one-drive system. If you have a Fat Mac (the only type now being manufactured), Microsoft BASIC runs at full speed with no delays and minimal disk access.

Now that a powerful and easy to use BASIC is available, we're beginning to see hundreds of new programs for the Macintosh as eager programmers churn out code for this two-year-old computer. All we need now is an equally powerful BASIC compiler.

Microsoft BASIC 2.1
Microsoft Corporation
10700 Northup Way
Box 97200
Bellevue, WA 98009
$150

Bank Street Mailer Bank Street Filer

James V. Trunzo

Requirements: 128K 80-column version available for Apple IIc or 128K IIe (with extended 80-column card); 64K 40-column versions available for Apple II+ and Commodore 64; one disk drive, printer.

Brøderbund Software continues to enhance its reputation and its home productivity line with the release of *Bank Street Filer* and *Bank Street Mailer*. Both products reflect Brøderbund's effort to provide products with power, flexibility, and ease-of-use. *Filer* and *Mailer* offer a wide variety of options and can be integrated with one another and with *Bank Street Writer*, Brøderbund's word processor.

Both programs are available in 40- or 80-column versions (Apple only); come with simple but thorough tutorial programs; have on-screen menus; let you make backups; offer a dual-drive option; and include a utility program that performs a variety of functions including disk formatting, printer setup, and report-formatting options.

Bank Street Mailer is so versatile that you can not only create mailing lists, but also write and edit a letter without using *Bank Street Writer*. In fact, the letter-writing option contains almost all the features you'd expect to find in a word processor. You can move, copy, center, set page breaks, and use up to 16 printer format commands. In addition, *Mailer* provides a somewhat unique and useful option known as *boilerplating*. You can store commonly used phrases (inside addresses, greetings, etc.) on the *Mailer* disk and insert them in the letters that you compose simply by hitting a key. The only limitation when writing with *Mailer* is a 1,000-word limit per letter.

Of course, the main purpose of *Bank Street Mailer* is to create, edit, and print mailing lists and merge the data into prepared letters. *Mailer* does this admirably. Creating and editing a mailing list is a snap, and all programs of this type are, on the surface, very similar. *Mailer* goes one step beyond by including a powerful data-searching routine which allows you to extract as much or as little information as desired. Furthermore, all data entered into *Mailer* can be sorted alphabetically or numerically.

Merging data with *Mailer* is also easy and can range from simply inserting addresses to customizing form letters by inserting stored messages (reminders, reference material, etc.) selected from disk. If you're only writing a single letter, *Mailer* lets you retrieve

an entire name and address simply by typing a person's last name while composing the letter.

Mailer can print labels and envelopes, too. In fact, this program boasts many additional options which are a delight to use and discover for yourself.

Flexible Report Formatting

Bank Street Filer is no less impressive than its sister programs. It includes all the features common to most database managers, so let's focus on *Filer's* special features.

For one thing, *Bank Street Filer* makes ample use of screen windows, overlaying one block of information atop another. This is especially handy when working with *Filer's* powerful search procedures. *Filer* also provides an on-screen notepad that lets you jot down comments (which can be used later when writing a report) without interrupting your work with the database itself.

Speaking of reports, *Bank Street Filer* can generate a variety of report types: It is already set up to generate four types of "quick" report formats, printing out your data in a selected report style with a few simple keystrokes. However, you can customize either table reports or page reports in almost any way you choose.

Creating forms is also quite flexible: You can position your fields anywhere you like. *Filer* also recognizes special fields like DATE and TIME. You can even call a full-featured calculator to the screen, perform your calculations, and insert them, if you want, into a data field. These are only a few of the many built-in functions that make *Bank Street Filer* an excellent tool and a worthy addition to the Bank Street software series.

Bank Street Mailer
Bank Street Filer
Brøderbund Software
17 Paul Drive
San Rafael, CA 94903-2101
$69.95

Psion Chess For IBM And Macintosh

John Krause, Assistant Technical Editor

Requirements: Apple Macintosh; IBM PC with color/graphics card and at least 128K RAM; or an Enhanced Model IBM PCjr.

So you think computer chess programs are a pushover? There's a new kid on the block. Joint winner of the 1984 World Microcomputer Chess Championship, *Psion Chess* provides a challenge for even the most experienced chess player.

Choose from 14 levels of difficulty, Novice to Infinite. The Novice level responds almost instantly and senses the strength of your play, playing more gently against weak opponents. Still, it's difficult to tame such a powerful beast, and some beginners may be unable to win even on this level. On the Infinite level, the computer keeps thinking until you stop it, at which point it plays the best move it has found. On the Equal level, the computer takes the same amount of time to think as you do. The longer you think, the better the computer plays.

On all levels, after the computer makes a move, it guesses what your next move will be and continues to think while waiting for your move. If it guesses correctly, it responds quickly with its next move and plays better since it can think longer. This feature can be turned off by selecting Handicap, effectively doubling the number of levels.

The program has a thorough knowledge of all the subtleties of chess. Sure, it has the usual library of opening moves, but even if you take it out of its library by making unusual moves, it understands the basic ideas well enough to offer a strong opening without relying on the library. It excels in the middle game, and even in the end game it won't get lost like so many other chess programs.

Psion Chess is quite impressive visually as well. The chess board and pieces can be displayed in either the conventional two-dimensional representation or a spectacular three-dimensional view as if you were seated at a real board. To make a move with the Macintosh version, you use the mouse to pick up the piece and drop it on the destination square. The piece moves smoothly, and in the 3-D view, it realistically passes in front of or behind the other pieces.

The program has almost every feature imaginable. You can take moves back, set up any position, change sides with the computer, replay all the moves of the current game from the beginning, save a game on disk, print out the move list and current position, ask the computer to suggest a move for you, and set up a checkmate problem as complex as mate-in-eight for the computer to solve. Play against the computer, against another player, or watch the computer play itself.

During a game, the computer records a list of the moves that have been made, keeps track of the time spent by each player, displays its analysis of the game indicating which side it thinks is winning, and predicts the next few moves. There's also a selection of 50 classic games drawn from 150 years of international chess which can be replayed move by move.

As if all that weren't enough, the

drop-down menus can be displayed in English, French, German, Italian, Spanish, and Swedish. The 23-page manual contains only three pages of English, but the program has several help screens which explain the features in greater detail. You can use the program quite easily without the manual.

Psion Chess is an impressive programming achievement. It may well be the best chess program ever written for a microcomputer.

Psion Chess
Psion, Inc.
40 Lindeman Drive
Trumbull, CT 06611
$59.95

Quest Of The Space Beagle For Atari

Steve Hudson

Requirements: Atari 400/800, XL, or XE computer with at least 48K RAM and a disk drive.

First came *Jupiter Mission 1999*, a package that combined several interwoven

games under one title. Now there's *Quest of the Space Beagle*. Despite the fact that it's a sequel, it also stands alone—admirably—and if you're an interactive fantasy fan it's sure to become one of your favorites.

In *Jupiter Mission*, you left Earth and became lost in space. Now you're the sole survivor of that mission, and you're trying to get home. As luck would have it, you've been adopted by the Faunians, who are about to be invaded by the barbaric Gentuzians. For some reason, the Faunians have decided that you are the only one in the entire universe who can save them—so it's you and the Faunian fighters against the entire Gentuzian fleet.

You've won the big battle and have been named emperor. But to see if you've got the right stuff, emperor-wise, the Faunians dump you into the smoothly scrolling, multicolored, three-dimensional Labyrinths of Kamerra. Find your way out, they say, and you've not only proven your emperor-hood, but you're also free to head for home. What's another labyrinth to a seasoned adventurer like yourself? Don't worry. It only holds pits and puzzles and Ardillian Whipstingers and Quardish Sycophants.

With practice, those, too, will prove no match for your consummate cosmic skills. You'll be hailed by Faunians far and wide, given a ship and plenty of supplies, and sent on your way. To help you out, you've got star maps—*real* star maps that show an accurate view from any location. You've also got a captured Gentuzian hyperdrive which will get you home fast if you can overcome those pesky "temporal perturbations" that get in the way every time you try to make a hyperspace jump.

Did anyone say this was going to be easy? Maybe not, but it will surely be spectacular. The game's programmers have used some pretty fancy techniques to jazz up an already exciting game. For instance, the graphics display that you see during the space battle is actually *multiple* displays combined into one. The multiple displays alternate 60 times a second, treating you to visuals that would otherwise be impossible to achieve. This is an incredibly realistic display. Don't be surprised if you jump away from the screen every time one of the bad guys comes swooping in.

As noted in the instructions, the price you pay for such graphics excitement is a slight amount of flicker. The flicker is more pronounced on lower-quality monitors or TV sets. However, on each of the monitors tested (including a $59 black-and-white TV) the flicker was all but eliminated by tweaking the color and contrast controls. A little extra effort is required, but it's well worth it.

Quest of the Space Beagle
Avalon Hill Microcomputer Games
4517 Hartford Road
Baltimore, MD 21214
$35

Where In The World Is Carmen Sandiego? For Apple

Karen G. McCullough

Requirements: Apple II-series computer with at least 64K RAM and a disk drive.

FLASH FROM INTERPOL: A national treasure, Aladdin's Lamp, has been stolen from Baghdad. It looks like the work of the Carmen Sandiego gang. Your assignment: track the thief to his/her hideout and recover the treasure. You'll have to work quickly and carefully, though. There's not much time and this gang plays for keeps. If you're the detective you think you are, you should be able to gather clues and decipher them, identify the thief, and track him down. You must've thought you were pretty good, or you wouldn't have signed up with the agency, right? Crack this case and you'll be in line for a promotion.

Where in the World is Carmen Sandiego?, a mystery/adventure game from Brøderbund makes you the detective, chasing an international crook from one exotic city to another, gathering clues to help identify the suspect, and finally cornering him in his hideout. To help you crack the case, you have the services of Interpol's crime computer to identify suspects and the detective's best friend—a copy of the *World Almanac and Book of Facts* (included in the package). Everyone knows that good research skills are as important to a detective as his shoulder holster, and you get plenty of opportunity to put yours to the test. When an informant tells you the suspect converted all her money to yen, can you figure out where she's going? If not, you're in the wrong business.

Starting the game is as easy as booting the disk and entering your name into Interpol's computer. Once the computer has identified you, it gives you the background of the case and whisks you off to the scene of the crime to start your investigation. When you arrive in a city, you have four options: you can see the connections (those are the places the suspect could have gone); depart by plane for one of those destinations; investigate; or visit Interpol to use the crime computer.

You'll want to start by doing some investigation. In each city, there are three places you can go to gather information about the suspect and where he/she was going. Once you've collected some facts about the thief, the crime computer helps you identify the guilty party and issue an arrest warrant if you've gathered sufficient data for a positive identification. You've got to have a warrant or the suspect will slip through your fingers on a legal technicality.

The Carmen Sandiego gang is a wily bunch, and they don't sit still for long. You'll have to track them through a number of cities in all parts of the world, and for that the *Almanac* is essential. The clues can be as subtle as the color of the flag flying on the car in which the suspect is believed to have departed.

Where in the World is Carmen Sandiego? is an entertaining game for anyone from fourth grade up, and even adults will learn something new. The puzzles are different each time you play, and become even tougher as you work your way up through the ranks. There are 10 possible suspects, 30 different cities, and nearly 1,000 clues to provide a variety of challenges. The program also has terrific graphics, clever animation, and some of the best music and sound effects around.

Attention to detail is what has made Brøderbund a leader in the home/entertainment software business; *Carmen Sandiego* reflects that level of care. That it helps teach research skills and fundamentals of geography as well seems almost too good to be true. This is an educational game, but the emphasis is on the game; it's entertaining enough to disguise the fact that you might be learning something while you play.

Where in the World Is Carmen Sandiego?
Brøderbund Software
17 Paul Drive
San Rafael, CA 94903
$39.95

HIGH RISE

Charles McGuyer

You're a construction worker, trapped in a partially completed high-rise building after dark. Can you make it safely to the ground floor without being snared by a giant bird or zapped by the patrol robot? This unique game was originally written in machine language for the Commodore 64. We've added new versions, also written completely in machine language, for Atari and Apple II-series computers. It's one of the best arcade-style games we've ever published, particularly for the Apple. A joystick is required to play the 64 version. The Atari version also requires a joystick and runs on any 400/800, XL, or XE with at least 48K RAM.

The time is the not-too-distant future, and the place is a downtown high-rise building under construction. You're just finishing the day's work when you realize that it's already dark. Everyone else has gone home, leaving you alone in a shadowy, multistory maze of naked girders and bare concrete. A chill creeps down your spine as you think about the recently installed antitheft robot. It patrols the structure from dusk to dawn, automatically disposing of any intruder it might find. Even worse, the half-completed building's shell is infested with nocturnal birds of prey. They're big enough to carry you away, and so hungry that they roam the empty structure all night long, swooping easily from one floor to the next.

Your only hope is to use the temporary elevators. They move randomly during the night hours, going up and down, stopping at some floors, skipping others. With skill and a little luck, you just might evade the dangers around you and make it safely to the ground floor—but it won't be a cakewalk.

Since all three versions of "High Rise" are similar, follow the general game rules under the Commodore 64 instructions. Then refer to the specific section for your computer for additional information and typing instructions.

Commodore 64 Version

Because the 64 version of High Rise is written entirely in machine language, you'll need to type it in with "MLX," the machine language entry program listed elsewhere in this issue. Read the MLX instructions carefully before you type in and save the program. Here's the information you'll need for MLX:

Starting address: C000
Ending address: CE7F

After you save High Rise, turn the computer off, plug a joystick into port 2, then turn it on again. Load the game with LOAD"*filename*",8,1 for disk or LOAD"*filename*",1,1 for tape, and enter SYS 49152 to start it up (substitute your own filename, of course).

The object of High Rise is to make your way to the ground floor via the elevators while evading the birds and patrol robot. When the game begins, you'll see several floors of the building and a number of elevators moving up and down. You can jump on any elevator that comes to your floor (move into the elevator; it picks you up automatically), but there's no way to control its direction or how far it goes. They're just temporary elevators, used to transport materials and workers during daytime hours. The trick is to catch one that's moving in the direction you want, and get off to catch another before it starts moving in an unwanted direction.

When you reach the lowest floor shown, the screen scrolls up one floor, revealing the next lower level. Once you reach ground floor, the player sprints off the screen to safety and you can play another game.

The patrol robot always starts

on an upper floor and moves systematically through the building, traveling up and down through special shafts that are closed to you. Designed to discourage theft and vandalism, its technique is simple and effective: It pushes any intruders (including you) off the building. If it runs into an elevator and detects you inside, it sends a high-voltage charge through the elevator shell until you drop.

Meanwhile, the birds of prey have no trouble moving from one floor to the next, and they'll carry you away whenever they get a chance. Stay as far from an approaching bird as you can, since they can take you even when you're inside an elevator. The birds present another hazard as well. Whenever one of them hits the patrol robot, the hapless fowl is immediately zapped and plummets straight to earth. If you're caught in the path of a falling bird, you'll be knocked down, too.

When the game begins, you have five players. Each time you're zapped or fall from the building, you lose a player; play ends when all five have been lost. When the game starts, you're on the 10th floor of the building. Moving down a level earns you 100 points. If you reach the bottom safely, you'll have another chance to play, beginning at a higher floor. High Rise keeps track of the highest score attained in the current session, as well as your score in the current game.

Atari Version

The Atari version of High Rise must be typed in with "Atari MLX," listed elsewhere in this issue. Be sure to read the MLX instructions carefully before entering and saving the program. When you're ready to save the program, choose the MLX option to make a boot disk or tape. Here are the addresses you'll need for MLX:

Starting address: 12288
Ending address: 14663
Run/init address: 12288

Once you've made a boot disk or tape, follow the instructions in MLX for activating the program. This version of High Rise is quite similar to the Commodore 64 game. However, you begin at the 15th floor of the building rather than the 10th, and the birds aren't zapped when they meet the robot. Also, in this version the robot does not shock the elevator.

Apple Version

High Rise for the Apple II-series computers must be entered with the "Apple MLX" machine language entry program found elsewhere in this issue. Since High Rise loads into the memory area normally used by BASIC programs, you must relocate the start of BASIC memory *before* loading MLX to type High Rise. To do this, enter the following line in direct mode (without a line number) and press RETURN:

POKE 104,28: POKE 7168,0: NEW

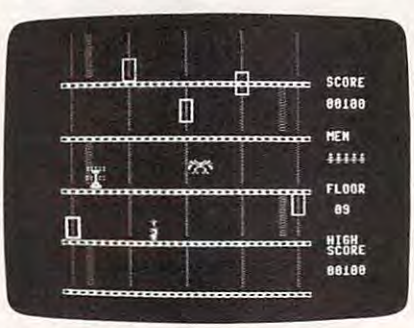
Then load and run MLX. Follow the MLX instructions carefully, using these addresses:

Starting address: 0801
Ending address: 1BD8

After you finish typing High Rise, save at least one copy on disk. Once that's done, you can activate High Rise by typing BRUN *filename* (substituting your own filename, of course).

The Apple version doesn't include birds, so the patrol robot is the only hazard you need to avoid. Move your player with keyboard controls: Press the left-arrow key to move left, right-arrow to move right, and the space bar to stop.

Please refer to the "MLX" articles in this issue before entering the following listings.



"High Rise" for the Commodore 64 features smooth machine language animation and eerie sound effects.

Program 1: Commodore 64 High Rise

```
C000:A9 00 A8 85 FB A9 20 85 10
C008:FC A9 00 85 FD A9 CB 85 7E
C010:FE B1 FD 91 FB C8 D0 F9 F4
C018:E6 FE E6 FC A5 FC C9 24 52
C020:D0 EF A0 18 B9 EE C9 99 52
C028:00 D4 88 10 F7 A0 04 A9 E4
C030:30 99 88 CA 88 10 FA 20 88
C038:E7 C8 20 44 E5 A9 05 8D 95
C040:51 CA A9 31 8D 5F CA A9 8E
C048:30 8D 60 CA A9 00 8D 20 86
C050:D0 8D 21 D0 8D EB C9 8D 0C
C058:88 C9 A0 04 B9 88 CA 99 04
C060:91 07 88 10 F7 20 A7 C7 D5
C068:20 DE C6 AD 97 C9 85 FB 50
C070:20 8B C0 18 AD 97 C9 69 47
C078:07 8D 97 C9 C9 24 F0 03 34
C080:4C 6B C0 A9 01 8D 97 C9 EC
C088:4C AB C0 A9 04 85 FC A2 A0
C090:19 A0 00 A9 1C 91 FB CA 4B
C098:F0 10 18 A5 FB 69 28 85 4F
C0A0:FB A5 FC 69 00 85 FC 4C 1C
C0A8:93 C0 60 A2 00 A9 1D 9D D8
C0B0:A0 04 9D 68 05 9D 30 06 C2
C0B8:9D F8 06 9D C0 07 E8 E0 B6
C0C0:1F D0 EA AC B0 C9 C0 0A 66
C0C8:F0 2D B9 A6 C9 85 FB C8 D4
C0D0:B9 A6 C9 85 FC C8 8C B0 3F
C0D8:C9 A2 04 A0 00 A9 1B 91 E0
C0E0:FB CA F0 10 18 A5 FB 69 EA
C0E8:28 85 FB A5 FC 69 00 85 CC
C0F0:FC 4C DD C0 4C C3 C0 A0 5F
C0F8:00 8C B0 C9 A9 80 A2 00 E4
C100:9D F8 07 E8 E0 05 D0 F8 B5
C108:A9 FF 8D 15 D0 A2 00 AD 22
C110:86 C9 9D 01 D0 E8 E8 18 21
C118:AD 86 C9 69 20 8D 86 C9 F1
C120:E0 0A F0 03 4C 0F C1 A9 B0
C128:32 8D 86 C9 A9 FF 8D 1C 1A
C130:D0 A9 0F 8D 25 D0 A9 02 03
C138:8D 26 D0 A2 00 AD 87 C9 DF
C140:9D 00 D0 E8 E8 18 AD 87 C5
C148:C9 69 38 8D 87 C9 E0 0A 1A
C150:F0 03 4C 3D C1 A9 18 8D DC
C158:87 C9 A0 01 B9 5F CA 99 B0
C160:A2 06 88 10 F7 A9 00 8D BC
C168:EB C9 A9 F0 8D 0A D0 A9 78
C170:B8 8D 0B D0 A9 7C 8D 0C 88
C178:D0 AD 1E D0 A9 68 8D 0D B7
C180:D0 A9 14 8D 0E D0 8D 0F 10
C188:D0 A9 8A 8D FF 07 4C D8 96
C190:C1 8A 48 A2 04 A0 00 88 F5
C198:D0 FD CA D0 FA 68 AA 60 99
C1A0:AC 98 C9 B9 92 C9 DD 01 ED
C1A8:D0 F0 09 C8 C0 05 F0 0D 87
C1B0:8C 98 C9 60 A9 02 9D 88 F8
C1B8:C9 20 61 C2 60 A0 00 8C 93
C1C0:98 C9 60 A5 C5 C9 3F F0 2E
C1C8:05 C9 04 F0 07 60 A9 00 DD
C1D0:8D 18 D4 00 4C B7 C3 60 E4
C1D8:A2 00 20 91 C1 20 C9 C3 B0
C1E0:20 69 C4 20 3B C3 20 A0 33
C1E8:C1 20 6B C2 20 C3 C1 20 A2
C1F0:1B C5 20 B2 C4 20 BD C2 87
C1F8:20 2F C8 BD 88 C9 C9 01 4D
C200:F0 41 90 02 B0 10 FE 01 45
C208:D0 BD 01 D0 C9 E0 F0 25 6B
C210:20 61 C2 4C DA C1 DE C0 77
C218:C9 BD C0 C9 F0 03 4C DA AD
C220:C1 BD 89 C9 9D 88 C9 20 86
C228:61 C2 20 9A C8 29 1F 9D 83
C230:C0 C9 4C DA C1 A9 01 9D 14
C238:88 C9 9D 89 C9 20 61 C2 15
C240:4C DA C1 DE 01 D0 BD 01 90
C248:D0 C9 40 F0 06 20 61 C2 F5
C250:4C DA C1 A9 00 9D 88 C9 D6
C258:9D 89 C9 20 61 C2 4C DA D3
C260:C1 E8 E8 E0 0A F0 01 60 A2
C268:A2 00 60 A0 05 B9 92 C9 53
C270:CD 0B D0 F0 04 88 10 F5 21
C278:60 AD 00 DC 29 0F C9 0B 8B
C280:F0 1D C9 07 F0 0B A9 00 76
C288:8D 0B D4 A9 85 8D FD 07 32
C290:60 AD 0A D0 C9 F5 F0 F8 01
C298:20 F4 C2 20 AD C2 60 AD AC
C2A0:0A D0 C9 17 F0 EA 20 11 8E
C2A8:C3 20 AD C2 60 A9 81 8D 34
C2B0:0B D4 A9 22 8D 08 D4 A9 28
C2B8:80 8D 0B D4 60 CE 54 CA 42
C2C0:AD 54 CA F0 01 60 AC 52 CF
C2C8:CA B9 0B CA 8D 00 D4 B9 FF
C2D0:0C CA 8D 01 D4 A9 11 8D CD
C2D8:04 D4 C8 C8 C0 20 F0 09 AC
C2E0:8C 52 CA A9 23 8D 54 CA F7
```

From the publishers of *COMPUTE!*

February 1986 *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 Apple II, II+, IIe, and IIc computers. The February 1986 *COMPUTE! Disk* contains the entertaining and useful Apple II programs from the December 1985 and January and February 1986 issues of *COMPUTE!*. This easy-to-use disk also features *SpeedCalc*, the spectacular new spreadsheet program written entirely in machine language for the Apple II-series, and the latest version of *SpeedScript*, the bestselling word processing program.

The February 1986 *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 Apple II machine from the previous three issues of *COMPUTE!*.

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

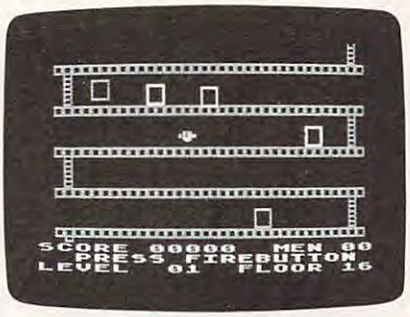
Disks and subscriptions are available for Apple, Atari, Commodore 64 and 128, and IBM personal computers. Call for details.

To subscribe to the *COMPUTE! Disk*, call toll free 1-800-247-5470 or write *COMPUTE! Disk*, P.O. Box 10036, Des Moines, Iowa 50950.

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

```
C2E8:60 A0 00 8C 52 CA A9 23 C3
C2F0:8D 54 CA 60 EE 0A D0 AD A0
C2F8:EA C9 C9 14 F0 05 C9 28 38
C300:F0 06 60 A9 88 4C 0D C3 7B
C308:A9 89 4C 0D C3 8D FD 07 78
C310:60 CE 0A D0 AD EA C9 C9 40
C318:14 F0 05 C9 28 F0 06 60 94
C320:A9 86 4C 0D C3 A9 87 4C 98
C328:0D C3 20 91 C1 CE 65 CA 23
C330:AD 65 CA D0 F5 A9 00 8D 32
C338:65 CA 60 AD 1E D0 8D 1E 79
C340:D0 60 20 91 C1 EE 0B D0 16
C348:20 22 C9 AD 0B D0 C9 F5 A1
C350:D0 F0 A9 00 8D 04 D4 A9 81
C358:B0 8D 6E CA AC 51 CA A9 FF
C360:20 99 D8 05 88 F0 28 8C AE
C368:51 CA 20 2A C3 AD 1E D0 D3
C370:A9 B8 8D 0B D0 A9 40 8D 98
C378:0D D0 8D 0C D0 A9 14 8D 10
C380:0F D0 A9 00 8D EB C9 8D 36
C388:64 CA 68 68 4C D8 C1 18 E9
C390:A2 0B A0 05 20 F0 FF A9 FE
C398:FF 8D 1B D0 A0 00 B9 8D F9
C3A0:CA F0 07 20 D2 FF C8 4C 21
C3A8:9E C3 20 B7 C3 A9 00 8D 42
C3B0:1B D0 68 68 4C 3A C0 A9 04
C3B8:00 8D 18 D4 AD 00 DC 29 44
C3C0:10 D0 F4 A9 7F 8D 18 D4 F4
C3C8:60 AD 1E D0 8D 9E C9 D0 08
C3D0:04 60 4C 66 C4 A0 04 AD C0
C3D8:9E C9 D9 A1 C9 F0 49 C9 E5
C3E0:60 F0 EF 88 10 F1 A0 00 E4
C3E8:AD 0A D0 D9 00 D0 90 25 0B
C3F0:38 F9 00 D0 C9 04 B0 16 F5
C3F8:AD 0B D0 D9 01 D0 90 1F 5D
C400:38 F9 01 D0 C9 0A B0 06 2F
C408:B9 01 D0 8D 0B D0 C8 C8 97
C410:C0 0A D0 D4 60 38 B9 00 3B
C418:D0 ED 0A D0 4C F4 C3 B9 4B
C420:01 D0 ED 0B D0 4C 04 C4 51
C428:8A 48 A2 32 A9 81 99 F8 FF
C430:07 8C EC C9 20 9A C8 8D 25
C438:08 D4 A9 81 8D 0B D4 20 AA
C440:91 C1 AC EC C9 A9 82 99 FA
C448:F8 07 8C EC C9 20 9A C8 3D
C450:8D 08 D4 20 91 C1 AC EC 19
C458:C9 CA 10 D0 A9 80 99 F8 04
C460:07 8D 0B D4 68 AA 4C 42 48
C468:C3 AD ED C9 CE ED C9 F0 4C
C470:01 60 A9 03 8D ED C9 20 CF
C478:D5 C7 20 24 C7 A0 05 B9 A9
C480:92 C9 CD 0D D0 F0 04 88 2B
C488:10 F5 60 AD 9F C9 D0 11 55
C490:AD 0C D0 C9 F8 F0 04 EE 2D
C498:0C D0 60 A9 01 8D 9F C9 4A
C4A0:60 AD 0C D0 C9 18 F0 04 E8
C4A8:CE 0C D0 60 A9 00 8D 9F C4
C4B0:C9 60 CE EA C9 AD EA C9 64
C4B8:F0 05 C9 14 F0 45 60 A9 7D
C4C0:83 8D FE 07 A9 8B 8D FF 56
C4C8:07 A9 29 8D EA C9 AD 0F 27
C4D0:D0 C9 32 90 28 C9 78 B0 8E
C4D8:24 AC 50 CA B9 07 CA 8D 63
C4E0:0E D4 B9 08 CA 8D 0F D4 DD
C4E8:A9 11 8D 12 D4 C8 C8 C0 7A
C4F0:04 F0 04 8C 50 CA 60 A0 11
C4F8:00 8C 50 CA 60 A9 20 8D D3
C500:12 D4 60 A9 84 8D FE 07 CF
C508:A9 8A 8D FF 07 60 A9 40 0A
C510:8D 0D D0 A9 00 8D EB C9 32
C518:4C 50 C5 AD 0B D0 C9 E0 81
C520:F0 01 60 8A 48 20 3A C9 1A
C528:8D 01 D4 20 91 C1 38 AD 09
C530:0B D0 E9 08 8D 0B D0 AD 1B
C538:0F D0 C9 32 90 05 E9 08 50
C540:8D 0F D0 AD 0D D0 C9 48 D2
C548:90 C4 38 E9 08 8D 0D D0 54
C550:20 DE C6 20 11 C7 A0 00 67
C558:A9 C1 85 FB A9 07 85 FC 0B
C560:B9 4B CA 8D 01 D4 A9 11 8D
C568:8D 04 D4 A9 1C 91 FB 18 28
C570:A5 FB 69 07 85 FB C9 E4 FF
C578:F0 03 4C 6B C5 EE 61 CA F4
C580:AD A0 C9 CE A0 C9 10 9D 1B
C588:A9 04 8D A0 C9 AD 1E D0 B7
C590:A2 00 A9 1D 9D C0 07 E8 5B
C598:E0 1F D0 F6 A9 04 8D A0 FE
C5A0:C9 AD 1E D0 A9 00 8D 61 17
C5A8:CA A2 01 DE A2 06 BD A2 9B
C5B0:06 C9 31 F0 52 C9 2F F0 EF
C5B8:3B A2 02 FE 11 05 BD 11 E3
C5C0:05 C9 3A F0 4D A2 00 BD 4A
C5C8:11 05 DD 91 07 F0 18 90 AF
C5D0:0E BD 11 05 9D 91 07 9D 24
C5D8:88 CA E8 E0 05 D0 F2 AD 85
C5E0:53 CA D0 3F 68 AA 60 E8 6E
C5E8:E0 05 D0 DB AD 53 CA D0 1F
C5F0:32 68 AA 60 A9 39 9D A2 1B
C5F8:06 CA BD A2 06 C9 30 F0 C4
C600:B8 DE A2 06 4C B9 C5 CA F5
C608:BD A2 06 C9 30 F0 0C 4C 24
C610:B9 C5 A9 30 9D 11 05 CA 2A
C618:4C BB C5 A9 01 8D 53 CA BD
C620:4C B9 C5 A9 00 8D 53 CA 3D
C628:18 AD 60 CA 69 05 8D 60 C0
C630:CA C9 3A F0 03 4C 40 C6 7C
C638:A9 30 8D 60 CA EE 5F CA F9
C640:18 A0 04 A2 0B 20 F0 FF 67
C648:A0 00 B9 C3 CA F0 07 20 E1
C650:D2 FF C8 4C 4A C6 A9 E0 C6
C658:8D 0B D0 8D 0A D0 A9 86 CF
C660:8D FD 07 CE 0A D0 20 91 67
C668:C1 20 91 C1 A9 87 8D FD B1
C670:07 CE 0A D0 A9 00 8D 04 EF
C678:D4 8D 12 D4 20 AD C2 20 C0
C680:91 C1 20 91 C1 AD 0A D0 0E
C688:C9 02 D0 D2 20 C6 C6 18 84
C690:A0 04 A2 0B 20 F0 FF A0 D9
C698:00 B9 DC CA F0 07 20 D2 93
C6A0:FF C8 4C 99 C6 20 2A C3 52
C6A8:A0 18 A9 20 99 BC 05 88 16
C6B0:10 FA A9 1C 8D C0 05 8D 03
C6B8:C7 05 8D CE 05 20 DE C6 37
C6C0:68 68 68 4C 5A C1 A0 00 89
C6C8:B9 D9 05 C9 20 F0 06 C8 80
C6D0:C0 05 D0 F4 60 A9 1E 99 E8
C6D8:D9 05 EE 51 CA 60 A9 00 B2
C6E0:85 FB A9 D8 85 FC 8A 48 70
C6E8:A2 19 A0 00 A9 02 91 FB 96
C6F0:C8 C0 1F D0 F9 A9 0A 91 20
C6F8:FB C8 C0 27 D0 F9 18 A5 85
C700:FB 69 28 85 FB A5 FC 69 1F
C708:00 85 FC CA D0 DC 68 AA BA
C710:60 AD 13 07 C9 1B F0 06 B0
C718:A9 1B 8D DB 07 60 A9 1B DA
C720:8D C3 07 60 A0 05 B9 B1 8C
C728:C9 CD 0C D0 F0 07 88 10 63
C730:F5 AD 1F D0 60 AD BF C9 1A
C738:D0 38 AD 0D D0 C9 E0 F0 25
C740:52 AD EB C9 D0 14 A0 03 99
C748:AD 1F D0 D9 B7 C9 F0 04 F8
C750:88 10 F8 60 A9 01 8D EB A5
C758:C9 60 EE 0D D0 20 9C C7 9B
C760:60 A9 00 8D EB C9 AD 1F 64
C768:D0 AD 9F C9 49 01 8D 9F 65
C770:C9 60 AD 0D D0 C9 40 F0 A2
C778:1A AD EB C9 D0 0E A0 03 9D
C780:AD 1F D0 D9 BB C9 F0 CC 1A
C788:88 10 F8 60 CE 0D D0 20 F1
C790:9C C7 60 AD BF C9 49 01 FF
C798:8D BF C9 60 EE EB C9 AD 86
C7A0:EB C9 C9 29 F0 BB 60 A0 3C
C7A8:04 B9 6F CA 99 C1 04 B9 D8
C7B0:74 CA 99 89 05 B9 79 CA C5
C7B8:99 51 06 B9 7E CA 99 19 31
C7C0:07 B9 83 CA 99 41 07 A9 E8
C7C8:1E 99 D9 05 A9 30 99 11 AB
C7D0:05 88 10 D5 60 AD 64 CA B1
C7D8:D0 15 AD 0F D0 CD 0B D0 61
C7E0:B0 37 AD 62 CA D0 0E 20 48
C7E8:9A C8 C9 BF F0 07 60 A9 3B
C7F0:8D 8D FF 07 60 A9 01 8D 54
C7F8:62 CA AD 0F D0 CD 0B D0 B7
C800:F0 1B EE 0F D0 AD 0E D0 C9
C808:CD 0A D0 F0 10 B0 06 EE 6A
C810:0E D0 EE 0E D0 CE 0E D0 4A
C818:60 C9 F0 F0 07 CE 0E D0 D9
C820:EE 0F D0 60 A9 00 8D 62 D7
C828:CA A9 20 8D 0F D0 60 AD 90
C830:64 CA D0 35 AD 0F D0 CD 2D
C838:0B D0 B0 1B AD 0A D0 CD 50
C840:0E D0 90 1B 38 ED 0E D0 37
C848:C9 08 B0 0B 38 AD 0B D0 E6
C850:ED 0F D0 C9 07 90 4D AD 16
C858:1E D0 C9 C0 F0 0B 60 38 1F
C860:AD 0E D0 ED 0A D0 4C 48 B9
C868:C8 A9 01 8D 64 CA EE 0F FC
C870:D0 20 9A C8 8D 0F D4 A9 4E
C878:15 8D 12 D4 AD 1E D0 C9 D8
C880:A0 F0 14 AD 0F D0 C9 F5 41
C888:F0 01 60 A9 00 8D 0F D0 9E
C890:8D 64 CA AD 1E D0 60 4C 77
C898:42 C3 8A 48 20 9E E0 68 B7
C8A0:AA A5 62 60 A9 10 8D 04 EF
C8A8:D4 A9 8A 8D FF 07 A9 88 31
C8B0:8D FD 07 20 91 C1 AD 0F 69
C8B8:D0 8D 0B D0 AD 0E D0 8D 59
C8C0:0A D0 CE 0F D0 A9 8B 8D 28
C8C8:FF 07 20 91 C1 20 9A C8 C5
C8D0:8D 0F D4 A9 11 8D 12 D4 D9
C8D8:AD 0F D0 C9 19 D0 C5 A9 FC
C8E0:10 8D 12 D4 4C 5C C3 78 41
C8E8:A9 33 85 01 A9 00 85 FB 31
C8F0:85 FD A9 D0 85 FC A9 30 AA
C8F8:85 FE A2 02 A0 00 B1 FB E5
C900:91 FD 88 D0 F9 E6 FC E6 46
C908:FE CA D0 F0 A9 37 85 01 2D
C910:58 A9 1C 8D 18 D0 A0 1F FA
C918:B9 CA C9 99 D8 30 88 10 B6
C920:F7 60 AC 6E CA 8C 01 D4 A3
C928:A9 11 8D 04 D4 88 8C 6E 17
C930:CA A9 00 8D 0B D4 8D 12 45
C938:D4 60 A9 04 85 FC A9 00 37
C940:85 FB A2 1E 18 A5 FB 69 84
C948:28 85 FD A5 FC 69 00 85 7E
C950:FE A0 00 B1 FD 91 FB E6 BB
C958:FB A5 FB D0 02 E6 FC CA 50
C960:10 E2 18 A5 FB 69 09 85 2F
C968:FB A5 FC 69 00 85 FC A5 4F
C970:FB C9 C0 D0 CD A5 FC C9 62
C978:07 D0 C7 A9 20 A0 1E 99 B0
C980:C0 07 88 10 FA 60 40 18 3A
C988:00 00 00 00 00 00 00 00 1C
C990:00 00 40 68 90 B8 E0 01 DC
C998:00 21 22 24 28 30 00 00 FC
C9A0:04 61 62 64 68 70 1B 04 60
C9A8:CB 04 AB 05 5B 06 3B 07 59
C9B0:00 27 28 29 E7 E8 E9 BF 1C
C9B8:9F 3F 1F FF DF 7F 5F 00 8B
C9C0:04 04 04 04 04 04 04 04 54
C9C8:04 04 82 54 82 54 82 54 B3
C9D0:82 54 20 10 20 10 20 10 51
C9D8:20 10 00 00 00 00 FF C3 44
C9E0:C3 FF 08 1C 08 1C 2A 5D 7B
C9E8:14 14 15 00 00 14 00 00 7E
C9F0:00 00 00 00 45 F0 00 00 00
C9F8:00 00 45 F0 00 00 00 00 44
CA00:11 58 58 00 00 02 7F 1E 64
CA08:19 1F 15 47 05 47 05 ED 46
CA10:05 47 05 47 06 47 05 0C 72
CA18:07 47 06 47 05 47 05 ED 75
CA20:05 47 05 47 06 47 05 0C 82
CA28:07 47 06 1F 15 1F 15 1F 34
CA30:15 1F 15 B5 17 B5 17 B5 89
CA38:17 B5 17 1E 19 1E 19 1E 1D
CA40:19 1E 19 B5 17 B5 17 B5 DB
CA48:17 B5 17 31 DF 87 A2 00 2F
CA50:00 05 00 00 19 00 F7 DC BC
CA58:C0 A4 88 6C 50 34 18 31 04
CA60:30 00 00 00 00 00 0D 07 2F
CA68:06 05 0E 0B 0A 09 A1 13 7F
CA70:03 0F 12 05 0D 05 0E 20 96
CA78:20 06 0C 0F 0F 12 08 09 EB
CA80:07 08 20 13 03 0F 12 05 4E
CA88:30 30 30 30 30 1E 59 4F 47
CA90:55 27 56 45 20 52 55 4E FC
CA98:20 4F 55 54 20 4F 46 20 EC
CAA0:4D 45 4E 0D 0D 1D 1D 1D FC
CAA8:1D 1D 50 52 45 53 53 20 81
CAB0:46 49 52 45 20 54 4F 20 6B
CAB8:50 4C 41 59 20 41 47 41 1D
CAC0:49 4E 00 47 52 4F 55 4E CB
CAC8:44 20 46 4C 4F 4F 52 2C 9E
CAD0:59 4F 55 27 52 45 20 53 3F
CAD8:41 46 45 00 4E 4F 57 20 C7
CAE0:57 45 20 47 4F 20 41 20 89
CAE8:4C 49 54 54 4C 45 20 48 C6
CAF0:49 47 48 45 52 00 FE 00 EA
CAF8:EC 00 EE 00 6E 62 02 62 46
CB00:00 00 00 00 00 00 00 00 97
CB08:00 00 00 00 0A AA 80 08 A3
```

```
CB10:00 80 08 00 80 08 00 80 6D
CB18:08 00 80 08 00 80 08 00 56
CB20:80 08 00 80 08 00 80 08 4B
CB28:00 80 08 00 80 08 00 80 85
CB30:08 00 80 08 00 80 08 00 6E
CB38:80 08 00 80 0A AA 80 83 99
CB40:40 10 00 44 01 01 10 04 70
CB48:04 04 01 04 0A AA 90 08 67
CB50:00 80 48 00 84 18 00 80 16
CB58:48 00 81 18 00 84 48 00 68
CB60:90 48 00 84 08 00 80 08 E3
CB68:00 81 18 00 84 48 00 81 2A
CB70:18 00 84 48 00 90 48 00 FB
CB78:84 08 00 81 0A AA 84 83 F3
CB80:40 00 10 10 44 40 04 10 76
CB88:11 10 00 04 0A AA 90 48 51
CB90:00 84 18 00 81 48 00 84 FD
CB98:18 00 90 08 00 84 08 00 F0
CBA0:80 18 00 80 48 00 90 18 02
CBA8:00 84 08 00 81 18 00 84 53
CBB0:48 00 84 48 00 81 08 00 97
CBB8:84 48 00 90 4A AA 80 83 2F
CBC0:00 00 00 00 00 00 00 00 58
CBC8:00 00 00 00 00 00 00 11 71
CBD0:11 10 00 00 00 11 11 10 6B
CBD8:00 A8 00 00 88 00 00 10 EE
CBE0:00 00 10 00 33 33 30 00 41
CBE8:20 00 33 33 30 00 20 00 EB
CBF0:00 20 00 00 A8 00 00 88 5E
CBF8:00 02 8A 00 02 AA 00 FF 1D
CC00:00 00 00 00 00 00 00 00 99
CC08:00 00 00 00 00 00 00 04 A5
CC10:44 40 10 00 10 04 44 40 37
CC18:00 A8 00 00 88 00 00 10 30
CC20:00 00 10 00 0C CC C0 30 01
CC28:20 30 0C CC C0 00 20 00 72
CC30:00 20 00 00 A8 00 00 88 9F
CC38:00 02 8A 00 02 AA 00 FF 5E
CC40:00 00 00 00 00 00 00 00 D9
CC48:00 00 00 00 00 00 00 00 E1
CC50:0C 00 00 33 00 00 04 00 2B
CC58:00 04 00 00 2A 00 00 A2 E6
CC60:80 00 AA 80 02 22 20 02 72
CC68:2A 20 03 22 30 00 15 00 4D
CC70:00 11 00 00 11 00 00 11 E7
CC78:00 00 11 00 00 FF C0 FF B5
CC80:00 00 00 00 00 00 00 00 1A
CC88:00 00 00 00 00 00 00 00 22
CC90:0C 00 00 33 00 00 04 00 6B
CC98:00 04 00 00 0A 00 03 0A 93
CCA0:80 00 8A 20 00 2B 80 00 7B
CCA8:05 00 00 15 00 00 44 00 9E
CCB0:00 44 00 00 41 00 00 40 A5
CCB8:40 00 C0 30 03 00 C0 00 27
CCC0:00 00 00 00 00 00 00 00 5A
CCC8:00 00 00 00 00 00 00 00 62
CCD0:0C 00 00 33 00 00 04 00 AB
CCD8:00 04 00 00 0A 00 00 0A CD
CCE0:00 00 0A 80 00 3A 80 00 AD
CCE8:05 00 00 31 00 00 01 00 1A
CCF0:00 05 00 00 14 00 00 15 81
CCF8:00 00 04 80 00 37 00 FF F7
CD00:00 00 00 00 00 00 00 00 9B
CD08:00 00 00 00 00 00 00 00 A3
CD10:30 00 00 CC 00 00 10 00 B0
CD18:00 10 00 00 A0 00 02 A0 61
CD20:C0 08 A2 00 02 E8 00 00 26
CD28:50 00 00 54 00 00 11 00 53
CD30:00 11 00 00 41 00 01 01 1D
CD38:00 0C 03 00 03 00 C0 00 D0
CD40:00 00 00 00 00 00 00 00 DB
CD48:00 00 00 00 00 00 00 00 E3
CD50:30 00 00 CC 00 00 10 00 F0
CD58:00 10 00 00 A0 00 00 A0 9D
CD60:00 02 A0 00 02 AC 00 00 53
CD68:50 00 00 4C 00 00 40 00 71
CD70:00 50 00 00 14 00 01 50 13
CD78:00 03 10 00 00 DC 00 FF 4A
CD80:00 00 00 80 00 02 A0 00 6D
CD88:0A C0 00 03 30 00 0C FC 20
CD90:00 3F F0 82 0F 3F 28 FC 05
CD98:0F 14 F0 00 82 00 00 14 07
CDA0:00 00 82 00 02 00 80 02 9F
CDA8:00 80 08 82 20 00 00 00 8E
CDB0:00 00 00 00 00 00 00 00 4C
CDB8:00 00 00 00 00 00 00 00 54
CDC0:00 00 00 00 00 00 00 00 5C
CDC8:00 00 00 00 00 00 00 00 64
CDD0:00 00 32 00 8C 3C 82 3C 49
CDD8:CF 14 F3 30 82 0C F0 14 1D
CDE0:0F 80 82 02 82 00 82 02 AF
CDE8:00 80 02 00 80 08 82 20 2E
CDF0:00 00 00 00 00 00 00 00 8C
CDF8:00 00 00 00 00 00 00 00 94
CE00:0C 00 80 80 00 02 A0 00 05
CE08:02 00 C3 03 00 00 0C 00 67
CE10:00 0F F0 82 0F 0F 28 F0 AD
CE18:C3 14 F0 00 82 03 30 14 4F
CE20:00 00 82 0C C2 00 80 02 E7
CE28:30 80 08 80 20 00 00 08 10
CE30:00 00 00 00 00 00 00 00 CD
CE38:00 00 00 00 00 00 00 00 D5
CE40:00 00 00 00 00 00 00 00 DD
CE48:00 00 00 00 C0 00 C0 00 6D
CE50:30 03 02 00 80 3C 82 30 31
CE58:C0 14 F3 30 02 0C C3 14 B8
CE60:00 80 82 02 8E 00 8E 02 22
CE68:20 00 02 00 80 08 02 23 A1
CE70:00 00 00 00 20 00 00 00 0F
CE78:00 00 00 00 00 00 00 00 16
```



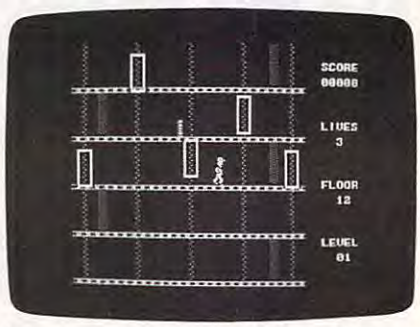
You must avoid hungry birds as well as an automatic patrol robot in the Atari version of "High Rise."

Program 2: Atari High Rise

Version by Kevin Mykytyn, Editorial Programmer

```
12288:032,139,052,032,018,052,069
12294:032,043,054,169,000,141,189
12300:249,055,032,014,051,032,189
12306:199,051,032,124,051,032,251
12312:035,057,206,254,054,208,070
12318:020,173,255,054,141,254,159
12324:054,032,115,050,173,065,013
12330:006,073,001,141,065,006,078
12336:032,252,055,162,000,160,197
12342:002,202,208,253,136,208,039
12348:250,032,172,053,032,212,043
12354:054,173,242,055,208,042,072
12360:160,014,185,214,055,153,085
12366:022,032,136,016,247,173,192
12372:251,055,024,105,005,141,153
12378:251,055,141,244,055,238,050
12384:241,055,032,172,053,169,050
12390:120,141,250,055,173,250,067
12396:055,208,251,076,014,048,248
12402:173,006,055,208,007,173,224
12408:012,208,041,002,208,017,096
12414:206,247,055,208,145,173,136
12420:248,055,141,247,055,032,142
12426:142,056,076,020,048,162,130
12432:006,136,208,253,202,208,133
12438:250,238,032,006,173,032,113
12444:006,141,000,210,208,237,190
12450:206,238,055,208,200,238,027
12456:249,055,032,172,053,160,121
12462:008,185,189,055,153,026,022
12468:032,136,016,247,169,100,112
12474:141,250,055,173,250,055,086
12480:208,251,160,015,185,198,185
12486:055,153,022,032,136,016,100
12492:247,173,132,002,208,251,193
12498:032,000,053,076,006,048,169
12504:216,206,250,055,173,249,085
12510:055,208,110,169,000,133,129
12516:077,165,217,240,003,076,238
12522:073,049,032,085,052,174,187
12528:250,054,189,035,055,240,039
12534:060,016,028,189,080,006,113
12540:056,233,040,157,080,006,056
12546:189,081,006,233,000,157,156
12552:081,006,222,045,055,016,177
12558:036,032,082,049,076,051,084
12564:049,189,080,006,024,105,217
12570:040,157,080,006,189,081,067
12576:006,105,000,157,081,006,131
12582:254,045,055,189,045,055,169
12588:201,070,208,003,032,082,128
12594:049,032,085,052,206,250,212
12600:054,206,250,054,016,008,132
12606:169,008,141,250,054,032,204
12612:047,050,032,181,049,032,203
12618:091,049,032,237,056,076,103
12624:098,228,189,035,055,073,246
12630:128,157,035,055,096,162,207
12636:002,138,024,105,116,133,098
12642:206,169,000,133,205,189,232
12648:048,006,168,169,007,141,131
12654:248,054,169,000,145,205,163
12660:200,206,248,054,016,248,064
12666:189,032,006,201,180,176,138
12672:042,201,030,144,038,168,239
12678:138,141,249,054,189,064,201
12684:006,010,010,010,170,169,003
12690:007,141,248,054,189,069,086
12696:055,145,205,232,200,206,171
12702:248,054,016,244,174,249,119
12708:054,189,016,006,157,000,074
12714:208,189,032,006,157,048,042
12720:006,202,016,169,096,173,070
12726:000,055,208,081,173,006,193
12732:055,208,076,173,120,002,054
12738:074,074,074,176,024,173,021
12744:016,006,201,055,240,042,248
12750:206,016,006,173,064,006,165
12756:201,008,208,032,169,006,068
12762:141,064,006,208,025,074,224
12768:176,071,173,016,006,201,099
12774:190,240,015,238,016,006,167
12780:173,064,006,201,008,208,128
12786:005,169,004,141,064,006,119
12792:206,003,055,208,013,169,134
12798:003,141,003,055,173,064,181
12804:006,073,001,141,064,006,039
12810:096,173,006,055,240,011,079
12816:173,034,006,024,105,001,103
12822:141,032,006,208,014,174,085
12828:001,055,189,045,055,010,127
12834:024,105,039,141,032,006,125
12840:096,169,008,141,064,006,012
12846:096,160,008,185,035,055,073
12852:208,013,173,010,210,201,099
12858:010,176,019,185,000,006,198
12864:153,035,055,185,045,055,080
12870:162,004,221,019,055,240,003
12876:008,202,016,248,136,136,054
12882:016,221,096,173,010,210,040
12888:201,100,176,244,185,035,005
12894:055,153,000,006,169,000,221
12900:153,035,055,204,001,055,091
12906:208,228,169,000,141,000,084
12912:055,240,221,173,253,054,084
12918:208,109,173,017,006,024,143
12924:109,251,054,141,017,006,190
12930:201,190,240,004,201,055,253
12936:208,012,173,251,054,073,139
12942:255,024,105,001,141,251,151
12948:054,096,141,002,055,162,146
12954:255,173,032,006,205,033,090
12960:006,240,065,144,014,162,023
12966:001,169,058,141,214,050,031
12972:169,186,141,220,050,208,122
12978:010,169,058,141,220,050,058
12984:169,186,141,214,050,142,062
12990:252,054,173,002,055,174,132
12996:033,006,236,025,055,240,023
13002:010,236,027,055,240,005,007
13008:236,029,055,208,006,201,175
13014:058,208,011,240,004,201,168
13020:186,208,005,169,001,141,162
13026:253,054,096,173,033,006,073
13032:024,109,252,054,141,033,077
13038:006,162,004,221,025,055,199
13044:240,004,202,016,248,096,026
13050:162,001,173,010,210,016,054
13056:002,162,255,142,251,054,098
13062:169,000,141,253,054,076,187
13068:120,050,160,002,185,059,076
13074:055,153,016,006,185,062,239
13080:055,153,032,006,153,048,215
13086:006,185,065,055,153,064,046
13092:006,136,016,232,169,000,083
13098:141,253,054,141,000,055,174
13104:141,006,055,169,001,141,049
13110:251,054,169,010,141,255,166
13116:054,141,254,054,169,000,220
13122:133,217,169,004,133,215,169
13128:169,010,141,003,055,169,107
13134:005,141,247,055,141,248,147
13140:055,169,000,141,008,210,155
13146:141,005,055,169,003,141,092
13152:015,210,169,163,141,001,027
13158:210,169,010,141,004,055,179
13164:169,000,141,030,208,160,048
13170:019,169,000,153,020,032,251
13176:136,016,250,096,173,000,023
```

```
13182:055,208,025,160,004,173,239
13188:016,006,056,249,030,055,032
13194:201,003,144,013,201,254,186
13200:176,009,136,016,238,169,120
13206:000,141,000,055,096,140,070
13212:247,054,169,004,056,237,155
13218:247,054,010,170,189,045,109
13224:055,010,024,105,038,056,200
13230:237,032,006,201,004,144,030
13236:004,201,255,208,220,189,233
13242:035,055,240,215,169,001,133
13248:141,000,055,142,001,055,074
13254:096,169,062,141,047,002,203
13260:169,001,141,111,002,160,020
13266:003,185,055,055,153,192,085
13272:002,169,000,153,008,208,244
13278:136,016,242,169,112,141,014
13284:007,212,133,206,169,003,190
13290:141,029,208,169,000,133,146
13296:205,162,008,169,000,168,184
13302:145,205,136,208,251,230,141
13308:206,202,016,246,096,160,154
13314:004,177,207,024,105,040,047
13320:145,207,200,177,207,105,025
13326:000,145,207,096,032,199,181
13332:051,160,008,165,088,133,113
13338:205,165,089,133,206,165,221
13344:205,024,105,006,153,080,093
13350:006,133,205,165,206,105,090
13356:000,153,081,006,133,206,111
13362:136,136,016,233,169,008,236
13368:141,250,054,032,085,052,158
13374:206,250,054,206,250,054,058
13380:016,245,169,008,141,250,129
13386:054,160,216,162,048,169,115
13392:007,032,092,228,096,172,195
13398:250,054,185,080,006,133,026
13404:203,185,081,006,133,204,136
13410:185,009,055,010,010,010,121
13416:010,170,169,007,141,245,078
13422:054,160,000,189,112,053,166
13428:081,203,145,203,200,232,156
13434:189,112,053,081,203,145,137
13440:203,032,098,053,232,206,184
13446:245,054,016,229,096,032,038
13452:075,054,032,109,056,032,242
13458:133,056,169,000,141,047,180
13464:002,169,000,141,048,002,002
13470:169,035,141,049,002,169,211
13476:034,141,047,002,169,000,045
13482:133,088,169,064,133,089,078
13488:160,010,169,000,162,002,167
13494:032,027,053,232,224,037,019
13500:208,248,152,024,105,016,173
13506:168,201,122,208,235,162,010
13512:003,160,014,169,002,141,177
13518:246,054,169,002,032,027,224
13524:053,200,200,200,200,206,247
13530:246,054,016,244,152,024,186
13536:105,020,168,201,118,144,212
13542:228,160,002,138,024,105,119
13548:032,170,201,067,240,004,182
13554:169,001,208,215,173,048,032
13560:002,133,207,173,049,002,046
13566:133,208,169,000,168,153,061
13572:000,032,136,208,250,160,022
13578:019,185,149,055,153,000,059
13584:032,185,169,055,153,040,138
13590:032,136,016,241,096,072,103
13596:010,010,141,245,054,138,114
13602:072,152,072,165,088,133,204
13608:203,165,089,133,204,192,002
13614:000,240,006,032,098,053,219
13620:136,208,250,138,024,101,141
13626:203,133,203,165,204,105,047
13632:000,133,204,173,245,054,105
13638:170,169,003,141,245,054,084
13644:160,000,189,160,053,145,015
13650:203,032,098,053,232,206,138
13656:245,054,016,242,104,168,149
13662:104,170,104,096,165,203,168
13668:024,105,040,133,203,165,002
13674:204,105,000,133,204,096,080
13680:170,170,128,002,128,002,200
13686:128,002,128,002,128,002,252
13692:128,002,170,170,255,255,080
13698:192,003,192,003,192,003,203
13704:192,003,192,003,192,003,209
13710:255,255,085,085,064,001,119
13716:064,001,064,001,064,001,087
13722:064,001,064,001,085,085,198
13728:255,195,195,255,012,048,096
13734:012,048,195,255,195,255,102
13740:173,239,055,174,240,055,084
13746:160,007,032,210,053,173,045
13752:238,055,162,000,160,015,046
13758:032,210,053,173,241,055,186
13764:162,000,160,045,032,210,037
13770:053,173,242,055,162,000,119
13776:160,055,141,007,055,142,000
13782:008,055,140,237,055,169,110
13788:000,141,243,055,162,003,056
13794:160,255,200,173,007,055,052
13800:056,253,229,055,141,007,205
13806:055,173,008,055,253,233,247
13812:055,141,008,055,176,234,145
13818:173,007,055,024,125,229,095
13824:055,141,007,055,173,008,183
13830:055,125,233,055,141,008,111
13836:055,152,208,009,224,002,150
13842:144,005,172,243,055,240,109
13848:011,009,016,238,243,055,084
13854:172,237,055,153,000,032,167
13860:238,237,055,202,016,184,200
13866:096,169,000,141,239,055,230
13872:141,240,055,169,001,141,027
13878:241,055,169,010,141,242,144
13884:055,169,005,141,238,055,211
13890:169,014,141,251,055,141,069
13896:244,055,096,032,133,056,176
13902:160,000,169,112,153,000,160
13908:035,200,192,003,208,248,202
13914:169,077,153,000,035,165,177
13920:210,200,153,000,035,200,126
13926:165,211,153,000,035,165,063
13932:210,024,105,040,133,210,062
13938:165,211,105,000,133,211,171
13944:200,192,243,208,221,169,073
13950:070,153,000,035,200,169,241
13956:000,153,000,035,200,169,177
13962:032,153,000,035,200,169,215
13968:006,153,000,035,200,192,218
13974:248,208,248,169,065,153,217
13980:000,035,200,169,000,153,201
13986:000,035,200,169,035,153,242
13992:000,035,096,165,210,133,039
13998:212,165,211,133,213,160,244
14004:004,165,212,153,000,035,237
14010:200,165,213,153,000,035,184
14016:165,212,024,105,040,133,103
14022:212,165,213,105,000,133,002
14028:213,200,200,192,241,208,178
14034:226,096,173,006,055,208,206
14040:027,173,032,006,056,233,231
14046:040,074,074,074,074,074,120
14052:141,245,055,169,005,056,131
14058:237,245,055,024,109,244,124
14064:055,141,242,055,096,000,061
14070:000,000,000,000,000,000,246
14076:000,000,000,000,000,000,252
14082:000,000,000,000,000,000,002
14088:000,000,000,001,001,002,012
14094:002,000,000,001,001,002,020
14100:018,034,050,066,000,040,228
14106:072,104,136,168,072,096,162
14112:120,144,168,001,001,001,211
14118:001,001,001,001,001,001,044
14124:001,000,000,000,000,000,045
14130:000,000,000,000,000,255,049
14136:055,026,255,100,100,100,180
14142:144,100,000,008,002,000,060
14148:000,000,000,102,153,000,067
14154:000,000,000,000,195,036,049
14160:024,000,000,000,000,060,164
14166:052,255,245,255,060,060,245
14172:024,060,044,255,175,255,137
14178:060,060,024,016,056,016,074
14184:048,056,016,024,048,016,056
14190:056,016,052,056,168,068,014
14196:004,016,056,016,024,056,032
14202:016,048,024,016,056,016,042
14208:088,056,042,068,064,016,206
14214:056,016,124,124,040,040,022
14220:040,243,243,217,243,204,050
14226:243,193,217,051,035,047,164
14232:050,037,000,016,016,016,031
14238:016,016,000,000,045,037,016
14244:046,000,000,000,000,044,254
14250:037,054,037,044,000,000,086
14256:000,000,000,000,038,044,002
14262:047,047,050,000,000,000,070
14268:000,167,161,173,165,128,214
14274:175,182,165,178,112,114,096
14280:101,115,115,064,102,105,034
14286:114,101,098,117,116,116,100
14292:111,110,161,128,172,169,039
14298:180,180,172,165,128,168,187
14304:169,167,168,165,178,001,048
14310:010,100,232,000,000,000,060
14316:003,000,000,000,000,000,239
14322:000,000,000,000,000,000,242
14328:000,000,000,000,173,032,197
14334:006,201,172,144,104,173,030
14340:006,055,208,099,144,097,101
14346:162,032,169,001,133,217,212
14352:160,002,185,032,006,240,129
14358:010,201,180,176,006,056,139
14364:233,002,153,032,006,136,078
14370:016,238,165,210,024,105,024
14376:040,133,210,165,211,105,136
14382:000,133,211,032,171,054,135
14388:169,020,133,213,160,000,235
14394:136,208,253,198,213,208,250
14400:249,202,208,204,032,133,068
14406:056,032,149,051,032,171,049
14412:054,169,000,133,217,206,087
14418:244,055,206,244,055,206,068
14424:244,055,173,239,055,024,110
14430:105,100,141,239,055,173,139
14436:240,055,105,000,141,240,113
14442:055,096,096,169,000,133,143
14448:205,169,064,133,206,162,027
14454:031,160,000,152,145,205,043
14460:136,208,251,230,206,202,077
14466:016,246,096,169,000,133,022
14472:210,169,064,133,211,096,251
14478:173,034,006,208,007,173,231
14484:010,210,201,003,176,022,002
14490:173,006,055,240,014,206,080
14496:034,006,173,034,006,141,042
14502:000,210,208,006,076,030,184
14508:057,238,034,006,165,216,120
14514:073,128,133,216,016,052,028
14520:198,215,208,012,169,003,221
14526:133,215,173,066,006,073,088
14532:001,141,066,006,173,018,089
14538:006,205,016,006,208,017,148
14544:173,014,208,041,001,240,117
14550:021,169,001,141,006,055,095
14556:141,000,055,208,011,176,043
14562:006,238,018,006,076,236,038
14568:056,206,018,006,096,206,052
14574:004,055,240,014,173,004,216
14580:055,201,003,208,036,169,148
14586:000,141,000,210,240,029,102
14592:169,012,141,004,055,174,043
14598:005,055,189,141,055,141,080
14604:000,210,238,005,055,173,181
14610:005,055,201,008,208,005,244
14616:169,000,141,005,055,096,234
14622:104,104,076,162,048,173,185
14628:000,055,208,029,173,006,251
14634:055,208,024,162,005,202,186
14640:189,024,055,024,105,010,199
14646:205,032,006,176,244,232,181
14652:189,024,055,024,105,003,204
14658:141,032,006,096,000,000,085
```



The Apple version of "High Rise" is an exceptional arcade-style game.

Program 3: Apple High Rise

Version by Tim Victor, Editorial Programmer

```
START ADDRESS:  0801
END ADDRESS:    1BD8

0801: A9 00 85 EC A9 80 85 ED AD
0809: 20 DC 13 20 A4 1B 20 6F 06
0811: 17 A9 20 85 E6 8D E0 1B BD
0819: A9 00 8D D7 1B 8D D6 1B 05
0821: A9 28 8D DD 1B A9 C0 8D 2E
0829: DE 1B 20 C7 15 A9 40 85 45
0831: E6 20 C7 15 A9 00 8D FE 6E
0839: 1B 8D FF 1B 2C 57 C0 2C 58
0841: 52 C0 2C 54 C0 2C 50 C0 8D
0849: A9 70 8D F8 1B A9 0E 8D B4
0851: F9 1B A9 C4 8D FA 1B A9 DE
0859: 0E 8D FB 1B A9 00 8D 74 E1
0861: 0E 8D 81 0E 8D 8E 0E 8D 3D
0869: 9B 0E 8D A4 0E 8D B1 0E DE
0871: 8D BB 0E 8D C8 0E 8D F6 62
0879: 1B A9 0A 8D F4 1B A9 00 03
0881: 8D F3 1B A9 05 8D 9E 0E FC
0889: A9 02 8D 84 0E 8D 85 0E A8
0891: 8D 86 0E AD F6 1B F0 03 AF
0899: 4C 10 0A A9 01 8D E3 1B D0
08A1: A9 01 8D E2 1B A9 00 8D B3
08A9: E4 1B A9 56 8D E5 1B A9 71
08B1: 00 8D E6 1B A9 00 8D E7 04
08B9: 1B A9 01 8D EC 1B A9 00 E1
08C1: 8D ED 1B 8D EA 1B A9 13 7A
08C9: 8D EE 1B A9 00 8D F1 1B 8F
08D1: A9 06 8D EF 1B A9 00 8D F5
```

```
08D9: F0 1B AD F3 1B AC F4 1B AE
08E1: 8C F3 1B 6A 90 0D EE F3 C9
08E9: 1B A9 7C 8D E5 1B A9 39 82
08F1: 8D EE 1B A9 FF 8D E9 1B A7
08F9: A9 00 8D F7 1B 20 A4 14 C6
0901: 20 3C 0D 20 45 0C 20 81 F1
0909: 0D 20 61 1A 20 5A 17 20 30
0911: 33 09 20 C0 09 20 A4 14 35
0919: 20 6D 15 20 B1 19 20 A7 15
0921: 0B 20 09 0B 20 BD 0A 20 BE
0929: 7B 0A AD F7 1B F0 CE 4C 37
0931: 94 08 AD E5 1B A0 00 C9 C8
0939: A2 B0 06 69 26 C8 4C 38 45
0941: 09 98 18 6D F3 1B CD E8 68
0949: 1B F0 73 8D E8 1B A2 02 67
0951: 8E B4 0E C9 0A 90 08 EE C7
0959: B4 0E E9 0A 4C 54 09 69 56
0961: 02 8D B5 0E A9 00 8D B1 89
0969: 0E 98 D0 52 A2 02 FE 84 87
0971: 0E BD 84 0E C9 0C D0 08 93
0979: A9 02 9D 84 0E CA 10 EE 87
0981: A9 00 8D 81 0E 20 35 1B A8
0989: AD F3 1B D0 2B 20 05 0E D1
0991: AD F4 1B 18 69 04 8D F4 08
0999: 1B EE CC 0E AD CC 0E C9 F5
09A1: 0C D0 08 A9 02 8D CC 0E 77
09A9: EE CB 0E A9 00 8D C8 0E 58
09B1: A9 01 8D F7 1B 4C 01 0A 20
09B9: CE F3 1B 20 EA 15 60 AD AF
09C1: E5 1B 38 ED EE 1B C9 08 F2
09C9: 10 35 C9 EE 30 31 AD E6 E1
09D1: 1B 38 ED EF 1B C9 02 10 50
09D9: 26 C9 FF 30 22 20 47 1B AF
09E1: 20 CA 0D A9 01 8D F7 1B 3C
09E9: CE 9E 0E A9 00 8D 9B 0E E2
09F1: A9 02 CD 9E 0E D0 05 A9 64
09F9: 01 8D F6 1B 20 01 0A 60 F9
0A01: A9 18 8D 0F 0A 20 60 0A 2E
0A09: CE 0F 0A D0 F8 60 00 A9 89
0A11: 0A A0 49 20 FC 1A A9 80 A1
0A19: 8D 4D 0A 20 60 0A 2C 10 1E
0A21: C0 AD 00 C0 10 FB 2C 10 E5
0A29: C0 C9 CE F0 12 C9 EE F0 7F
0A31: 0E A9 40 8D 4D 0A 20 60 CA
0A39: 0A 20 60 0A 4C 0F 08 2C E1
0A41: 54 C0 2C 51 C0 4C D0 03 26
0A49: 20 2E 20 53 2E 34 07 1B 9D
0A51: 1D 10 1E 1E 01 19 01 1F 2B
0A59: 1A 01 1C 20 14 1F 00 20 7D
0A61: A4 14 20 6D 15 20 B1 19 4D
0A69: 20 A7 0B 20 A4 14 20 3C CC
0A71: 0D 20 81 0D 20 61 1A 4C 1C
0A79: 5A 17 AD E2 1B F0 36 10 7D
0A81: 1B AD E6 1B D0 07 AD E7 03
0A89: 1B C9 02 90 29 AD E7 1B D1
0A91: 38 E9 02 B0 1D 69 07 CE F2
0A99: E6 1B 10 16 AD E6 1B C9 54
0AA1: 1C B0 13 AD E7 1B 18 69 72
0AA9: 02 C9 07 90 05 EE E6 1B E7
0AB1: E9 07 8D E7 1B 60 A9 00 5A
0AB9: 8D E2 1B 60 2C E9 1B 30 26
0AC1: 0D AD E5 1B 18 69 1C 38 0D
0AC9: E9 26 90 2F D0 FA AD 00 2F
0AD1: C0 10 28 8D 10 C0 C9 8D CC
0AD9: F0 22 C9 A0 F0 13 C9 88 22
0AE1: F0 07 C9 95 D0 15 A9 01 F1
0AE9: 2C A9 FF A0 FF 8C E9 1B A9
0AF1: 2C A9 00 8D E2 1B A9 01 37
0AF9: 8D E3 1B 60 2C 10 C0 AD 08
0B01: 00 C0 10 FB 2C 10 C0 60 8C
0B09: AD EA 1B F0 03 4C 84 0B 80
0B11: AD E5 1B 18 69 09 CD EE 56
0B19: 1B D0 0A AD E6 1B CD EF 3C
0B21: 1B B0 38 90 39 90 03 A9 BC
0B29: 02 2C A9 FE 8D EB 1B AD 70
0B31: F1 1B 4A A9 0B 2C EB 1B E6
0B39: 10 04 90 06 B0 02 B0 02 BB
0B41: A9 F5 18 69 0E CD EF 1B E5
0B49: D0 0F A9 03 CD F0 1B D0 2A
0B51: 08 AD EB 1B 8D EA 1B D0 25
0B59: 2A 90 03 A9 02 2C A9 FE B6
0B61: 8D EB 1B 18 6D F0 1B C9 4D
0B69: 07 90 14 2C EB 1B 10 09 61
0B71: 18 69 07 CE EF 1B 4C 80 C0
0B79: 0B 38 E9 07 EE EF 1B 8D CB
0B81: F0 1B 60 18 6D EE 1B 8D 4F
0B89: EE 1B 18 69 13 38 E9 26 EA
0B91: 90 13 D0 FA A9 00 8D EA D1
0B99: 1B 2C EB 1B 10 04 CE F1 97
0BA1: 1B 60 EE F1 1B 60 A0 00 F5
0BA9: 8C FC 1B B9 C5 0D 30 09 10
0BB1: AA CA 8A 99 C5 0D 4C 39 EE
0BB9: 0C 18 B9 B1 0D 79 C0 0D 0B
0BC1: 99 B1 0D C9 06 F0 04 C9 15
0BC9: 9E D0 09 38 A9 00 F9 C0 0A
0BD1: 0D 99 C0 0D CC E9 1B D0 D2
0BD9: 0B B9 B1 0D 18 69 04 8D E6
0BE1: E5 1B D0 33 B9 B1 0D 18 C5
0BE9: 69 06 CD E5 1B 90 28 E9 A3
0BF1: 05 CD E5 1B B0 21 B9 B6 A0
0BF9: 0D CD E6 1B D0 19 B9 BB B2
0C01: 0D CD E7 1B D0 11 A9 00 DF
0C09: 8D E2 1B 8C E9 1B B9 B1 AD
0C11: 0D 18 69 04 8D E5 1B B9 17
0C19: B1 0D C9 9E F0 12 38 E9 9A
0C21: 06 F0 0D E9 26 90 11 D0 1F
0C29: FA 20 C6 18 29 E0 D0 08 97
0C31: 20 C6 18 4A 4A 99 C5 0D 04
0C39: C8 8C FC 1B C0 05 F0 03 29
0C41: 4C AC 0B 60 CE E3 1B D0 1F
0C49: 1C AD E2 1B F0 03 A9 02 D1
0C51: 2C A9 30 8D E3 1B EE E4 17
0C59: 1B AD E4 1B C9 04 D0 05 BD
0C61: A9 00 8D E4 1B AD E4 1B C2
0C69: 0A 0A 0A AE E2 1B F0 1C B6
0C71: 30 0D 69 FC 85 FA A9 00 4D
0C79: 69 0C 85 FB 4C 97 0C 69 FB
0C81: 1C 85 FA A9 00 69 0D 85 48
0C89: FB 4C 97 0C 69 DC 85 FA 2B
0C91: A9 00 69 0C 85 FB A0 00 C9
0C99: 8C FC 1B AD FC 1B 0A 0A E7
0CA1: A8 B1 FA 85 EE C8 B1 FA 2B
0CA9: 85 EF C8 18 AD E5 1B 71 C7
0CB1: FA 8D D7 1B C8 18 AD E6 40
0CB9: 1B 8D D6 1B AD E7 1B 71 04
0CC1: FA C9 07 90 05 EE D6 1B 60
0CC9: E9 07 8D D5 1B 20 C0 14 96
0CD1: EE FC 1B AD FC 1B C9 02 C8
0CD9: 90 C1 60 74 12 01 01 92 27
0CE1: 12 09 01 B0 12 01 01 92 99
0CE9: 12 09 01 74 12 01 01 92 DD
0CF1: 12 09 01 B0 12 01 01 92 A9
0CF9: 12 09 01 74 12 02 01 CE 2E
0D01: 12 0A 00 74 12 01 01 EC 71
0D09: 12 09 04 74 12 02 01 0A DA
0D11: 13 0A 00 74 12 01 01 EC 02
0D19: 12 09 04 B0 12 02 01 28 CC
0D21: 13 0A 05 B0 12 01 01 46 CF
0D29: 13 09 05 B0 12 02 01 64 B9
0D31: 13 0A 04 B0 12 01 01 46 BF
0D39: 13 09 05 CE EC 1B D0 0D 2F
0D41: A9 14 8D EC 1B A9 01 4D 84
0D49: ED 1B 8D ED 1B AD ED 1B 38
0D51: F0 02 A9 1E 18 69 82 85 6C
0D59: EE A9 13 85 EF 90 02 E6 BC
0D61: EF AD EF 1B 8D D6 1B AD 3A
0D69: F0 1B 8D D5 1B AD EE 1B 5A
0D71: 38 ED ED 1B ED ED 1B 69 59
0D79: 01 8D D7 1B 20 C0 14 60 B0
0D81: A9 00 8D FC 1B A9 BE 85 74
0D89: EE A9 13 85 EF AC FC 1B 87
0D91: C0 05 F0 1B B9 B1 0D 8D 59
0D99: D7 1B B9 B6 0D 8D D6 1B 70
0DA1: B9 BB 0D 8D D5 1B 20 C0 1E
0DA9: 14 EE FC 1B 4C 8E 0D 60 F1
0DB1: 10 30 50 70 90 00 07 0E 91
0DB9: 14 1B 04 01 00 06 01 FE 4E
0DC1: 02 FE 02 FE FF FF FF FF CC
0DC9: FF 20 A4 14 20 6D 15 20 C2
0DD1: B1 19 A9 FF 8D E9 1B AD 38
0DD9: E5 1B 18 69 08 8D E5 1B A4
0DE1: 20 A7 0B 20 A4 14 20 3C 4B
0DE9: 0D AD E5 1B C9 A8 B0 03 B9
0DF1: 20 45 0C 20 81 0D 20 61 D2
0DF9: 1A 20 5A 17 AD E5 1B C9 EA
0E01: A8 90 C6 60 A9 FF 8D E2 BF
0E09: 1B A9 01 8D E3 1B A0 59 3C
0E11: A9 0E 20 FC 1A A9 B0 8D 5F
0E19: 5D 0E 20 A4 14 20 6D 15 C6
0E21: 20 B1 19 20 A7 0B A9 FF 9B
0E29: 8D E9 1B 20 7B 0A 20 A4 D4
0E31: 14 20 3C 0D AD E2 1B F0 D7
0E39: 03 20 45 0C 20 81 0D 20 89
0E41: 61 1A 20 5A 17 AD E2 1B 8E
0E49: D0 D0 A9 0A 8D F3 1B AD F7
0E51: 5D 0E 09 40 8D 5D 0E 60 23
0E59: 4F 7F 55 45 20 34 07 1D F8
0E61: 1A 20 19 0F 01 0E 1A 18 33
0E69: 1B 17 10 1F 10 0F 00 7D 07
0E71: 0E F8 1B 00 0E 20 1E 0E 71
0E79: 1A 1D 10 00 8A 0E 70 0E 67
0E81: 00 1A 20 02 02 02 02 02 66
0E89: 00 97 0E 7D 0E 00 3C 20 2E
0E91: 17 14 21 10 1E 00 A0 0E A3
0E99: 8A 0E 00 48 22 02 00 AD C9
0EA1: 0E 97 0E 00 6A 20 11 17 79
0EA9: 1A 1A 1D 00 B7 0E A0 0E 42
0EB1: 00 76 22 02 03 00 C4 0E 7F
0EB9: AD 0E 00 98 20 17 10 21 58
0EC1: 10 17 00 00 00 B7 0E 00 A6
0EC9: A4 22 02 03 00 A0 81 80 37
0ED1: B0 83 80 B0 8D 80 E0 83 F5
0ED9: 80 B0 81 80 F0 83 80 E0 12
0EE1: 80 80 C0 81 80 A0 81 80 98
0EE9: B0 83 80 AC 83 80 F0 81 9B
0EF1: 80 A0 83 80 F0 83 80 C0 46
0EF9: 81 80 E0 80 80 9C 8E 80 2F
0F01: BE 9F 80 F6 9B 80 F6 9B 4E
0F09: 80 E6 99 80 8C 8C 80 E0 D4
0F11: 81 80 F0 83 80 B0 83 80 B4
0F19: B0 83 80 B0 83 80 B0 83 8E
0F21: 80 B0 83 80 B0 83 80 90 49
0F29: 82 80 B0 80 80 D8 81 80 B1
0F31: D8 83 80 B0 83 80 C0 82 D9
0F39: 80 F0 83 80 F0 81 80 E8 C3
0F41: 81 80 D8 83 80 9C 87 80 99
0F49: 8E 87 80 C4 83 80 98 81 BD
0F51: 80 80 86 80 E0 80 80 F0 A3
0F59: 81 80 D0 83 80 B0 83 80 F8
0F61: C0 84 80 F0 83 80 F0 81 A1
0F69: 80 F8 81 80 F8 82 80 9C AD
0F71: 87 80 8E 87 80 C4 83 80 5C
0F79: 98 81 80 80 86 80 B0 80 74
0F81: 80 F8 80 80 DC 80 80 EC 0D
0F89: 80 80 92 80 80 FC 80 80 DB
0F91: F8 80 80 F8 81 80 F4 81 65
0F99: 80 CE 83 80 8E 87 80 9C 54
0FA1: 82 80 C8 81 80 86 80 80 F1
0FA9: E0 80 80 D8 81 80 DC 81 3F
0FB1: 80 EC 80 80 94 80 80 FC 08
0FB9: 80 80 F8 80 80 B8 81 80 C9
0FC1: DC 81 80 CE 83 80 8E 87 6E
0FC9: 80 9C 82 80 C8 81 80 86 7B
0FD1: 80 80 86 80 8F 80 9B 80 5F
0FD9: 96 80 9C 80 9B 80 8F 80 7D
0FE1: 8F 80 8F 80 8F 80 9F 80 FF
0FE9: 87 80 87 80 82 80 8C 80 94
0FF1: 98 80 BC 80 B6 80 9A 80 89
0FF9: 8E 80 B6 80 BC 80 BC 80 40
1001: BC 80 BC 80 BC 80 B8 80 19
1009: B8 80 90 80 8C 80 F0 83 8B
1011: 80 F8 87 80 F8 9F 80 F0 E0
1019: 87 80 F8 87 80 F8 87 80 2C
1021: F0 83 80 E0 83 80 F0 83 3C
1029: 80 F8 87 80 FE 87 80 F8 D0
1031: 83 80 F8 87 80 F8 87 80 42
1039: F0 83 80 F0 81 80 FE 9F 7D
1041: 80 FF BF 80 FF BF 80 FF A1
1049: BF 80 FF BF 80 FE 9F 80 25
1051: F0 83 80 F8 87 80 F8 87 22
1059: 80 F8 87 80 F8 87 80 F8 D0
1061: 87 80 F8 87 80 F8 87 80 74
1069: B8 87 80 FC 80 80 FC 83 2B
1071: 80 FC 87 80 F8 87 80 F8 E9
1079: 87 80 F8 87 80 F8 83 80 84
1081: FC 83 80 FC 87 80 FE 8F AC
1089: 80 DF 8F 80 EE 87 80 FC 6F
1091: 83 80 80 8F 80 F0 81 80 E7
1099: F8 83 80 F8 87 80 F8 87 6E
10A1: 80 F8 8F 80 F8 87 80 F8 1A
10A9: 83 80 FC 83 80 FC 87 80 0B
10B1: FE 8F 80 DF 8F 80 EE 87 27
10B9: 80 FC 83 80 80 8F 80 F8 0E
10C1: 80 80 FC 81 80 FE 81 80 7D
10C9: FE 81 80 FF 81 80 FE 81 67
10D1: 80 FC 81 80 FC 83 80 FE 9F
10D9: 83 80 FF 87 80 DF 8F 80 77
10E1: BE 87 80 FC 83 80 8F 80 E0
10E9: 80 F0 81 80 FC 83 80 FE B4
10F1: 83 80 FE 81 80 FE 81 80 6F
10F9: FE 81 80 FC 81 80 FC 83 65
```

```
1101: 80 FE 83 80 FF 87 80 DF 9A
1109: 8F 80 BE 87 80 FC 83 80 E2
1111: 8F 80 80 9E 80 BF 80 FF 19
1119: 80 FE 80 FC 80 FF 80 BF DF
1121: 80 BF 80 BF 80 BF 80 BF 43
1129: 80 9F 80 9F 80 8E 80 BC 79
1131: 80 98 80 BC 80 B6 80 9A 10
1139: 80 8E 80 B6 80 BC 80 BC 6F
1141: 80 BC 80 BC 80 BC 80 B8 5F
1149: 80 B8 80 90 80 8C 80 94 BE
1151: 80 9C 80 A2 80 9C 80 9C 29
1159: 80 AA 80 9C 80 AA 80 9C 8C
1161: 80 AA 80 9C 80 AA 80 94 8C
1169: 80 BE 80 BE 80 FF 80 BE 3B
1171: 80 BE 80 FF 80 BE 80 FF 93
1179: 80 BE 80 FF 80 BE 80 FF 9B
1181: 80 BE 80 94 80 9C 80 A2 07
1189: 80 9C 80 88 80 88 80 9C 6F
1191: 80 AA 80 9C 80 AA 80 9C C4
1199: 80 AA 80 9C 80 AA 80 94 C4
11A1: 80 BE 80 BE 80 FF 80 BE 73
11A9: 80 9C 80 9C 80 BE 80 FF 0D
11B1: 80 BE 80 FF 80 BE 80 FF D3
11B9: 80 BE 80 FF 80 BE 80 7E 5A
11C1: 7F 00 7E 7F 00 06 60 00 44
11C9: 06 60 00 06 60 00 06 60 D6
11D1: 00 06 60 00 06 60 00 06 39
11D9: 60 00 06 60 00 06 60 00 CB
11E1: 06 60 00 06 60 00 06 60 EE
11E9: 00 06 60 00 06 60 00 06 51
11F1: 60 00 06 60 00 06 60 00 E3
11F9: 06 60 00 06 60 00 06 60 07
1201: 00 06 60 00 06 60 00 06 6A
1209: 60 00 06 60 00 06 60 00 FC
1211: 06 60 00 7E 7F 00 7E 7F B0
1219: 00 7F 7F 01 7F 7F 01 0F 28
1221: 70 01 0F 70 01 0F 70 01 CC
1229: 0F 70 01 0F 70 01 0F 70 18
1231: 01 0F 70 01 0F 70 01 0F 03
1239: 70 01 0F 70 01 0F 70 01 E4
1241: 0F 70 01 0F 70 01 0F 70 30
1249: 01 0F 70 01 0F 70 01 0F 1B
1251: 70 01 0F 70 01 0F 70 01 FC
1259: 0F 70 01 0F 70 01 0F 70 48
1261: 01 0F 70 01 0F 70 01 0F 33
1269: 70 01 0F 70 01 7F 7F 01 F4
1271: 7F 7F 01 CE 0E 20 20 20 93
1279: 20 20 20 2E 20 53 2E 20 67
1281: 20 03 08 27 10 2E 20 53 B6
1289: 52 20 7F 54 20 20 20 00 D5
1291: 20 FE 0E 20 53 20 20 3A DE
1299: 20 20 20 20 20 20 20 03 A0
12A1: 0F 3F 10 53 20 20 20 53 69
12A9: 20 20 50 2E 20 00 44 E6 43
12B1: 0E 00 20 20 45 2E 20 20 26
12B9: 20 20 20 20 20 03 08 0F 28
12C1: 10 20 53 20 4A 20 20 20 95
12C9: 20 41 20 20 20 2B 0F 20 40
12D1: 53 20 4A 20 41 53 20 56 E0
12D9: 20 48 20 03 0E 6C 10 7F 16
12E1: 20 20 20 20 20 20 36 20 32
12E9: 20 2E 20 D3 0F 20 20 20 44
12F1: 20 20 20 20 7F 20 20 4F 40
12F9: 20 02 0F 14 11 20 20 20 3B
1301: 00 20 20 20 50 33 20 33 F7
1309: 20 55 0F 45 20 20 00 41 8D
1311: 7F 20 20 49 20 44 20 03 EC
1319: 0E 96 10 20 32 4E 20 20 1B
1321: 20 20 20 20 20 20 20 7F A6
1329: 0F 43 20 20 03 20 20 20 A6
1331: 20 20 20 20 20 03 0E C0 5F
1339: 10 20 33 20 20 20 20 41 DA
1341: 20 33 20 33 20 F1 0F 20 82
1349: 53 43 20 00 55 20 20 49 A2
1351: 7F 20 00 02 0F 32 11 20 E2
1359: 20 00 20 20 20 20 49 53 FC
1361: 20 20 20 A9 0F 20 2E 20 B3
1369: 52 2E 20 20 20 20 20 20 2C
1371: 20 03 0E EA 10 20 7F 54 2D
1379: 20 20 48 00 20 7F 20 52 52
1381: 20 50 11 20 20 20 20 20 D1
1389: 20 20 53 53 45 20 45 02 9E
1391: 0D 6A 11 32 20 20 20 20 FF
1399: 3B 20 20 20 20 53 2E 84 9A
13A1: 11 20 20 20 00 53 20 20 0C
13A9: 20 7F 20 7F 20 02 0F A2 85
13B1: 11 20 53 43 52 20 20 53 AD
13B9: 20 20 00 20 00 C0 11 20 3F
13C1: 20 20 7F 20 7F 20 52 2E 41
13C9: 20 33 20 03 1E 1A 12 20 9E
13D1: 20 20 20 20 20 20 2E 20 14
13D9: 52 2E 20 A9 00 8D FC 1B 9E
13E1: A9 74 85 EE A9 12 85 EF 2A
13E9: AD FC 1B C9 0C F0 2F A0 49
13F1: 0E B1 EE 8D DD 1B C8 B1 E0
13F9: EE 8D DE 1B 20 20 14 A5 D7
1401: EE 18 69 10 85 EE 90 02 DF
1409: E6 EF 20 20 14 18 A5 EE E1
1411: 69 0E 85 EE 90 02 E6 EF 5B
1419: EE FC 1B 4C E9 13 60 A9 26
1421: 01 8D D5 1B AD D5 1B 0A 9E
1429: A8 A5 EC 91 EE C8 A5 ED 99
1431: 91 EE A0 00 B1 EE 85 FC 43
1439: C8 B1 EE 85 FD AD DE 1B E7
1441: 8D DF 1B A0 00 8C DB 1B 9A
1449: A9 00 8D DC 1B B1 FC 8D EC
1451: D4 1B 0A AE D5 1B 0A 2E 34
1459: DC 1B CA D0 F9 2C D4 1B 62
1461: 10 02 38 24 18 6A 0D DB BB
1469: 1B 91 EC AD DC 1B 8D DB 46
1471: 1B C8 CC DD 1B D0 D1 18 A8
1479: A5 FC 6D DD 1B 85 FC 90 B8
1481: 02 E6 FD 18 A5 EC 6D DD 3F
1489: 1B 85 EC 90 02 E6 ED CE 9D
1491: DF 1B D0 AF AC D5 1B C8 41
1499: 8C D5 1B C0 07 F0 03 4C 3B
14A1: 25 14 60 A5 E6 C9 20 D0 37
14A9: 08 A9 03 A0 1C A2 00 F0 07
14B1: 06 A9 53 A0 1C A2 01 8E B7
14B9: FD 1B 85 1E 84 1F 60 A0 3C
14C1: 0E B1 EE 8D DD 1B C8 B1 B2
14C9: EE 8D DE 1B A0 00 AD D7 92
14D1: 1B 91 1E C8 AD D6 1B 91 CC
14D9: 1E C8 AD D5 1B 91 1E C8 7A
14E1: A5 EE 91 1E C8 A5 EF 91 FA
14E9: 1E C8 AD DE 1B 91 1E C8 1B
14F1: AD DD 1B 91 1E A5 1E 18 C0
14F9: 69 08 85 1E 90 02 E6 1F E4
1501: AE FD 1B FE FE 1B AD D5 EA
1509: 1B 0A 65 EE 85 FC A5 EF 3A
1511: 69 00 85 FD 18 A5 FC 69 3B
1519: 10 85 1A A5 FD 69 00 85 65
1521: 1B A0 00 B1 FC AA C8 B1 F1
1529: FC 86 FC 85 FD A0 00 B1 8F
1531: 1A AA C8 B1 1A 86 1A 85 EB
1539: 1B AD D7 1B 8D DA 1B AD C4
1541: DE 1B 8D DF 1B 20 EC 16 9A
1549: 20 B5 15 EE DA 1B A5 FC 0E
1551: 18 6D DD 1B 85 FC 90 02 93
1559: E6 FD A5 1A 18 6D DD 1B 1A
1561: 85 1A 90 02 E6 1B CE DF 28
1569: 1B D0 DA 60 AE FD 1B BD 18
1571: FE 1B F0 3F A0 00 B1 1E 7A
1579: 8D D7 1B C8 B1 1E 8D D6 48
1581: 1B C8 B1 1E 8D D5 1B C8 46
1589: B1 1E 85 EE C8 B1 1E 85 82
1591: EF C8 B1 1E 8D DE 1B C8 E4
1599: B1 1E 8D DD 1B 20 C7 15 B1
15A1: A5 1E 18 69 08 85 1E 90 E2
15A9: 02 E6 1F AE FD 1B DE FE 76
15B1: 1B D0 C1 60 AC DD 1B 88 77
15B9: B1 FE 31 1A 51 FE 11 FC E9
15C1: 91 FE 88 10 F3 60 AD D7 DA
15C9: 1B 8D DA 1B AD DE 1B 8D 9E
15D1: DF 1B 20 EC 16 AC DD 1B BF
15D9: 88 B1 1C 91 FE 88 10 F9 85
15E1: EE DA 1B CE DF 1B D0 EA 82
15E9: 60 A9 00 8D F5 1B A9 04 FA
15F1: 8D D8 1B AD D8 1B 8D D7 7D
15F9: 1B A9 00 8D D9 1B A9 BE 42
1601: 38 ED D8 1B 8D DE 1B 20 CF
1609: B2 16 A9 26 8D D7 1B 38 E5
1611: A9 BE ED D8 1B 8D D9 1B EA
1619: AD D8 1B 8D DE 1B 20 B2 E4
1621: 16 AD F5 1B F0 23 A9 00 9B
1629: 8D DA 1B 8D D6 1B A9 C0 46
1631: 8D DF 1B 20 EC 16 A0 1D 9F
1639: B1 FE 91 1C 88 10 F9 EE 59
1641: DA 1B CE DF 1B D0 EC F0 60
1649: 12 38 AD E5 1B E9 04 8D B6
1651: E5 1B 38 AD EE 1B E9 04 D4
1659: 8D EE 1B 20 A4 14 AE FD 3E
1661: 1B A9 00 9D FE 1B 20 3C 40
1669: 0D 20 45 0C 20 81 0D 20 CE
1671: 61 1A 20 5A 17 AD D8 1B BA
1679: C9 26 B0 22 69 04 8D D8 9B
1681: 1B 38 E9 26 90 15 48 18 6A
1689: 6D E5 1B 8D E5 1B 68 18 A6
1691: 6D EE 1B 8D EE 1B A9 26 C9
1699: 8D D8 1B 4C F4 15 AD F5 38
16A1: 1B D0 08 A9 01 8D F5 1B 70
16A9: 4C F4 15 A9 FF 8D E9 1B 9B
16B1: 60 AD DE 1B 8D DF 1B A9 D2
16B9: 00 8D D6 1B AD D7 1B 8D 66
16C1: DA 1B 20 EC 16 A5 1C 85 F9
16C9: FC A5 1D 85 FD AD D9 1B 4F
16D1: 8D DA 1B 20 EC 16 A0 1D FE
16D9: B1 FC 91 FE 88 10 F9 EE A7
16E1: D7 1B EE D9 1B CE DF 1B 2B
16E9: D0 D2 60 AD DA 1B 29 3F EE
16F1: A8 B9 1A 17 05 E6 85 FF 64
16F9: 09 60 85 1D AD DA 1B 29 7D
1701: 08 F0 02 A9 80 18 2C DA E1
1709: 1B 70 04 10 04 69 28 69 E1
1711: 28 6D D6 1B 85 FE 85 1C 8A
1719: 60 00 04 08 0C 10 14 18 59
1721: 1C 00 04 08 0C 10 14 18 3F
1729: 1C 01 05 09 0D 11 15 19 C6
1731: 1D 01 05 09 0D 11 15 19 4F
1739: 1D 02 06 0A 0E 12 16 1A D6
1741: 1E 02 06 0A 0E 12 16 1A 5F
1749: 1E 03 07 0B 0F 13 17 1B E6
1751: 1F 03 07 0B 0F 13 17 1B 6F
1759: 1F A5 E6 C9 40 A9 00 2A CC
1761: AA BD 54 C0 A5 E6 8D E0 AF
1769: 1B 49 60 85 E6 60 A9 60 48
1771: 85 E6 20 F2 F3 A9 00 8D 23
1779: DA 1B 8D D6 1B 20 EC 16 44
1781: A9 00 8D FC 1B 20 B0 17 D7
1789: C9 05 D0 F9 EE DA 1B EE 9F
1791: DA 1B 20 EC 16 20 B0 17 70
1799: C9 0A D0 F9 EE DA 1B EE F0
17A1: DA 1B AD DA 1B C9 BA B0 8D
17A9: 03 4C 7E 17 4C D9 17 AC 52
17B1: FC 1B B9 C5 17 48 B9 CF D5
17B9: 17 A8 68 91 FE EE FC 1B 8C
17C1: AD FC 1B 60 08 01 40 20 54
17C9: 01 20 04 02 01 04 01 08 43
17D1: 0E 15 1C 01 08 0F 16 1C A4
17D9: A9 00 8D D6 1B 8D FC 1B 20
17E1: AC FC 1B C0 05 F0 18 B9 EA
17E9: 59 18 8D DA 1B 20 03 18 A1
17F1: 20 15 18 20 15 18 20 03 C6
17F9: 18 EE FC 1B 4C E1 17 4C A5
1801: 5E 18 20 EC 16 A0 00 B9 26
1809: 4F 18 91 FE C8 C0 02 D0 27
1811: F6 4C 24 18 20 EC 16 A0 57
1819: 00 B9 51 18 91 FE C8 C0 3E
1821: 08 D0 F6 A5 FE A6 FF 85 DA
1829: FC 86 FD 98 18 65 FE 85 9C
1831: FE A0 00 B1 FC 91 FE C8 19
1839: 98 18 65 FE 29 7F C9 1E 51
1841: F0 08 C9 46 F0 04 C9 6E 23
1849: D0 E9 EE DA 1B 60 D5 AA 98
1851: D0 A0 C1 82 85 8A 94 A8 9A
1859: 22 48 6E 94 BA A9 19 8D FF
1861: D6 1B A9 00 8D FC 1B 8D 1D
1869: DA 1B AC FC 1B C0 05 F0 0A
1871: 1B 20 92 18 38 A9 1C ED 99
1879: D6 1B 8D D6 1B 18 AD DA 6A
1881: 1B 69 06 8D DA 1B EE FC 51
1889: 1B 4C 6B 18 60 84 82 D0 34
1891: 80 A2 00 A9 8E 85 FC A9 73
1899: 18 85 FD A9 01 8D DE 1B A8
18A1: 20 EC 16 A0 01 B1 FC 91 44
18A9: FE 88 10 F9 18 A5 FC 69 D7
18B1: 02 85 FC 90 02 E6 FD EE 83
18B9: DA 1B CE DE 1B 10 E1 E8 AB
18C1: E0 10 D0 CF 60 AD E1 1B 16
18C9: 0A 0A 38 6D E1 1B 8D E1 D7
18D1: 1B 60 80 BC 98 BC BC B0 65
18D9: FE BC FE BC BC FC BE BC 78
18E1: BE FE FE BC E6 98 E0 E6 1F
18E9: 86 E6 BE BC BE BC BE BC DD
18F1: FE E6 E6 E6 E6 E6 FE 80 F7
18F9: E6 9C E6 E6 B8 86 86 E0 DD
1901: E6 E6 E6 E6 E6 E6 86 86 12
1909: E6 E6 98 E0 E6 86 FE E6 BF
1911: E6 E6 E6 E6 E6 98 E6 E6 0A
1919: E6 E6 E6 B0 80 F6 98 B0 22
1921: B0 B4 BE BE B0 BC E6 E6 C9
```

```
1929: E6 86 E6 86 86 86 E6 98 6A
1931: E0 B6 86 E6 E6 E6 E6 48
1939: E6 8C 98 E6 E6 E6 E6 E6 0B
1941: 98 80 EE 98 8C E0 FE E0 0E
1949: E6 98 E6 FC FE BE 86 E6 A8
1951: BE BE F6 FE 98 E0 9E 86 6D
1959: E6 E6 E6 BE E6 BE B0 98 AD
1961: E6 E6 E6 BC BC 8C 80 E6 69
1969: 98 E6 E6 B0 E6 E6 8C E6 5C
1971: B0 E6 E6 E6 E6 86 86 E6 46
1979: E6 98 E6 E6 86 E6 E6 E6 15
1981: 86 B6 E6 E6 98 E6 E6 FE 1D
1989: E6 98 86 80 BC BC FE BC C1
1991: B0 BC BC 8C BC 98 E6 FE C0
1999: BE BE FE 86 BE E6 98 BC A2
19A1: E6 FE E6 E6 BC 86 EC E6 13
19A9: BE 98 BE 98 E6 E6 98 FE C5
19B1: AD F8 1B 85 EE AD F9 1B F1
19B9: 85 EF A5 EE 05 EF D0 01 D8
19C1: 60 A0 04 A9 01 AE FD 1B 41
19C9: F0 01 0A 31 EE F0 4B B1 8C
19D1: EE 30 05 2C 34 1B F0 4D 28
19D9: C8 B1 EE 8D DA 1B C8 B1 1A
19E1: EE 8D D6 1B A5 EE 18 69 FD
19E9: 07 85 FC A5 EF 69 00 85 A5
19F1: FD A9 06 8D DF 1B 20 EC BF
19F9: 16 A0 00 B1 1C 91 FE C8 68
1A01: B1 FC D0 F7 EE DA 1B CE CE
1A09: DF 1B D0 EA A9 01 AE FD 69
1A11: 1B F0 01 0A A0 04 51 EE 76
1A19: 91 EE B1 EE 2C 34 1B F0 50
1A21: 04 29 03 F0 06 20 EF 1A BB
1A29: 4C BB 19 A0 02 B1 EE 85 D9
1A31: FC C8 B1 EE 85 FD 20 EF 8F
1A39: 1A A0 00 A5 EE 91 FC C8 7D
1A41: A5 EF 91 FC C8 05 EE D0 4F
1A49: 0B A5 FC 8D FA 1B A5 FD 72
1A51: 8D FB 1B 60 A5 FC 91 EE E7
1A59: C8 A5 FD 91 EE 4C BB 19 6D
1A61: AD F8 1B 85 EE AD F9 1B A3
1A69: 85 EF A5 EE 05 EF D0 01 8A
1A71: 60 A0 04 B1 EE 2C 34 1B 45
1A79: D0 6E C9 00 30 0C A9 01 F0
1A81: AE FD 1B F0 01 0A 31 EE 80
1A89: D0 5E C8 B1 EE 8D DA 1B 70
1A91: C8 B1 EE 8D D6 1B A9 D2 96
1A99: 8D BE 1A A9 18 8D BF 1A B2
1AA1: A9 06 8D DF 1B A5 EE 18 41
1AA9: 69 07 85 FC A5 EF 69 00 94
1AB1: 85 FD 20 EC 16 A0 00 B1 DF
1AB9: FC F0 09 AA BD FF FF 91 F3
1AC1: FE C8 D0 F3 EE DA 1B AD C7
1AC9: BE 1A 18 69 25 8D BE 1A 74
1AD1: 90 03 EE BF 1A CE DF 1B CF
1AD9: D0 D8 A9 01 AE FD 1B F0 86
1AE1: 01 0A A0 04 11 EE 91 EE C3
1AE9: 20 EF 1A 4C 6B 1A A0 00 37
1AF1: B1 EE AA C8 B1 EE 86 EE E1
1AF9: 85 EF 60 48 98 48 AD FA B9
1B01: 1B 85 EE AD FB 1B 85 EF 26
1B09: A0 00 68 91 EE C8 68 91 B2
1B11: EE 20 EF 1A A9 00 A8 91 96
1B19: EE C8 91 EE C8 AD FA 1B 28
1B21: 91 EE C8 AD FB 1B 91 EE 2E
1B29: A5 EE 8D FA 1B A5 EF 8D 2C
1B31: FB 1B 60 40 A9 E8 8D 87 CF
1B39: 1B A9 F4 8D 7D 1B A9 1E A8
1B41: 8D 6A 1B 4C 56 1B A9 E0 54
1B49: 8D 87 1B A9 E8 8D 7D 1B B9
1B51: A9 1F 8D 6A 1B A9 01 8D 8B
1B59: 00 1C A0 00 AD 87 1B 8D F9
1B61: 02 1C 4E 00 1C 90 0C B9 5E
1B69: 00 1E C8 8D 01 1C A9 80 65
1B71: 8D 00 1C 4E 01 1C 90 03 73
1B79: AD 30 C0 A2 FF E8 D0 FD 18
1B81: 90 03 AD 30 C0 A2 FF E8 F2
1B89: D0 FD EE 02 1C D0 D3 18 89
1B91: AD 87 1B E9 01 8D 87 1B EA
1B99: AD 7D 1B 69 01 8D 7D 1B 54
1BA1: 90 BA 60 A9 88 8D 00 1E 0E
1BA9: A0 01 B9 FF 1D 99 00 1E 15
1BB1: C8 D0 F7 A9 80 A0 03 99 40
1BB9: 00 1F 88 10 FA A9 AA A0 3E
1BC1: 03 99 04 1F 88 10 FA A0 6D
1BC9: 08 B9 F8 1E 99 00 1F C8 47
1BD1: D0 F7 60 00 00 00 00 00 7A
```

©

High-Speed String Search
For Atari BASIC

Tom R. Halfhill, Editor

Here's a short machine language routine that adds a valuable new function to Atari BASIC—a high-speed search that can find any string of characters within a larger string almost instantly. It lets you add machine language speed to BASIC databases and sorts, and a sample address book program shows how. For all 400/800, XL, and XE computers.

Have you ever dreamed up a wish list of new commands for the ultimate BASIC language? Ideally, this super-BASIC would combine in a single package the best features that all BASICs have to offer.

A sure way to collect ideas for this wish list is to look at other BASICs. For instance, IBM BASIC, Amiga BASIC, Atari ST BASIC, and other large BASIC languages have a function called INSTR (pronounced "in-string"). INSTR rapidly searches through a character string and returns the position of any substring you specify. Once you know the position of the substring, it's a simple matter to retrieve it with the usual BASIC string statements.

Atari BASIC lacks an INSTR function, yet needs it much more than most other BASICs do. Many BASIC languages don't allow strings longer than 255 characters, so you can write a search routine in BASIC that's not significantly slower than BASIC's own INSTR function. But Atari BASIC allows character strings of virtually *any* length, up to the limit of available memory. Although you can write a search routine in Atari BASIC that simulates INSTR, it would take ages to find a substring hidden near the end of a really long string.

The answer, as usual, is to mix BASIC with a dash of machine language: an ML routine that duplicates INSTR and works in a flash.

When this powerful function is combined with the megastring capability of Atari BASIC, all kinds of possibilities arise—database and sorting programs written in BASIC that perform at near-ML speeds, and simulated string arrays that really let you retrieve any substring as fast as true Microsoft-style string arrays. With the INSTR routine accompanying this article, it's a snap.

No Memory Confusion

Take a look at Program 1. It's a BASIC loader that encodes the machine language INSTR routine in DATA statements, ready to merge with your own programs. Because the ML is written to be completely relocatable, you can add this routine to any BASIC program without worrying about memory conflicts—it avoids such overused memory areas as page 6. Line 10, which should be near the beginning of your program, reads the machine language into a string, ML$. Then it uses Atari BASIC's string-address function (ADR) to set the variable ML equal to the starting address of ML$. After this setup, all it takes to call the INSTR routine is a simple USR statement with a few arguments arranged in this format:

```
INSTR=USR(ML,ADR(XX$),LEN(XX$),
  ADR(SUB$),LEN(SUB$),START)
```

where *XX$* is the larger character string you're searching through, and *SUB$* is the smaller substring you want to find. The result, returned in the variable INSTR, is the position within *XX$* of the first character in *SUB$*. (Of course, you may use any variable names you prefer in the USR statement, as long as the statement conforms to this general format.)

For instance, if XX$ contains these characters:

```
ABCDEFGHELLOABCDEFG
```

and if SUB$ contains these characters:

```
HELLO
```

the result of calling the INSTR routine would be INSTR=8, because the substring HELLO begins at the eighth character position in XX$. If you redefine SUB$—say, SUB$="DEF"—and call the routine again, the result would be INSTR=4, because the substring DEF begins at the fourth character position in XX$.

The rest is easy. Once you know a substring's position within a larger string, you can retrieve it with a statement like this:

```
PRINT XX$(INSTR,INSTR+LEN
  (SUB$)-1)
```

To make the Atari INSTR routine even more useful, it has one additional feature. Looking again at the USR statement above, you'll notice another argument that wasn't mentioned. This argument, START, also was inspired by the INSTR function in larger BASICs. It lets you specify the starting point of the search within XX$.

Normally, if you're searching through the entire string, you'd set START=1 before calling the INSTR routine. But there may be times when you want to start the search elsewhere within XX$. A prime example is when you're searching for more than one occurrence of a substring. If the search always began at the first character in XX$, the INSTR routine would find only the first occurrence of SUB$ every time. To get around this, all you have to do is call the routine again after executing a statement such as START=INSTR+1. This starts the next search at a position which is one character past the point where the previous search stopped. You can repeat this procedure to find as many occurrences of the substring as you want.

A Speed Test

When programming in BASIC, it's nearly impossible to cause a system crash or lockup unless you're guilty of a wayward POKE. But in machine language, unrecoverable crashes are much more common. Therefore, the INSTR routine is carefully error-trapped. If it does not find the substring you specify, it returns a zero. If you include the wrong number of arguments in the USR statement, the routine clears the 6502 stack of all faulty arguments before returning to BASIC, then returns a zero.

The only limitation to keep in mind when using the INSTR routine is that the substring you're looking for cannot be longer than 255 characters—not a serious limitation. The string you're searching through can be any length, of course.

So, just how fast *is* the INSTR routine? Wickedly fast. For a demonstration, enter Program 2. It creates a monster string that is 30,000 characters long—almost all of the usable program memory that's left in a 48K or 64K Atari when DOS and BASIC are active. (There may not be enough memory to run this program if you're using a non-Atari DOS that uses more RAM.) This string is filled entirely with X's, except for a Z at the thirty-thousandth position. When you run Program 2, it uses a search routine written in BASIC to find the Z. Prepare yourself for a long wait. It takes almost eight minutes.

Now add the lines in Program 3 to the INSTR routine in Program 1 and repeat the test. INSTR finds the Z and prints it on the screen in about two seconds. With strings of any normal length, INSTR works almost instantly.

Personal Address Book

For a more practical demonstration of INSTR, delete line 10 from Program 1 and add the lines in Program 4, "Personal Address Book." This is a simple address book program that uses INSTR in two ways: to retrieve any entry in the blink of an eye, and to alphabetically sort the entries when dumping the whole list to a printer. Actually, Personal Address Book is a skeleton program that with some more work could be turned into a full-fledged, general-purpose filer. And thanks to INSTR, it works nearly as fast as programs written entirely in machine language.

When you run Personal Address Book, it automatically adjusts itself to hold the maximum number of address entries practical with the memory available in your computer. In a 48K or 64K machine running DOS 2.5, there's room for more than 24,700 characters of data. In a 16K machine with tape, there's room for more than 5,600 characters. You can check how much room is left at any time by selecting option 2 on the main menu, "Enter a new name."

Other menu options let you retrieve any entry, including multiple entries of people with the same last name; delete any entry; call a disk directory; print out the entire list in alphabetical order; and save/load address files with disk or tape (when you see the prompt DEVICE: FILENAME, respond D:*filename.ext* for disk or C: for tape). Screen prompts make all these options self-explanatory, and the program is error-trapped against common mistakes.

Most programs of this type written in Atari BASIC simulate string arrays by dividing a large string into many substrings of equal length, allocating a substring for each record. This makes it easier to retrieve an individual record, because the position of its substring within the larger string can be readily calculated. Unfortunately, there are two drawbacks with this method: records can't be longer than the substrings, and shorter records waste memory because they're padded out with blanks. But Personal Address Book gives you the freedom to enter as many lines for each address entry as you want. You can even include short notes to yourself as part of the entry, such as SAM'S COUSIN WHO IS A LAWYER.

Since Personal Address Book is a bare-bones demo program, it does have a few limitations. First, the search routines are case-sensitive. If the names are entered in uppercase/lowercase format, such as "Smith, Margaret," a search for all-uppercase "SMITH" results in a NOT FOUND message. Second, if you search for a keyword that isn't the first word in the record—such as "Margaret" in the previous example—the program retrieves only the fraction of the record which starts with that keyword. This means you should type in your entries using the format in which you plan to retrieve them, like this:

Smith, Margaret
604 Geronimo Avenue
Hometown, New York 10000
(212) 555-1212

To retrieve this record, you'd select option 1 and type "Smith" or "Smith, Margaret."

And finally, the print option sorts the names alphabetically by the first character only, so "Smith, Margaret" might not be printed before "Smith, Zelda" or even "Szabo, Martin." This was done to keep the demo program as short as possible.

Programming Notes

To see how easily search routines are written with INSTR, examine lines 360–450 and 600–660. Notice how the START argument is updated after each search to find any following occurrences of the same keyword (if the user so desires).

Since Personal Address Book accepts records of any size, you may be wondering how it figures out the length of each record after it finds the specified keyword. The

answer is a string called EOR$ (End Of Record). EOR$ consists of two carriage returns, and it's tagged onto the end of each record you enter with Personal Address Book. After the INSTR routine finds the keyword, it searches for the next occurrence of EOR$. Then it retrieves the substring between those two points.

BASIC programmers should note the subroutines starting at lines 760, 850, and 1170. By calling the Central Input/Output (CIO) routine built into the Atari operating system, these BASIC subroutines can load and save to tape or disk at machine language speed. If you study the REM statements and descriptive variable names, it isn't too difficult to figure out how these subroutines work. Just be sure to read the machine language data in line 1270 into CIO$ before calling the CIO subroutine in your own programs.

As mentioned above, Personal Address Book is a skeleton program intended mainly for demo purposes. You can add more options of your own to transform it into a mailing-label generator, a general-purpose filer, or even a full-featured database manager. With help from the lightning-fast INSTR routine, the results can be impressive.

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

Program 1: INSTR Routine BASIC Loader

```
NA 10  DIM ML$(260):FOR X=1 T
       O 260:READ A:ML$(X)=CH
       R$(A):NEXT X:ML=ADR(ML
       $)
PC 1275 REM *** INSTR ML DAT
       A ***
OK 1280 DATA 169,0,133,212,1
       33,213
LE 1290 DATA 104,201,5,240,1
       8,141
LC 1300 DATA 0,4,162,0,236,0
FD 1310 DATA 4,208,1,96,104,
       104
OH 1320 DATA 232,169,0,240,2
       43,104
BL 1330 DATA 141,5,4,104,141
       ,4
BJ 1340 DATA 4,104,141,3,4,1
       04
OP 1350 DATA 24,109,4,4,141,
       2
MF 1360 DATA 4,173,3,4,109,5
BO 1370 DATA 4,141,3,4,104,1
       33
ED 1380 DATA 204,104,133,203
       ,104,104
IC 1390 DATA 141,1,4,104,133
       ,206
EP 1400 DATA 104,24,109,4,4,
       133
IK 1410 DATA 205,165,206,109
       ,5,4
CC 1420 DATA 133,206,165,205
       ,56,233
OJ 1430 DATA 1,133,205,165,2
       06,233
HO 1440 DATA 0,133,206,162,0
       ,160
OM 1450 DATA 0,177,205,209,2
       03,240
DE 1460 DATA 37,173,2,4,56,2
       29
CB 1470 DATA 205,141,0,4,173
       ,3
PE 1480 DATA 4,229,206,13,0,
       4
JH 1490 DATA 208,1,96,165,20
       5,24
OF 1500 DATA 105,1,133,205,1
       65,206
IC 1510 DATA 105,0,133,206,1
       69,0
IG 1520 DATA 240,209,232,236
       ,1,4
FM 1530 DATA 208,4,169,0,240
       ,54
EO 1540 DATA 200,177,205,209
       ,203,240
DE 1550 DATA 37,173,2,4,56,2
       29
CB 1560 DATA 205,141,0,4,173
       ,3
PE 1570 DATA 4,229,206,13,0,
       4
JH 1580 DATA 208,1,96,165,20
       5,24
OO 1590 DATA 105,1,133,205,1
       65,206
IC 1600 DATA 105,0,133,206,1
       69,0
IG 1610 DATA 240,155,232,236
       ,1,4
ID 1620 DATA 240,4,169,0,240
       ,202
GA 1630 DATA 173,2,4,56,229,
       205
LN 1640 DATA 141,0,4,173,3,4
FI 1650 DATA 229,206,13,0,4,
       144
ME 1660 DATA 30,240,28,165,2
       05,56
JA 1670 DATA 237,4,4,133,212
       ,165
IP 1680 DATA 206,237,5,4,133
       ,213
LL 1690 DATA 165,212,24,105,
       1,133
OC 1700 DATA 212,165,213,105
       ,0,133
BE 1710 DATA 213,96
```

Program 2: BASIC Search Demo

```
DP 10  DIM XX$(30000),SUB$(10
       )
EB 20  XX$="X":XX$(30000)=XX$
       :XX$(2)=XX$:XX$(30000)
       ="Z"
EM 30  SUB$="Z"
LI 40  FOR X=1 TO LEN(XX$)
BP 50  IF XX$(X,X+LEN(SUB$)-1
       )=SUB$ THEN ? X:? XX$(
       X,X+LEN(SUB$)-1):END
PN 60  NEXT X
LB 70  PRINT "NOT FOUND":END
```

Program 3: INSTR Search Demo

```
EA 20  DIM XX$(30000),SUB$(10
       )
EC 30  XX$="X":XX$(30000)=XX$
       :XX$(2)=XX$:XX$(30000)
       ="Z"
IF 40  SUB$="Z":START=1:? "PR
       ESS ANY KEY TO START S
       EARCH"
HI 50  IF PEEK(764)=255 THEN
       50
PF 60  INSTR=USR(ML,ADR(XX$),
       LEN(XX$),ADR(SUB$),LEN
       (SUB$),START)
NO 70  ? XX$(INSTR,INSTR+LEN(
       SUB$)-1)
BD 80  IF INSTR=0 THEN ? "NOT
        FOUND"
EA 90  END
```

Program 4: Personal Address Book

```
MC 110 FILELEN=FRE(0)-1000:D
       IM FILE$(FILELEN)
GG 120 DIM SUB$(255),ML$(260
       ),FILENAME$(14),EOR$(
       2),ALPHA$(3),PROMPT$(
       28),CIO$(7),DATE$(40)
       ,BOOK$(21)
PJ 130 BOOK$="PERSONAL ADDRE
       SS BOOK"
HM 140 OPEN #1,4,0,"K:":GRAP
       HICS 2:SETCOLOR 2,0,0
       :POKE 752,1
EN 150 POSITION 2,2:? #6;BOO
       K$(1,16):POSITION 8,3
       :? #6;BOOK$(18,21):?
       "Please wait..."
GN 160 EOR$(1)=CHR$(155):EOR
       $(2)=CHR$(155):PROMPT
       $="■PRESS■RETURN■FOR■
       MAIN■MENU■"
DJ 170 FOR X=1 TO 7:READ A:C
       IO$(X)=CHR$(A):NEXT X
EO 180 FOR X=1 TO 260:READ A
       :ML$(X)=CHR$(A):NEXT
       X:ML=ADR(ML$)
EC 190 ? CHR$(125):? PROMPT$
NJ 200 GET #1,A:IF A=155 THE
       N 230
FP 210 GOTO 210
MO 220 REM *** MAIN MENU ***
KJ 230 POKE 82,0:GRAPHICS 0:
       POKE 752,1
ON 240 FOR X=1 TO 9:? CHR$(1
       8);:NEXT X:? BOOK$;:F
       OR X=1 TO 10:? CHR$(1
       8);:NEXT X:POKE 82,10
BD 250 POKE 82,10:? :? "1> R
       etrieve a name":? :?
       "2> Enter a new name"
       :?
MA 260 ? "3> Delete an old n
       ame":? :? "4> Load ad
       dress book":?
AC 270 ? "5> Save address bo
       ok":? :? "6> Print ad
       dress book":? :? "7>
       Disk directory"
AD 280 POKE 82,8:? :? "■PRES
       S■NUMBER■OF■CHOICE■"
FO 290 CLOSE #1:OPEN #1,4,0,
       "K:":GET #1,A:IF A=25
       5 THEN 290
LI 300 ON A-48 GOTO 330,470,
       580,770,860,950,1130
GI 310 GOTO 290
DN 320 REM *** RETRIEVE A NA
```

```
   ME ***
EE 330 POKE 82,2:POKE 752,0:
   ? CHR$(125):POSITION
   5,20:? PROMPT$
GF 340 POSITION 2,10:? "Name
    to retrieve";:INPUT
   SUB$
JF 350 IF LEN(SUB$)=0 THEN 2
   20
JF 360 START=1
AJ 370 INSTR=USR(ML,ADR(FILE
   $),LEN(FILE$),ADR(SUB
   $),LEN(SUB$),START)
KI 380 IF INSTR=0 THEN POKE
   752,1:? CHR$(125):POS
   ITION 5,10:? "NAME NO
   T FOUND":GOTO 420
LN 390 RECORD=INSTR:START=IN
   STR
PL 400 INSTR=USR(ML,ADR(FILE
   $),LEN(FILE$),ADR(EOR
   $),LEN(EOR$),START)
OK 410 ? CHR$(125):? :? FILE
   $(RECORD,INSTR+1):? :
   ? "PRESS SPACE BAR TO
    RETRIEVE":? "NEXT OC
   CURRENCE OF SAME NAME
   "
LI 420 POKE 752,1:? :? PROMP
   T$
NO 430 GET #1,A:IF A=155 THE
   N 230
PL 440 IF A=32 THEN START=ST
   ART+1:GOTO 370
GJ 450 GOTO 430
FK 460 REM *** ENTER A NAME
    ***
NL 470 ? CHR$(125):POKE 752,
   0:POKE 82,2:? :? FILE
   LEN-LEN(FILE$);" CHAR
   ACTERS FREE IN MEMORY
   "
LG 480 ? :? "PRESS RETURN AT
    THIS PROMPT":? "WITH
   OUT INPUT FOR MAIN ME
   NU"
HB 490 ? :? "PRESS RETURN AT
    ANY NEXT PROMPT":? "
   WITHOUT INPUT TO END
   ENTRY"
MF 500 ? :? "NAME";:INPUT SU
   B$
JE 510 IF LEN(SUB$)=0 THEN 2
   30
OC 520 IF LEN(FILE$)=0 THEN
   FILE$(LEN(FILE$)+1)=E
   OR$
BB 530 FILE$(LEN(FILE$)+1)=S
   UB$:FILE$(LEN(FILE$)+
   1)=CHR$(155)
KK 540 ? :? "NEXT LINE OF AD
   DRESS";:INPUT SUB$
JK 550 IF LEN(SUB$)=0 THEN F
   ILE$(LEN(FILE$)+1)=CH
   R$(155):GOTO 230
GM 560 GOTO 530
JB 570 REM *** DELETE A NAME
    ***
FC 580 ? CHR$(125):POKE 752,
   0:POKE 82,2:? :? PROM
   PT$
FM 590 ? :? "NAME TO DELETE"
   ;:INPUT SUB$:IF LEN(S
   UB$)=0 THEN 230
DL 600 START=1:INSTR=USR(ML,
   ADR(FILE$),LEN(FILE$)
   ,ADR(SUB$),LEN(SUB$),
   START)
CK 610 IF INSTR<>0 THEN 650
NJ 620 POKE 752,1:? CHR$(125
   ):POSITION 5,10:? "NA
   ME NOT FOUND":? :? PR
   OMPT$
OA 630 GET #1,A:IF A=155 THE
   N 230
GM 640 GOTO 630
LM 650 RECORD=INSTR:START=IN
   STR
AD 660 INSTR=USR(ML,ADR(FILE
   $),LEN(FILE$),ADR(EOR
   $),LEN(EOR$),START)
MC 670 ? CHR$(125):POKE 752,
   1:? :? FILE$(RECORD,I
   NSTR+1):? :? "PRESS SPACE BAR TO DELETE"
II 680 ? :? "PRESS RETURN FO
   R MAIN MENU"
OG 690 GET #1,A:IF A=155 THE
   N 230
NB 700 IF A=32 THEN 720
HA 710 GOTO 690
IH 720 GAP=INSTR-RECORD+2
OP 730 FILE$(RECORD-2,LEN(FI
   LE$))=FILE$(INSTR,LEN
   (FILE$))
IH 740 FILE$=FILE$(1,LEN(FIL
   E$)-GAP)
GK 750 GOTO 230
LN 760 REM *** LOAD FILE ***
BL 770 ? CHR$(125):POKE 752,
   0:POSITION 5,20:? PRO
   MPT$
OH 780 POSITION 1,10:? "DEVI
   CE:FILENAME TO LOAD";
   :INPUT FILENAME$
PF 790 IF LEN(FILENAME$)=0 T
   HEN 230
CF 800 TRAP 820:CLOSE #2:OPE
   N #2,4,0,FILENAME$:LE
   T READ=1:X=32:MAXLEN=
   FILELEN:SADR=ADR(FILE
   $):GOSUB 1180
OJ 810 CLOSE #2:FILE$(TRUELE
   N)=CHR$(155):TRAP 400
   00:GOTO 230
HE 820 ? CHR$(125):POKE 752,
   1:POSITION 5,10:? "I/
   O ERROR #";PEEK(195):
   POSITION 5,20:? PROMP
   T$:CLOSE #2:TRAP 4000
   0
OC 830 GET #1,A:IF A=155 THE
   N 230
HA 840 GOTO 830
MM 850 REM *** SAVE FILE ***
BL 860 POKE 752,0:? CHR$(125
   ):POSITION 5,20:? PRO
   MPT$
PG 870 POSITION 1,10:? "DEVI
   CE:FILENAME TO SAVE";
   :INPUT FILENAME$
PF 880 IF LEN(FILENAME$)=0 T
   HEN 230
KG 890 TRAP 910:CLOSE #2:OPE
   N #2,8,0,FILENAME$:LE
   T READ=0:X=32:MAXLEN=
   LEN(FILE$):SADR=ADR(F
   ILE$):GOSUB 1180
NB 900 CLOSE #2:TRAP 40000:G
   OTO 230
HE 910 POKE 752,1:? CHR$(125
   ):POSITION 5,10:? "I/
   O ERROR #";PEEK(195):
   POSITION 5,20:? PROMP
   T$:CLOSE #2:TRAP 4000
   0
OC 920 GET #1,A:IF A=155 THE
   N 230
HA 930 GOTO 920
CK 940 REM *** PRINT FILE **
   *
FL 950 POKE 82,2:POKE 752,0:
   ? CHR$(125)
OG 960 ? :? "BE SURE PRINTER
    IS ONLINE"
PE 970 ? :? PROMPT$:? :? "TO
   DAY'S DATE";:INPUT DA
   TE$
ND 980 IF LEN(DATE$)=0 THEN
   230
EP 990 TRAP 1000:CLOSE #2:OP
   EN #2,8,0,"P:":GOTO 1
   010
GA 1000 ? CHR$(125):? :? "I/
   O ERROR #";PEEK(195)
   :TRAP 40000:CLOSE #2
   :GOTO 960
AE 1010 ALPHA$(1)=CHR$(155):
   ALPHA$(2)=CHR$(155):
   START=1:TRAP 40000
II 1020 PRINT #2;BOOK$:PRINT
    #2;"UPDATED ";DATE$
   :X=65
ID 1030 ALPHA$(3)=CHR$(X):IF
    X>90 THEN CLOSE #2:
   GOTO 230
CM 1040 INSTR=USR(ML,ADR(FIL
   E$),LEN(FILE$),ADR(A
   LPHA$),LEN(ALPHA$),S
   TART)
EK 1050 IF INSTR>0 THEN 1070
HN 1060 X=X+1:START=1:GOTO 1
   030
EG 1070 RECORD=INSTR:START=I
   NSTR+2
DA 1080 INSTR=USR(ML,ADR(FIL
   E$),LEN(FILE$),ADR(E
   OR$),LEN(EOR$),START
   )
DP 1085 IF INSTR=0 THEN X=X+
   1:START=1:GOTO 1030
KC 1090 PRINT #2,FILE$(RECOR
   D+1,INSTR-1)
OM 1100 IF INSTR+2<=LEN(FILE
   $) THEN START=INSTR:
   GOTO 1030
HJ 1110 X=X+1:START=1:GOTO 1
   030
IE 1120 REM *** DISK DIRECTO
   RY ***
II 1130 TRAP 1140:? CHR$(125
   ):CLOSE #2:OPEN #2,6
   ,0,"D:*.*":FOR X=1 T
   O 10000:GET #2,A:? C
   HR$(A);:NEXT X
HF 1140 CLOSE #2:? PROMPT$:T
   RAP 40000
HN 1150 GET #1,A:IF A<>155 T
   HEN 1150
JG 1160 GOTO 230
AC 1170 REM *** CIO LOAD/SAV
   E ***
CP 1180 REM CIO LOAD/SAVE re
   quires file#2 opened
   , READ=0 for save, R
   EAD=1 for load
FG 1190 REM file#2,$20
GG 1200 ICCOM=834:ICBADR=836
   :ICBLEN=840:ICSTAT=8
   35
AO 1210 H=INT(SADR/256):L=SA
   DR-H*256:POKE ICBADR
   +X,L:POKE ICBADR+X+1
   ,H
FF 1220 H=INT(MAXLEN/256):L=
   MAXLEN-H*256:POKE IC
   BLEN+X,L:POKE ICBLEN
   +X+1,H
MG 1230 POKE ICCOM+X,11-4*RE
   AD:A=USR(ADR(CIO$),X
   )
LN 1240 TRUELEN=PEEK(ICBLEN+
   X)+256*PEEK(ICBLEN+X
   +1)
KI 1250 RETURN
DH 1260 REM *** CIO ML DATA
    ***
LK 1270 DATA 104,104,104,170
   ,76,86,228
```

IBM Screen Swapping

Paul W. Carlson

If you've ever needed to temporarily store a graphics screen for later recall in a program, or load screens from disk and flash them on the monitor whenever you want, this article shows you how. The programs work on any IBM PC with color/graphics adapter and BASICA or Enhanced Model PCjr with Cartridge BASIC.

You can achieve many interesting effects, including animation, by rapidly switching between several graphics screens stored in memory. Unfortunately, this capability isn't a standard feature on the IBM PC. With help from two very short machine language subroutines, however, you can write programs that swap screens almost instantly. The subroutines copy the video bitmap to or from an array in about five thousandths of a second, much too fast for the eye to see. In fact, this is even faster than the video monitor can display a frame, so the effect is truly instantaneous.

To get started, type in Program 1 below. It creates two files, SCRNARRY.BAS and ARRYSCRN.BAS, which contain the two machine language subroutines. The first routine copies the video bitmap to an array, and the second copies the contents of an array to the video bitmap. The routines achieve their speed by treating the bitmap as a continuous string of 16,192 bytes.

For an example of how to use these routines in your own programs, type in Program 2 and save it on the same disk with SCRNARRY.BAS and ARRYSCRN.BAS. Before running Program 2, make sure the disk is in the active drive; it accesses the two routines as it runs. After typing RUN, don't press any keys until you want to halt the program.

You should see three multicolored spirals drawn on the screen. The first two disappear as soon as they're completed, and the third seems to rotate. The rotation, of course, is an illusion. Here's what happens: In the split-second between the time the first two spirals are completed and then erased, each screen is copied into an array by SCRNARRY.BAS. The third spiral is also copied into an array. Finally, the contents of all three arrays are repeatedly copied to the screen by ARRYSCRN.BAS to get the rotating effect. Actually, the program requires a time-delay loop to keep the screen-flipping from happening *too* fast.

See the figure below for an explanation of Program 2.

Computerized Slide Show

You can load a graphics screen from disk directly into an array the same way Program 2 loads the machine language into arrays. Why would you want to do this? Suppose you had saved graphics screens from three different programs on disk using statements such as this:

```
DEF SEG=&HB800:BSAVE"filename"
,0,16192
```

with filenames of PIC1, PIC2, and PIC3. You could then use Program 3 to display a "slide show" of your creations.

Explanation of Program 2

Line	Description
20,30	Loads the machine language subroutines into the STOA and ATOS arrays.
40–140	Draws and paints three spirals, each one with the colors shifted.
150	GETSCRN is the entry point for the subroutine that copies the screen to an array. *Important:* No new simple variables can be assigned from the point GETSCRN is computed to the point it is used in a CALL statement. Assigning simple variables causes array addresses to move.
160–200	Copies the screen to array SCRN1, SCRN2, or SCRN3 after each spiral is complete.
210	PUTSCRN is the entry point for the subroutine that copies an array to the screen. The same note for line 150 applies here also.
220–250	Repeatedly copies the arrays SCRN1, SCRN2, and SCRN3 to the screen until a key is pressed.

This interesting program displays one screen while loading another. Pressing the space bar (after giving the next screen time to load) displays the next picture. The program could be extended to accommodate any number of screens, even prompting you to change disks if necessary. It needs only one array to store the screens no matter how many you want to display, since it stores only one screen at any moment.

Notice that the statement LA=0 in line 10 of Program 3 prevents the address of the ATOS array from changing after it is assigned a value for PUTSCRN in line 30. (See the note for line 150 in the breakdown of Program 2.)

Programs 4 and 5 show the source code for the SCRNARRY and ARRYSCRN subroutines. They aren't required for use with Programs 1–3; they're listed so machine language programmers can observe the techniques involved. An assembler is required to enter these listings.

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

Program 1: Screen Swapping Routines

```
GH 10 DIM M(7),J(6):DEF SEG
FK 20 FOR N=0 TO 26:READ B
NI 30 POKE VARPTR(M(0))+N,B:NEXT
QQ 40 BSAVE"SCRNARRY",VARPTR(M(0
      )),27
BN 50 FOR N=0 TO 22:READ B
KL 60 POKE VARPTR(J(0))+N,B:NEXT
BI 70 BSAVE"ARRYSCRN",VARPTR(J(0
      )),23:END
KE 80 DATA 6,30,7,30,139,236,184
      ,0
PO 90 DATA 184,142,216,185,160,3
      1,51,246
DH 100 DATA 139,126,8,252,243,16
       5,31,7
KN 110 DATA 202,2,0,6,139,236,18
       4,0
KF 120 DATA 184,142,192,185,160,
       31,51,255
OE 130 DATA 139,118,6,252,243,16
       5,7,202
BP 140 DATA 2,0
```

Program 2: Spiral Demo

```
MK 10 DIM SCRN1(4048),SCRN2(4048
      ),SCRN3(4048),STOA(7),ATOS
      (6)
GJ 20 DEF SEG:BLOAD"SCRNARRY",VA
      RPTR(STOA(0))
NK 30 BLOAD"ARRYSCRN",VARPTR(ATO
      S(0))
HJ 40 KEY OFF:SCREEN 1:COLOR 0,0
FJ 50 FOR C=1 TO 3:W=C:CLS
NM 60 TP=6.283185:F=80/TP:DA=TP/
      9:DB=TP/20:A=0
PM 70 FOR I=1 TO 9:B=0:A=A+DA:PS
      ET(160,100)
GJ 80 FOR J=1 TO 20:B=B+DB:R=F*B
LN 90 X=160+1.2*R*SIN(A+B):Y=100
      +R*COS(A+B)
LD 100 LINE -(X,Y),3:NEXT J,I
JK 110 CIRCLE(160,100),96,3:A=DA
       /2
PM 120 FOR I=1 TO 9:A=A+DA
FL 130 X=160+1.18*R*SIN(A):Y=100
       +.96*R*COS(A)
FA 140 C=C MOD 3+1:PAINT(X,Y),C,
       3:NEXT I
FN 150 GETSCRN=VARPTR(STOA(0))
GD 160 ON W GOTO 170,180,190
LC 170 CALL GETSCRN(SCRN1(0)):GO
       TO 200
NK 180 CALL GETSCRN(SCRN2(0)):GO
       TO 200
EC 190 CALL GETSCRN(SCRN3(0))
JM 200 NEXT C
BJ 210 PUTSCRN=VARPTR(ATOS(0))
NB 220 CALL PUTSCRN(SCRN1(0)):FO
       R J=0 TO 100:NEXT
PJ 230 CALL PUTSCRN(SCRN2(0)):FO
       R J=0 TO 100:NEXT
AB 240 CALL PUTSCRN(SCRN3(0)):FO
       R J=0 TO 100:NEXT
ME 250 IF INKEY$="" THEN 220
MO 260 CLS:SCREEN 0:WIDTH 80:KEY
       ON:END
```

Program 3: Slide Show Demo

```
FE 10 DIM SCRN(4048),ATOS(6):LA=
      0
HP 20 DEF SEG:BLOAD"ARRYSCRN",VA
      RPTR(ATOS(0))
DD 30 PUTSCRN=VARPTR(ATOS(0)):LA
      =VARPTR(SCRN(0))
NA 40 BLOAD"PIC1",LA
FD 50 KEY OFF:CLS:SCREEN 1:COLOR
       0,1
AB 60 CALL PUTSCRN(SCRN(0)):BLOA
      D"PIC2",LA
GB 70 IF INKEY$<>" " THEN 70
CG 80 CALL PUTSCRN(SCRN(0)):BLOA
      D"PIC3",LA
JD 90 IF INKEY$<>" " THEN 90
DO 100 CALL PUTSCRN(SCRN(0))
LI 110 IF INKEY$<>" " THEN 110
LF 120 CLS:SCREEN 0:WIDTH 80:KEY
       ON:END
```

Program 4: SCRNARRY Source Code

Note: This source code is provided for information only. It is not required for Programs 1–3. An assembler is required to enter this listing.

```
;  This subroutine copies 16192 bytes from the video display
;  into a BASIC array.
;
CSEG    SEGMENT
STOA    PROC    FAR
        ASSUME CS:CSEG
        PUSH    ES              ;Save extra segment
        PUSH    DS              ;Set the extra segment
        POP     ES              ;    eqaul to the data segment
        PUSH    DS              ;Save the data segment
        MOV     BP,SP           ;Make BP point to the stack
        MOV     AX,0B800H       ;Set data segment to beginning
        MOV     DS,AX           ;   of video RAM.
        MOV     CX,8096         ;Initialize move counter
        XOR     SI,SI           ;Initialize source index
        MOV     DI,8[BP]        ;Init. dest. index to array offset
        CLD                     ;Set direction flag
REP     MOVSW                   ;Move the display to the array
        POP     DS              ;Restore the data segment
        POP     ES              ;Restore the extra segment
        RET     2               ;Clean up the stack
STOA    ENDP
CSEG    ENDS
        END
```

Program 5: ARRYSCRN Source Code

Note: This source code is provided for information only. It is not required for Programs 1–3. An assembler is required to enter this listing.

```
;  This subroutine copies 16192 bytes from a BASIC array
;  to the video display.
;
CSEG    SEGMENT
ATOS    PROC    FAR
        ASSUME CS:CSEG
        PUSH    ES              ;Save extra segment
        MOV     BP,SP           ;Make BP point to stack
        MOV     AX,0B800H       ;Set extra segment to beginning
        MOV     ES,AX           ;   of video RAM.
        MOV     CX,8096         ;Initialize move counter
        XOR     DI,DI           ;Initialize destination index
        MOV     SI,6[BP]        ;Init. source index to array offset
        CLD                     ;Set direction flag
REP     MOVSW                   ;Move the array to the screen
        POP     ES              ;Restore extra segment
        RET     2               ;Clean up stack
ATOS    ENDP
CSEG    ENDS
        END
```

Speedy Strings For Commodore

Tibor Friedman

Here's a fast machine language routine that lets you load large amounts of data into memory very quickly. You can use it without knowing anything about machine language, and the demonstration programs include two handy disk utilities. A disk drive is required, and a printer is optional.

In Commodore BASIC, the conventional ways to retrieve information from a disk are the GET# or INPUT# commands. Though GET# is the more flexible of the two, INPUT# is much faster, since it pulls in an entire string at once rather than reading one character at a time. Even with INPUT#, however, reading large files from BASIC can be a slow and tedious process.

"Speedy Strings" offers a faster alternative which you may find useful in a variety of applications. Here's the idea: First, you create a string array in memory, making every individual array element the same length. Then you load the array data from disk with a fast machine language (ML) routine, putting it directly into the already-established array elements. Don't worry if that sounds a bit confusing—the examples show you how much time the technique can save. And you don't need to understand machine language to use the routine in your own programs.

Type in and save Programs 1, 2, and 3 on disk before doing anything else. Then load and run Program 1, which demonstrates the speed difference between ordinary string retrieval and the Speedy Strings technique. The program begins by creating a 600-element string array and filling each element with a string that consists of 20 spaces (lines 100–120). Then it POKEs the ML routine into memory (line 130) and creates a disk file of string data (line 140). Then the program calls a subroutine that retrieves the data from disk and stores it in the string array using INPUT# statements within a conventional FOR-NEXT loop. After displaying the time elapsed during that operation, it retrieves the data using the Speedy Strings technique. As you'll see, the second method is considerably faster.

Before you can call the ML routine to load the data from disk, you must have created a string array in memory to receive the data. And the array elements must all be of equal length so the ML routine knows where to put each piece of data. When creating the array, you must make sure that every string begins with a space (character code 32) as shown in line 120 of Program 1. Otherwise the ML routine won't work properly.

Fast Disk Menu

Program 2, "Fast Disk Menu," demonstrates a practical application of this technique. Even if you're not interested in the technique itself, you may find this a valuable addition to your program library. It lets you quickly scan the directory of a disk, sort it alphabetically if you like, and quickly load any program shown on the screen.

When you run Fast Disk Menu, it reads the directory of the current disk and displays a screenful of information in much the same format as if you had entered LOAD "$0",8 followed by LIST. Each program is listed by name, with the familiar PRG, SEQ, or REL type indicator at the right. Non-PRG files are highlighted in a different color. At the left of each filename is a number. If the disk contains more programs than the screen can hold, you can press the space bar to view the rest of the directory.

To load and run a program from the directory, simply press the f1 key and enter the number of the program you want to run. It automatically loads and runs, replacing Fast Disk Menu in memory (note that this works only for conventional BASIC programs that you can start with LOAD and RUN). You can also dump the directory on a printer by pressing f5. Before doing this, you may want to sort the filenames into alphabetical order by pressing f3.

In its present form, Fast Disk Menu POKEs the ML code into memory every time you run it. By making some slight modifications, you can resave the program with the ML routine "pasted onto" the end of the BASIC program itself. The routine beginning at line 620 does most of the work for you. Execute this routine by typing GOTO 630 and pressing RETURN. Replace line 120 with 120 QQ=(PEEK(45)+256*PEEK(46)−73). Then delete every line from 560 to the end of the program and resave it as you would any other BASIC program. When you reload and run the program, it already includes the ML routine.

Fast Disk Catalog

Program 3 uses the same technique to speed up the process of cataloging a number of disk directories. It catalogs and alphabetizes as many as 600 filenames for you and prints the results on a printer, aligning all the information into three neat columns.

When you first run Program 3, it indicates that 600 records are available for storing directory information. To read a disk directory, simply place a disk in the drive and press R. Afterward, the program shows how many records are still available for storage; how many files have been recorded in total; how many files were found on the disk; and the disk ID.

You can continue this process, inserting new disks and pressing R to read their directories, as long as the display shows there is record space available (or until you run out of disks). If you need more than 600 entries, increase the value of MM in line 40. Be sure you have enough space allocated, since attempting to add entries when no more space is available will crash the program.

Once you've read as many directories as you want, press Q to exit this portion of the program and proceed to the next. Just as in Program 2, the filenames are displayed on the screen with non-PRG names highlighted in a different color. If the screen cannot hold all the filenames you've recorded, press the space bar to view the next screenful of names.

At this point you can print out the disk directory. Before doing so, you may want to press the f5 key to alphabetize the master directory. Then press f1 to dump the directory on a printer. The master catalog is printed in three columns, with the first column indented a few spaces so you can insert the printout in a three-ring binder.

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

Program 1: Speedy Strings Demonstration

```
100 MM=600:REM MAX. MEMORY
                              :rem 9
110 DIMF$(MM):PRINT"{CLR} SETT
    ING UP STRINGS IN BASIC"
                              :rem 69
120 FORI=0TOMM:F$(I)=CHR$(32)+
    "{19 SPACES}":NEXT:REM* 19
     SPACES=LEN 20     :rem 118
130 GOSUB600:REM LOAD ML
                              :rem 129
140 GOSUB300:REM CREATE A FILE
     OF STRINGS ON DISK:rem 98
150 GOSUB400:REM CONVENTIONAL
    {SPACE}RETRIEVAL AND TIME
                              :rem 8
160 GOSUB200:REM SPEEDY STRING
     METHOD                   :rem 41
170 END                       :rem 111
200 PRINT"{2 DOWN} SPEEDY RETR
    IEVAL"                    :rem 253
210 TI$="000000":OPEN15,8,15:O
    PEN1,8,0,"0:TEST 0,S,R":SY
    S(832):CLOSE1:CLOSE15
                              :rem 127
220 C=PEEK(980)+256*PEEK(981)
                              :rem 53
230 T$=TI$                    :rem 11
240 FORI=0TOC-1:PRINTI;F$(I):N
    EXT                       :rem 63
250 PRINT,MID$(T$,3,2)":"RIGHT
    $(T$,2)                   :rem 27
260 RETURN                    :rem 120
290 REM CREATE FILE           :rem 83
300 PRINT"{2 DOWN}CREATING A T
    EST FILE AND SAVING TO DIS
    K"                        :rem 221
310 FORI=0TOMM:F$(I)="*1234567
    890123456789":NEXT:REM * 2
    0 CHARACTERS              :rem 68
320 OPEN2,8,2,"@0:TEST 0,S,W"
                              :rem 71
330 FORI=0TOMM:PRINT#2,F$(I):N
    EXT:CLOSE2:RETURN         :rem 49
400 PRINT"{2 DOWN} CONVENTIONA
    L RETRIEVAL"              :rem 197
410 TI$="000000":OPEN15,8,15:O
    PEN2,8,0,"0:TEST 0,S,R":C=
    0                         :rem 78
420 IFST=64THEN440            :rem 58
430 INPUT#2,F$(C):C=C+1:GOTO42
    0                         :rem 8
440 CLOSE2:CLOSE15:T$=TI$
                              :rem 6
450 FORI=0TOC-1:PRINTI;F$(I):N
    EXT                       :rem 66
460 PRINT,MID$(T$,3,2)":"RIGHT
    $(T$,2)                   :rem 30
480 RETURN                    :rem 124
590 REM *M/L STRING UPLOAD*
    {2 SPACES}RELOCATABLE/*SET
     # OF CHARACTERS IN STRING
    *                         :rem 205
600 PRINT"{2 DOWN} LOADING ML"
                              :rem 32
610 AD=832:FORI=0TO79:READD:PO
    KEI+AD,D:NEXT:RETURN
                              :rem 223
620 DATA169,255,141,212,3,141,
    213,3,165,55              :rem 61
630 DATA133,252,165,56,133,253
    ,162,1,32,198,255         :rem 64
640 DATA174,212,3,232,142,212,
    3,208,3,238,213,3,216,56,1
    65,252,233                :rem 210
650 DATA21:REM STRING LEN+1
                              :rem 72
660 DATA133,252,176,5,166,253,
    202,134,253,32,207,255
                              :rem 52
670 DATA164,144,208,18,201,13,
    240,245,160,0,145,252,200,
    192                       :rem 122
680 DATA20:REM STRING LEN
                              :rem 238
690 DATA240,209,32,207,255,208
    ,244,32,204,255,96,0,0,0
                              :rem 138
```

Program 2: Fast Disk Menu

```
100 POKE53280,6:POKE53281,6
                              :rem 242
110 PRINTCHR$(14)CHR$(8):PRINT
    "{CLR}{BLK}{10 DOWN}{YEL}"
    SPC(12)"FAST DISK MENU
                              :rem 160
120 REM{2 SPACES}QQ=PEEK(45)+2
    56*PEEK(46)-73            :rem 158
130 GOSUB570:QQ=AD            :rem 78
140 DIMF$(120):FORI=0TO120
                              :rem 202
150 F$(I)=CHR$(32)+"
    {20 SPACES}":NEXT:Y$=CHR$(
    34):Z$=CHR$(190)          :rem 4
160 OPEN1,8,0,"$":SYSQQ:CLOSE1
                              :rem 10
170 C=PEEK(0):A=1             :rem 167
180 IFA=>CTHENA=1             :rem 5
190 PRINT"{CLR}{DOWN}
    {7 SPACES}{RVS}{8}"Y$LEFT$
    (F$(0),16)Y$"{3 SPACES}"MI
    D$(F$(0),18,2)            :rem 69
200 PRINT:FORI=ATOA+8:IFI=>CTH
    ENPRINT:GOTO270           :rem 230
210 IFLEFT$(F$(I),1)=Z$THENE=2
    :GOTO230                  :rem 201
220 E=1:IFMID$(F$(I),18,3)<>"P
    RG"THENE=2:F$(I)=Z$+LEFT$(
    F$(I),20)                 :rem 55
230 IFE=2THENPOKE646,3:GOTO250
                              :rem 63
240 POKE646,7                 :rem 200
250 PRINTTAB(5)I;Y$MID$(F$(I),
    E,16)Y$;                  :rem 145
260 PRINT"{3 SPACES}"MID$(F$(I
    ),17+E,3)                 :rem 31
270 PRINT:NEXT                :rem 159
280 PRINT"{DOWN}{8}{5 SPACES}F
    1-LOAD{3 SPACES}F3-SORT
    {3 SPACES}F5-PRINT":POKE19
    8,0:WAIT198,1             :rem 211
290 IFPEEK(197)=4THEN340
                              :rem 120
300 IFPEEK(197)=5THEN390
                              :rem 118
310 IFPEEK(197)=6THEN460
                              :rem 118
320 IFPEEK(197)=60THENA=A+9:GO
    TO180                     :rem 60
330 PRINT"{3 UP}":GOTO280
                              :rem 39
340 INPUT"{DOWN}{5 SPACES}PROG
    RAM #";N$:N=VAL(N$):IFN<1O
    RN>CTHEN190               :rem 12
350 F$=LEFT$(F$(N),16):rem 123
360 IFRIGHT$(F$,1)=" "THENF$=L
    EFT$(F$,LEN(F$)-1):GOTO360
                              :rem 115
370 PRINT"{CLR}LOAD"Y$F$Y$",8
                              :rem 8
380 POKE631,19:POKE632,13:POKE
    633,82:POKE634,117:POKE635
    ,13:POKE198,5:END         :rem 8
390 PRINT"{CLR}{5 DOWN}
    {8 SPACES}S{2 SPACES}O
    {2 SPACES}R{2 SPACES}T
    {2 SPACES}I{2 SPACES}N
    {2 SPACES}G               :rem 217
400 FORI=1TOC-2:IFF$(I)<F$(I+1
    )THEN450                  :rem 168
410 Q$=F$(I+1)                :rem 167
420 FORJ=ITO1STEP-1:IFF$(J)<Q$
```

```
    THENF$(J+1)=Q$:GOTO450
                          :rem 136
430 F$(J+1)=F$(J):NEXT:rem 179
440 F$(1)=Q$          :rem 54
450 NEXTI:A=1:GOTO190  :rem 23
460 PRINT"{CLR} {DOWN}IS PRINT
    ER ON?":POKE198,0:WAIT198,
    1:GETMT$:IFMT$<>"Y"THEN190
                          :rem 77
470 OPEN4,4,7          :rem 196
480 PRINT#4,Y$LEFT$(F$(0),16)Y
    $"{3 SPACES}"MID$(F$(0),18
    ,2):PRINT#4,CHR$(20)CHR$(1
    5);                   :rem 38
490 U=C-1:V=INT(U/3)+1:rem 236
500 FORI=1TOV:FORJ=0TO3:Q=I+J*
    V:IFQ>UTHEN530        :rem 84
510 E=1:IFLEFT$(F$(Q),1)=Z$THE
    NE=2                  :rem 185
520 PRINT#4,Y$MID$(F$(Q),E,16)
    Y$;:IFE=2THENPRINT#4,Z$;
                          :rem 176
530 PRINT#4,"{5 SPACES}";:NEXT
    :PRINT#4              :rem 190
540 NEXT:CLOSE4:A=1:GOTO190
                          :rem 178
550 END                   :rem 113
560 REM LOAD ML           :rem 56
570 AD=830:FORI=0TO72:READD:PO
    KEI+AD,D:NEXT:RETURN
                          :rem 219
580 DATA169,255,133,0,165,55,1
    33,71,165,56,133,72,162,1,
    32,198,255,166,0,232
                          :rem 219
590 DATA134,0,216,56,165,71,23
    3,22,133,71,176,5,166,72,2
    02,134,72,32,207,255
                          :rem 207
600 DATA164,144,208,22,201,34,
    208,245,160,0,32,207,255,2
    01,34,240,249,145,71
                          :rem 187
610 DATA200,192,21,208,242,240
    ,207,32,204,255,96,0,0,0
                          :rem 117
620 REM *TO TACK M/L TO END OF
     THE PRGR.{8 SPACES}*
    {2 SPACES}FIRST: RUN 630
    {2 SPACES}*           :rem 49
630 POKE45,((PEEK(45)+73)AND25
    5):POKE46,PEEK(46)-(PEEK(4
    5)<72)                :rem 195
640 POKE47,PEEK(45):POKE48,PEE
    K(46):POKE49,PEEK(45):POKE
    50,PEEK(46)           :rem 219
650 AD=PEEK(45)+256*PEEK(46)-7
     3:RESTORE            :rem 3
660 FORI=0TO72:READD:POKEI+AD,
    D:NEXT                :rem 42
```

Program 3: Fast Disk Catalog

```
10 POKE56,PEEK(56)-1:CLR:POKE5
   3281,6:POKE53280,6  :rem 28
20 PRINTCHR$(14)CHR$(8):PRINT"
   {CLR}{8 DOWN}","{CYN} FAST
   {SPACE}DISK CATALOG":rem 57
25 PRINT,"{8 DOWN}"TAB(14)"PLE
   ASE WAIT":GOSUB1010:rem 168
30 A$="{DOWN}{CYN}{2 SPACES}
   {RVS}R{OFF}EAD A DISK OR
   {RVS}Q{OFF}UIT":X$="{UP}
   {24 SPACES}{2 UP}" :rem 151
40 MM=600:REM MAX. MEM.-UP TO
   {SPACE}16000          :rem 124
50 DIMF$(MM):A=0:EA=PEEK(45)+2
   56*PEEK(46):AD=EA-373
                         :rem 118
60 FORI=0TOMM:F$(I)=CHR$(32)+"
   {20 SPACES}":NEXT     :rem 90
70 PRINT,"{CLR}",,"{RVS}ROOM F
   OR"MM"{LEFT} RECORDS
                         :rem 180
75 PRINT"{YEL}{2 @}ID{5 @}#COU
   NT{5 @}TOTAL{3 @}AVAIL.SPAC
   E{@}                  :rem 61
80 PRINTA$:POKE198,0:WAIT198,1
   :IFPEEK(197)=62THEN130
                         :rem 119
82 IFPEEK(197)<>17THENPRINTX$:
   GOTO80                :rem 5
85 PRINT"{UP}{5 SPACES}READING
   {9 SPACES}"           :rem 73
90 OPEN15,8,15,"I0"      :rem 190
100 OPEN1,8,0,"$0":SYS(AD):CLO
    SE1:CLOSE15          :rem 126
110 C=PEEK(980)+256*PEEK(981)
                         :rem 51
120 PRINTX$:PRINT"{CYN}
    {2 SPACES}"RIGHT$(F$(C-1),
    2),C-B,C,MM-C:B=C:AD=EA-35
    7:GOTO80             :rem 155
130 IFA=>CTHENA=.:IFC=ATHENPOK
    E198,0:END           :rem 244
140 PRINT"{CLR}":FORI=ATOA+16:
    IFI>=CTHENPRINT:GOTO149
                         :rem 244
143 PRINTTAB(5)"{CYN}";:IFMID$
    (F$(I),18,1)<>"P"THENPRINT
    "{YEL}";             :rem 149
145 PRINTCHR$(34)LEFT$(F$(I),1
    6)CHR$(34);          :rem 11
147 IFMID$(F$(I),18,1)<>"P"THE
    NPRINTTAB(25)MID$(F$(I),18
    ,2);                 :rem 74
148 PRINTTAB(31)RIGHT$(F$(I),2
    )                    :rem 11
149 NEXT                 :rem 221
150 PRINT,"{3 DOWN}{RVS}F5
    {OFF}-SORT{3 SPACES}{RVS}F
    1{OFF}-PRINT":POKE198,0:WA
    IT198,1              :rem 132
160 IFPEEK(197)=6ANDFL=0THEN19
    0                    :rem 75
170 IFPEEK(197)=4THEN260
                         :rem 118
180 A=A+17:GOTO130       :rem 242
190 FL=1:PRINT"{CLR}"SPC(250)"
    S{2 SPACES}O{2 SPACES}R
    {2 SPACES}T{2 SPACES}I
    {2 SPACES}N{2 SPACES}G
                         :rem 206
200 POKE987,70:POKE988,0:SYS(E
    A-261)               :rem 64
220 A=0:GOTO140          :rem 74
260 PRINT"{CLR} {CYN}{RVS}
    {2 SPACES}IS THE PRINTER O
    N?":POKE198,0:WAIT198,1
                         :rem 127
270 OPEN4,4,7            :rem 194
280 PRINT#4,"{5 SPACES}C A T A
     L O G":PRINT#4,CHR$(20)CH
    R$(15);              :rem 90
290 U=C:V=INT(U/3)+1     :rem 140
300 FORI=1TOV:FORJ=0TO3:Q=I+J*
    V                    :rem 222
305 IFQ>UORLEFT$(F$(Q),1)=" "T
    HEN320               :rem 250
310 PRINT#4,CHR$(34)LEFT$(F$(Q
    ),16)CHR$(34)" "RIGHT$(F$(
    Q),2)"{3 SPACES}";:rem 117
320 NEXT:PRINT#4         :rem 242
330 NEXT:PRINT#4:CLOSE4:A=.:GO
    TO140                :rem 197
1000 REM ML LOADER       :rem 245
1010 AA=832:FORI=0TO110:READD:
     POKEAA+I,D:NEXT     :rem 12
1020 AD=PEEK(55)+256*PEEK(56)
                         :rem 56
1030 FORI=0TO255:READD:POKEAD+
     I,D:NEXT:RETURN     :rem 159
1040 REM{2 SPACES}*READ DISK*
                         :rem 68
1050 DATA169,0,141,212,3,141,2
     13,3,165,55,133,252,165,5
     6,133,253,234,162,1,32
                         :rem 231
1060 DATA198,255,160,26,32,207
     ,255,136,208,250,32,207,2
     55,133,254,32,207,255
                         :rem 213
1070 DATA133,255,32,207,255,16
     4,144,208,60,201,34,208,2
     45,174,212,3,232,142,212
                         :rem 81
1080 DATA3,208,3,238,213,3,216
     ,56,165,252,233,22,133,25
     2,176,5,166,253,202,134
                         :rem 40
1090 DATA253,160,0,32,207,255,
     201,34,240,249,145,252,20
     0,192,19,208,242 :rem 203
1100 DATA165,254,145,252,200,1
     65,255,145,252,192,0,208,
     189,32,204,255,96 :rem 17
1110 REM *SORT*          :rem 67
1120 DATA173,219,3,41,127,141,
     219,3,173,220     :rem 103
1130 DATA3,9,128,141,220,3,165
     ,47,133,254,165   :rem 208
1140 DATA48,133,255,160,0,177,
     254,205,219,3,208,8,200,1
     77,254,205,220,3,240,30
                         :rem 31
1150 DATA160,2,177,254,141,216
     ,3,200,177,254,141,217,3,
     24,165,254,109,216,3,133
                         :rem 83
1160 DATA254,165,255,109,217,3
     ,133,255,144,209,24,165,2
     54,105,7,141,216,3,165
                         :rem 3
1170 DATA255,105,0,141,217,3,5
     6,173,212,3,233,1,141,212
     ,3,173,213,3,233,0,141
                         :rem 207
1180 DATA213,3,174,213,3,208,7
     ,173,212,3,201,0,240,23,1
     73,216,3,133,254,173,217
                         :rem 66
1190 DATA3,133,255,169,0,141,2
     18,3,141,214,3,141,215,3,
     240,19,32,95,229,96,208
                         :rem 41
1200 DATA198,24,165,254,105,3,
     133,254,165,255,105,0,133
     ,255,160,1,177,254,133
                         :rem 249
1210 DATA218,200,177,254,133,2
     19,200,200,177,254,133,22
     8,200,177,254,133,229
                         :rem 201
1220 DATA160,0,177,218,209,228
     ,240,4,144,36,176,7,200,1
     92,16,240,29,144,239,160
                         :rem 96
1230 DATA1,165,228,145,254,200
     ,165,229,145,254,200,200,
     165,218,145,254,200,165
                         :rem 40
1240 DATA219,145,254,169,1,141
     ,218,3,238,214,3,208,3,23
     8,215,3,173,214,3,205
                         :rem 195
1250 DATA212,3,208,159,173,215
     ,3,205,213,3,208,151,174,
     218,3,240,140,208,142
                         :rem 184
                                ©
```

Introduction To AmigaDOS Part 2

Charles Brannon, Program Editor

Last month, Part 1 covered the conventions of AmigaDOS and explained its most useful commands. This month's article wraps up the reference guide to AmigaDOS's interactive commands. A future article will cover the use of batch files and batch programming in AmigaDOS.

After working with the powerful AmigaDOS commands covered last month, you may decide that you prefer working with the AmigaDOS Command Line Interface (CLI) instead of the Workbench. If so, you may want to do away with the Workbench altogether. It wastes time and memory to load the Workbench every session merely to open a CLI if all you want to use is AmigaDOS anyway.

Fortunately, it's fairly simple to create an AmigaDOS-only disk. This disk can be used whenever the system asks for a Workbench disk. You probably won't want to modify your original Workbench disk, however; it's better to modify a copy of it and set aside the original for safekeeping. You can make several copies of your AmigaDOS disk for future use, if you want. Just follow these steps:

A Custom DOS Disk

1. Open the System drawer on the Workbench disk. If you don't see the CLI icon—a small cube labeled with a 1> symbol—run Preferences. (Otherwise continue to step 2.) One of the settings on the first Preferences screen is labeled CLI [ON] [OFF]. Click it ON, then click on the Save box to save the change to disk. Return to the Workbench and reopen the System folder. You should now see the CLI icon.

2. Double-click on the CLI icon. A window titled "New CLI Window" appears. Click inside the window to make the CLI active.

3. At the 1> prompt, type ED S/Startup-Sequence and press RETURN. This loads a program called ED, a full-screen editor, and loads the file Startup-Sequence from the S subdirectory. Startup-Sequence is the batch file that makes AmigaDOS automatically start the Workbench when you boot the Workbench disk. After ED starts, you should see something like this on the screen:

```
ECHO "WorkBench Disk. Version 1.00"
ECHO " "
ECHO "Use Preferences tool to set date"
ECHO " "
LoadWb
endcli > nil:
```

These are the batch file commands that AmigaDOS executes each time you boot up the Workbench disk. The ECHO commands are similar to PRINT statements in BASIC; they merely display messages on the screen. It's the last two commands in this file that we're interested in changing.

4. Using the cursor keys, move the cursor to the line with the LoadWb command and press CTRL-B twice to erase the last two lines. The batch file should now consist of the four ECHO commands only. If you wish, you can change the text in the ECHO commands to give your boot disk that "personal touch."

5. Press the ESC key. An asterisk prompt (*) appears at the bottom of the screen. Type X at this prompt and press RETURN. This exits the ED program and saves the new Startup-Sequence file to disk. If you've made a mistake and would like to start over, press ESC-Q to quit the editor without changing the file.

6. After the disk busy light goes off, simultaneously press CTRL and both Amiga keys on each side of the space bar to reboot the system. This time, and from now on whenever you boot with this disk, AmigaDOS ends up in memory instead of the Workbench.

The Workbench Option

To conserve space on your new AmigaDOS disk, you may want to erase some files used by the Workbench, such as the LOADWB command in the C subdirectory, the Notepad, the clock, and all .INFO files. However, it's convenient to have the Workbench available when you need it. You could use the editor to create another batch file that includes LOADWB and ENDCLI > NIL:. You would then type EXECUTE WB at a CLI prompt to bring up the Workbench (assuming you named the batch file WB by typing ED WB to create the batch file). ED is useful for creating all kinds of simple batch files, in fact. We'll examine the editor in more detail in a future article on batch file programming.

Last month's article presented a tutorial on AmigaDOS along with a reference of the most often-used commands. Following is a reference to additional commands that, although useful, are not likely to be used casually. This list excludes commands such as ECHO that are

really useful only in batch files.

When experimenting with AmigaDOS commands, it's safest to use a copy of your DOS disk in case you accidentally erase a file or even the entire disk.

Advanced AmigaDOS Commands

< and > (Input/output redirection.) These symbols redirect the normal input/output flow of a command. For example, a program that normally accepts input from the keyboard and prints its output on the screen could be coerced into accepting input from a file or to send its output to the printer. The < and > symbols are used to point in the direction that I/O should flow; the less-than sign (<) redirects input, and the greater-than sign (>) redirects output. When using < to redirect input, you may need to use a question mark for the parameter that the redirection file is replacing.

Examples:

```
DIR > DIRFILE
```

This redirection of the DIR command sends the disk directory to the file DIRFILE instead of to the screen. To confirm this, you can enter TYPE DIRFILE to display the contents of DIRFILE.

```
STACK < BASIC.STACK ?
```

The stack command normally accepts a command line parameter. Here, a file (BASIC.STACK) containing the number 8000 can be substituted. In order for the file to replace the command line parameter, you must use a question mark to hold that parameter's position.

FILENOTE This command attaches a comment to a file. Although AmigaDOS's 30-character filenames let you be quite descriptive, an optional FILENOTE lets you attach an additional 80-character comment to a file. This comment is displayed beneath the filename when you use the LIST (not DIR) command. Follow FILENOTE with the name of the file you're describing, then the comment. You must enclose the comment in quotes if it includes spaces. The FILENOTE command also lets you include two optional keywords, FILE and COMMENT, presumably for the sake of readability.

Files have no comment by default. The comment is retained if the file is changed or overwritten. However, if you copy a file, its filenote does not get copied with it.

Examples:

```
FILENOTE waver.bas "Program lets you
  create sound waves."
```

After you attach this comment to the file waver.bas, LIST waver.bas yields this result:

```
waver.bas 2272 rwed 11-Oct-85 10:09:53
: Program lets you create sound waves
```

Second example:

```
FILENOTE FILE waver.bas COMMENT
  "Program lets you create sound
waves."
```

This is identical to the first example, except for the optional keywords FILE and COMMENT.

INFO This command shows a disk report. INFO displays the size of each mounted drive (normally 880K, except for the RAM disk), the number of sectors used, number of sectors free, percentage of capacity used, number of disk errors that have occurred, the read/write status, and the disk's name. INFO also separately displays the names of the currently inserted disks. INFO has no additional parameters. Use LIST to display information about a particular file or directory.

INSTALL This command makes a disk bootable. In other words, an INSTALLed disk can be inserted at the Workbench prompt to bring up the system. Just follow INSTALL with the optional keyword DRIVE and the drive number. If you want to be able to execute AmigaDOS commands after booting, you must copy the C subdirectory from your master disk onto the copy. (All AmigaDOS commands are extrinsic and contained in the C subdirectory.)

Example:

```
INSTALL DRIVE DF1:
```

This makes the disk currently mounted in the external drive bootable.

JOIN This command combines two or more files. Follow JOIN with up to ten filenames separated by spaces. The destination file, holding the conglomerate, is specified with the keyword AS. The original files are unchanged.

Example:

```
JOIN Checks/Oct Checks/Nov Checks/
  Dec AS "Checks/4th Quarter"
```

This combines the files Oct, Nov, and Dec from the subdirectory Checks into a single file called "4th Quarter" to be created in the Checks subdirectory. The destination filename is enclosed in quotes because it contains a space character.

PROMPT Defines a new CLI prompt. Follow PROMPT with a message, enclosing it in quotes if the message contains any spaces. The message is a replacement for the normal 1> or 2> prompt of AmigaDOS. You can imbed the characters %N to display the current task number.

Examples:

```
PROMPT "%N> "
```

Displays the default prompt.

```
PROMPT "Ready, Master:"
```

Displays Ready, Master: as the new AmigaDOS prompt.

SEARCH Finds text within files. This command searches for the target string through any directories you specify. Follow SEARCH with the optional keyword FROM, the pathname of the directories to be searched, the optional keyword SEARCH followed by the search string, and the optional keyword ALL, which forces SEARCH to look through all subdirectories contained in the specified directory. When SEARCH finds the target string, it displays the line containing the string as well as the line number of the line containing the string. If you're searching through a directory, SEARCH also displays the filename of each file it's searching through.

SEARCH is not case-sensitive; it matches regardless of upper- or lowercase. You can cancel the command with CTRL-C. To force SEARCH to abandon the current file and begin searching the next, press CTRL-D. During a search, you may see the message "Line xx truncated." This isn't anything to worry about; it just indicates that the line was too long to be searched, so if your search string was contained somewhere near the end of a too-long line, the search program could not find it.

Examples:

SEARCH FROM DF0: SEARCH LoadWb ALL

This looks for the phrase "LoadWb". The entire contents of the internal drive are searched, including all subdirectories, so this command takes a long time to finish.

SEARCH Progs/Tempfile LIBRARY

This looks for the word LIBRARY in the file Tempfile within the subdirectory Progs.

SORT This command alphabetically sorts a file you specify. Each record in the file to be sorted must end with a carriage return. Use SORT followed by the optional keyword FROM, the file to be sorted, the optional keyword TO, and the name of the file where the sorted output should be stored. SORT collates based on the entire line unless you include the keyword COLSTART and a column number. The sort comparison then starts by comparing two lines from that column to the end of the line. If that partial comparison succeeds, the first portion of the line is compared. This lets you specify two levels of sorting (see example).

Unless the file to be sorted is less than about 200 lines, increase the stack size with STACK to prevent a crash (see below). It's better to use too much stack space than too little.

Example:

If you have a list of first and last names, with the first name and initial in columns 1-19, and the last name always starting in column 20, you could use:

SORT FROM Route TO Sorted.Route COLSTART 20

The files are sorted by last name, and each group of identical last names is subsorted by first name.

STACK Sets the stack size. Follow STACK with the new stack size in bytes. The normal stack size is 4,000, sufficient for most commands. When using SORT, MetaComCo ABasiC, programs with lots of nested subroutines, or programs using flood-fill, you may need to increase the stack size to prevent a crash. A value from 8,000 to 10,000 is usually generous enough for these cases.

WAIT This makes AmigaDOS pause and do nothing for a span of time. Although this might seem like a dumb command, WAIT has certain advantages over walking away from the computer or simply turning the machine off. Only the current CLI is frozen; multitasked processes continue. WAIT by itself pauses for one second; you can follow WAIT with a number of seconds, followed by either SEC or SECS, and a number of minutes, followed by either MIN or MINS. You can optionally include the keyword UNTIL followed by a time of day, specified as HH:MM (as measured by the Amiga's internal clock, so make sure it's set correctly). WAIT is useful within batch files to allow time for a message to be read, or as a background task to wait until a particular time before executing another command.

Examples:

WAIT 10 MINS 20 SECS

Waits for 10 minutes, 20 seconds.

WAIT UNTIL 17:00

Waits until the current time is 5 p.m.

```
RUN WAIT 10 SECS +
DIR +
ECHO "All done."
```

Waits for ten seconds, calls a directory as a second CLI task, then prints the message "All done."

WHY This interesting command calls up an additional explanation of what caused the most recent error. When an AmigaDOS command fails, you'll usually get a terse error message. If you want a more detailed, technical description, ask WHY. However, many times WHY isn't any more helpful—it just explains in more detail why a command failed.

Example:

WAIT 10 SECONDS

AmigaDOS responds with the error message "Bad Args" because the correct notation is WAIT 10 SECS, not WAIT 10 SECONDS. If you type WHY, you get this answer:

Last command failed because argument line invalid or too long.

Although more descriptive, it still doesn't explain that SECS should be SECONDS—but it does point you in the right direction. ©

MessageMaker 64

Erik Larsen

Create attractive, attention-grabbing displays for the Commodore 64 (or Commodore 128 in 64 mode) by choosing from eight different sets of oversize letters. The program even works with custom character sets and lets you dump the screens to a Commodore printer. It's easy to use and adds impact to virtually any BASIC application.

Have you ever wished that your Commodore 64 could display bigger screen characters? Though obviously handy for children's educational programs, jumbo characters are useful for many other purposes as well. Nearly every program starts with a title of some sort: Large letters can emphasize it. For anyone who's visually impaired, oversize characters are an invaluable aid to using and understanding computers. And if you have something to sell or trade, what better way to gain attention than by printing your message in giant script?

"MessageMaker 64" answers all of these needs by offering a set of eight different oversize character fonts, ranging from characters four times the normal size to characters that fill the entire screen. All the fonts can be used at any time, and the entire Commodore character set is available for enlargement. That includes all of the normal uppercase/graphics and lowercase/uppercase characters—alphabetic letters, numbers, punctuation, and graphics symbols—in reverse video as well as the normal form.

Using MessageMaker

The first thing to do is type in and save a copy of MessageMaker 64. Before you run the program, disable any programming aids or utilities that use the function keys (f1–f8). Since MessageMaker uses the function keys, this would only lead to conflicts.

As soon as you run MessageMaker, it blacks out the screen and waits for you to press a key. Though you don't see the familiar blinking cursor, the keyboard works as usual in most other respects. If you type A-B-C, it prints ABC on the screen. To print the heart-shaped graphics character, press SHIFT-S. You can activate reverse video with CTRL-9, turn the characters red by pressing CTRL-2, and so on. To select a new font, press one of the eight function keys as shown in the accompanying table.

The eight available fonts are described in terms of width and height. Thus, 4 × 4 characters are four times wider and four times higher than normal; 4 × 8 characters are four times wider and eight times higher. You can mix different fonts freely on the same screen. For instance, you might want to print 8 × 16 characters at the top of the screen, and 4 × 4 characters further down. Of course, the bigger the font, the fewer letters you'll be able to fit on the screen before it begins to scroll. Press CTRL-L to switch to lowercase/uppercase mode, or CTRL-U to switch to uppercase/graphics mode.

MessageMaker Function Keys

Font	Key
4 × 4	f1
4 × 8	f2 (SHIFT-f1)
4 × 16	f3
4 × 24	f4 (SHIFT-f3)
8 × 8	f5
8 × 16	f6 (SHIFT-f5)
8 × 24	f7
32 × 24	f8 (SHIFT-f7)

Note that the cursor keys move one normal character space at a time (not the width of an oversize character). If you accidentally print the wrong character, you must press CRSR-LEFT to back up to the beginning of the character, then replace it with a new one. You can create some interesting effects by printing a character, moving the cursor back near the same position, and printing the same character again.

Signs And Banners

MessageMaker lets you dump a screen to any printer that can handle Commodore graphics characters (specifically, the reverse video space character). This lets you record your screens for posterity, make signs and banners, and so on. Press CTRL-P to print the screen in normal width, or CTRL-X to print

in double-width characters. Since double width prints all the way to the margins, you may prefer this mode for signs. However, note that the symmetrical fonts (4 × 4 and 8 × 8) look squashed when printed in double width;use the taller fonts (4 × 8 and 8 × 16) to alleviate this problem.

The program works as published with the odd-numbered Commodore printers—the 1525, 801, and 803. To use the program with the even-numbered Commodore printers—the 1526 and 802—add the following new line 5:

```
5 OPEN 6,4,6:PRINT#6,CHR$(22):
  CLOSE 6
```

You also need to change the SI$=CHR$(15) in line 890 to SI$="" and the GR$=CHR$(8) in line 900 to GR$="" (in both cases, type the null string—nothing between the quotes).

MessageMaker also works with custom character sets. Of course, this assumes you have already designed the characters and stored their definitions in an appropriate memory area. Only one change is needed: Replace the value 53248 in line 30 with the memory location where your character definitions start.

Integrating MessageMaker Screens

The simplest way to add a MessageMaker screen to an existing program is with "Commodore 64 AutoPRINT," found on page 80 of the July 1985 issue of COMPUTE!. Load and run AutoPRINT, then answer the question about line increments as described in that article. Do *not* type SYS 51000 at this point. Instead, load and run MessageMaker, then create the screen you want. You can use the RETURN key as usual, since AutoPRINT is not yet active. However, you must leave enough blank screen space to enter a few direct-mode commands.

When your title screen is complete, press RUN/STOP to break out of MessageMaker, then erase the BREAK IN (line number) and READY messages. Now load the program to which you want to add the screen, and erase the SEARCHING and LOADING messages. At this point the screen should contain nothing but the screen you designed. Type SYS 51000 and press RETURN, then erase that message from the screen. Now press RETURN anywhere on the screen. AutoPRINT adds the screen to your program as a series of PRINT statements. At this time you can resave the program, renumber it, or modify it in any other way.

MessageMaker 64

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

```
10 GOSUB610              :rem 120
20 PRINT"{CLR}{DOWN}{BLU}";:PO
   KE53281,0:POKE53280,0
                         :rem 192
30 SE(1)=53248:SE(2)=SE(1)+102
   4:SE(3)=SE(1)+2048:SE(4)=SE
   (2)+2048              :rem 172
40 SO=SE(1):SR=SE(2):IS=SO
                         :rem 73
50 GETD$:IFD$=""THEN50:rem 243
60 AA=256:GOSUB130       :rem 16
70 IFAA=256THEN50        :rem 230
80 IFS(4)=1THENFORJJ=0TOS(1)ST
   EPS(2):FORII=1TOS(5):GOSUB2
   70:GOSUB320:NEXTII,JJ
                         :rem 232
90 IFS(4)=2THENFORJJ=0TO7:GOSU
   B270:FORII=1TOS(5):GOSUB390
   :NEXTII,JJ            :rem 65
100 GOSUB510             :rem 167
110 IFPEEK(211)>38THENGOSUB580
                         :rem 32
120 GOTO50               :rem 49
130 REMARK-CHANGE ASCII TO POK
    E VALUE{14 SPACES}-OR- PRI
    NT SPECIAL CHAR      :rem 136
140 BB=ASC(D$):IFBB>143THEN230
                         :rem 205
150 IFBB=16THENGOSUB880 :RETUR
    N                    :rem 182
160 IFBB=24THENGOSUB1070:RETUR
    N                    :rem 222
170 IFBB=21THENSO=SE(1):SR=SE(
    2):IS=SO             :rem 95
180 IFBB=12THENSO=SE(3):SR=SE(
    4):IS=SO             :rem 100
190 IFBB>132THENIFBB<141THENGO
    SUB850:RETURN        :rem 251
200 IFBB=13ORBB=141THENGOSUB55
    0:RETURN             :rem 161
210 IFBB=18THENIS=SR:RETURN
                         :rem 19
220 IFBB=129THENPRINT"{13}";:RE
    TURN                 :rem 86
230 IFBB=146THENIS=SO:RETURN
                         :rem 68
240 IFBB<32THENPRINTMID$("
    {5 OFF}{WHT}{11 OFF}{DOWN}
    {OFF}{HOME}{8 OFF}{RED}
    {RIGHT}{GRN}{BLU}",BB+1,1)
    ;:RETURN             :rem 56
250 IFBB>=144 AND BB<160THENPR
    INTMID$("{BLK}{UP}{OFF}
    {CLR}{OFF}{2}{3}{4}{5}{6}
    {7}{8}{PUR}{LEFT}{YEL}
    {CYN}",BB-143,1);:RETURN
                         :rem 212
260 AA=(BBAND31)+0.5*(BBAND128
    ):IF(BBAND64)=0THENAA=AA+3
    2:RETURN             :rem 240
270 REMARK-FIND CHAR IN
    {27 SPACES}MEMORY    :rem 55
280 POKE56334,0:POKE1,51
                         :rem 86
290 KK=PEEK(IS+8*AA+JJ):LL=PEE
    K(IS+S(3)+8*AA+JJ)   :rem 63
300 POKE1,55:POKE56334,1
                         :rem 84
310 RETURN               :rem 116
320 REMARK-PRINT BINARY REPRES
    ENTATION{13 SPACES}FOSR TI
    MES AS LARGE         :rem 196
330 NN=64:FORMM=0TO3     :rem 215
340 PP=1+8*INT(KK/NN)+2*INT(LL
    /NN)                 :rem 245
350 KK=KK-INT(KK/NN)*NN:LL=LL-
    INT(LL/NN)*NN        :rem 202
360 PRINTMID$("{OFF} {OFF}{D}
    {OFF}{F}{OFF}{I}{OFF}{C}
    {RVS}{K}{RVS}{B}{RVS}{V}
    {OFF}{V}{OFF}{B}{OFF}{K}
    {RVS}{C}{RVS}{I}{RVS}{F}
    {RVS}{D}{RVS} ",PP,2);
                         :rem 4
370 NN=INT(NN/4):NEXT MM
                         :rem 193
380 PRINT"{DOWN}{4 LEFT}";:RET
    URN                  :rem 70
390 REMARK-PRINT BINARY REPRES
    ENTATION{13 SPACES}EIGHT T
    IMES AS LARGE        :rem 2
400 SP$=RIGHT$("{5 SPACES}",S(
    10))                 :rem 0
410 XX=KK                :rem 24
420 YY=256:FORX1=1TO8    :rem 21
430 YY=YY/2              :rem 153
440 IFXX>=YYTHENXX=XX-YY:PRINT
    "{RVS}"SP$"{OFF}";:GOTO460
                         :rem 177
450 PRINTSP$;            :rem 40
460 NEXTX1               :rem 98
470 IFS(9)=0THENIFJJ=7THENIFS(
    10)=1THENGOTO490     :rem 112
480 PRINT"{DOWN}";       :rem 185
490 FORT=1TOLEN(SP$):PRINT"
    {8 LEFT}";:NEXTT     :rem 219
500 RETURN               :rem 117
510 REMARK-ADVANCE TO NEXT POS
    ITION                :rem 206
520 IFS(6)>0THENFORT=1TOS(6):P
    RINT"{UP}";:NEXTT    :rem 103
530 IFS(7)>0THENFORT=1TOS(7):P
    RINT"{RIGHT}";:NEXTT
                         :rem 246
540 RETURN               :rem 121
550 REMARK-PRINT RETURN
                         :rem 246
560 IFS(8)>0THENFORT=1TOS(8):P
    RINT"{DOWN}":NEXTT:rem 180
570 RETURN               :rem 124
580 REMARK-PRINT TO NEW LINE
                         :rem 206
590 IFS(9)>0THENFORT=1TOS(9):P
    RINT"{DOWN}":NEXTT:rem 185
600 RETURN               :rem 118
610 REMARK-DATA FOR 4X4:rem 71
620 S(1)=6:S(2)=2:S(3)=1:S(4)=
    1:S(5)=0: S(6)=4:S(7)=4:S(
    8)=2:S(9)=1:S(10)=1
                         :rem 160
630 RETURN               :rem 121
640 REMARK-DATA FOR 4X8:rem 78
650 S(1)=7:S(2)=1:S(3)=0:S(4)=
    1:S(5)=0: S(6)=8:S(7)=4:S(
    8)=4:S(9)=3:S(10)=1
                         :rem 170
660 RETURN               :rem 124
```

```
670 REMARK-DATA FOR 4X16
                        :rem 128
680 S(1)=7:S(2)=1:S(3)=0:S(4)=
    1:S(5)=2:S(6)=16:S(7)=4:S(
    8)=4:S(9)=3:S(10)=1
                        :rem 222
690 RETURN              :rem 127
700 REMARK-DATA FOR 4X24
                        :rem 121
710 S(1)=6:S(2)=1:S(3)=0:S(4)=
    1:S(5)=3:S(6)=24:S(7)=4:S(
    8)=0:S(9)=0:S(10)=1
                        :rem 208
720 RETURN              :rem 121
730 REMARK-DATA FOR 8X8:rem 82
740 S(4)=2:S(5)=1: S(6)=8:S(7)
    =8:S(8)=4:S(9)=3:S(10)=1
                        :rem 49
750 RETURN              :rem 124
760 REMARK-DATA FOR 8X16
                        :rem 132
770 S(4)=2:S(5)=2:S(6)=16:S(7)
    =8:S(8)=8:S(9)=7:S(10)=1
                        :rem 108
780 RETURN              :rem 127
790 REMARK-DATA FOR 8X24
                        :rem 134
800 S(4)=2:S(5)=3:S(6)=24:S(7)
    =8:S(8)=0:S(9)=0:S(10)=1
                        :rem 87
810 RETURN              :rem 121
820 REMARK-DATA FOR 32X24
                        :rem 173
830 S(4)=2:S(5)=3:S(6)=32:S(7)
    =0:S(8)=0:S(9)=0:S(10)=4
                        :rem 84
840 RETURN              :rem 124
850 REMARK-CHANGE FONTS
                        :rem 188
860 ONBB-132GOSUB610,670,730,7
    90,640,700,760,820 :rem 12
870 RETURN              :rem 127
880 REMARK-PRINT SCREEN TO PRI
    NTER                :rem 163
890 SI$=CHR$(15)        :rem 86
900 RV$=CHR$(18):RO$=CHR$(146)
    :GR$=CHR$(8)        :rem 68
910 VR=1024             :rem 70
920 OPEN4,4:PRINT#4     :rem 127
930 FORCL=0TO24:AS$=SI$:QF=0:F
    ORRO=0TO39          :rem 243
940 SC=PEEK(VR+40*CL+RO)
                        :rem 162
950 GOSUB1000           :rem 223
960 AS$=AS$+CHR$(AS)    :rem 93
970 NEXTRO              :rem 128
980 PRINT#4,SI$+AS$+RO$+GR$
                        :rem 44
990 NEXTCL:PRINT#4,SI$:CLOSE4:
    RETURN              :rem 120
1000 REMARK-CONVERT PEEK TO AS
     CII                :rem 2
1010 IFSC>=128THENSC=SC-128:AS
     $=AS$+RV$          :rem 59
1020 IFSC<32ORSC>95THENAS=SC+6
     4:GOTO1050         :rem 208
1030 IFSC>31ANDSC<64THENAS=SC:
     GOTO1050           :rem 105
1040 IFSC>63ANDSC<96THENAS=SC+
     32                 :rem 203
1050 IFRIGHT$(AS$,1)<>RV$THENA
     S$=AS$+RO$         :rem 111
1060 RETURN             :rem 167
1070 REMARK-EXTENDED MODE
                        :rem 45
1080 SI$=CHR$(14):GOTO 900
                        :rem 137
                                ©
```

Commodore 64 Program Profiler

D.E. Walker

Interested in speeding up your Commodore 64 BASIC programs? This convenient utility tells you which program lines take the most time to execute so you can rewrite them to run faster. The utility is written entirely in machine language, but you don't need to understand machine language to use it.

BASIC 2.0, the version of BASIC used by the Commodore 64, is what purists call an *unstructured* programming language. While some other languages (like Pascal) force you to write every program in a predefined structure, BASIC gives you the freedom to create whatever structure you like. That can be an advantage, but it can also result in slow, inefficient code if the program spends a lot of time performing unnecessary GOTOs or GOSUBs or is otherwise poorly structured.

If you want to improve a program's efficiency, you could examine it line by line, looking for superfluous REMs, a group of single-statement lines that could be combined into one multiple-statement line, and so on. But that would probably produce uneven results. Many parts of the program are performed only once, or so infrequently that it doesn't matter whether they're efficient or not. What you want to look for are the heavily used routines—FOR-NEXT loops, subroutines that are called frequently with GOSUB, long sequences of IF-THEN tests, and so on—where the program itself spends the most time.

"64 Program Profiler" generates an automatic time report for any 64 BASIC program, making it easy to identify the areas where time savings may be possible. Though it's written in machine language, you can use it without knowing machine language at all. First, type in the program as shown below, then save it on disk or tape.

A Resident Efficiency Expert

When you run the BASIC loader program, it puts the Profiler ML code in memory beginning at location 49152. Now you're ready to put your built-in efficiency expert to work. Enter SYS 49152 and press RETURN. Profiler asks three questions. First, you can choose to print the profile information on the screen or the printer. Second, you can decide whether to output the report in numeric form or in graphic form as a simple bar chart. Finally, enter the sampling rate you want Profiler to use in evaluating your program. This value (a number

from 1–9) determines how frequently Profiler looks at your program as it runs. The lower the sampling rate, the more frequently the program is checked.

At this point you can load and run the BASIC program you want to profile. While the program is running, the Profiler keeps track of which line is being executed according to the sampling rate you selected. After the program ends, there will be a short delay. Then the Profiler prints out its report. In each case you'll see a series of line numbers, followed by numeric values (if you chose a numeric display) or a series of asterisks (if you chose the bar chart).

It's important to understand exactly what the report means. The Profiler doesn't merely count the number of times that a particular line executed. Instead, it tells you how many times it saw the line being executed at the designated sampling rate. Thus, program lines that execute very quickly may not show up on the report at all. That's fine: If a program line executes too fast to be detected, you don't need to worry that it's slowing down your program. What you're concerned with are the lines that show big time values—they mark the places where the program is doing most of its work.

Adjustable Sampling Rate

Depending on what your program does, you may need to adjust the sampling rate to get a useful report. The largest possible time value is 255. If nearly every line in the report shows a value of 255, then the sampling rate is too small (Profiler is looking at the program too frequently). Reactivate Profiler with SYS 49152, select a larger sample rate, and rerun the program. This time Profiler looks at the program less often, which results in smaller time values and a more useful basis for comparison.

On the other hand, if every line in the program has a time value under ten, and many important lines are missing altogether, the sampling rate is too large; you'll get a more meaningful report by using a smaller rate. Remember, the time values are meaningful only in relative terms. The most meaningful report is one that shows a wide distribution of values, rather than a cluster of extreme values at one end of the scale.

Of course, common sense comes into play as well. If you have a massive database program that takes six days to run, don't expect Profiler to report anything smaller than 255 for every line, even at the slowest sampling rate. However, you can call Profiler from within a program, just as you can from direct mode. Thus, you could profile an individual subroutine in a large program by inserting SYS 49152 at the beginning of the subroutine, and putting a STOP or END statement just before the subroutine terminates with RETURN. If you then activate the subroutine with an appropriate GOTO, Profiler treats it like a separate program.

Commodore 64 Program Profiler

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

```
100 PRINT"{CLR}{CYN}PROFILER":
    PRINT"{3 DOWN}PLEASE WAIT.
    .."                    :rem 174
110 A=49152:CS=0:FORI=ATOA+550
    :READB:POKEI,B:CS=CS+B:NEX
    T                      :rem 36
120 IFCS<>68166THENPRINT"ERROR
     IN DATA STATEMENTS.":STOP
                           :rem 46
130 DATA 169,131,160,192,32,30
    ,171,32,207,255        :rem 157
140 DATA 162,129,160,192,32,96
    ,192,176,237,169       :rem 237
150 DATA 179,160,192,32,30,171
    ,32,207,255,162        :rem 164
160 DATA 130,160,192,32,96,192
    ,176,218,169,209       :rem 232
170 DATA 160,192,32,30,171,?32
    207,255,201,13         :rem 99
180 DATA 240,10,201,48,144,200
    ,201,58,176,196        :rem 158
190 DATA 144,2,169,53,56,233,4
    8,10,141,126           :rem 18
200 DATA 192,120,169,234,141,2
    0,3,169,192,141        :rem 157
210 DATA 21,3,169,0,133,251,13
    3,253,169,195          :rem 62
220 DATA 133,252,133,254,88,96
    ,134,251,132,252       :rem 218
230 DATA 201,78,240,10,201,13,
    240,6,201,89           :rem 251
240 DATA 240,6,56,96,169,0,240
    ,2,169,1               :rem 75
250 DATA 160,0,145,251,24,96,0
    ,0,0,0                 :rem 203
260 DATA 0,147,80,82,79,70,73,
    76,69,82               :rem 99
270 DATA 32,32,32,32,32,32,32,
    13,13,79               :rem 58
280 DATA 85,84,80,85,84,32,84,
    79,32,80               :rem 104
290 DATA 82,73,78,84,69,82,63,
    32,40,89               :rem 108
300 DATA 44,78,47,67,82,41,58,
    32,0,13                :rem 29
310 DATA 79,85,84,80,85,84,32,
    72,73,83               :rem 103
320 DATA 84,79,71,82,65,77,63,
    32,40,89               :rem 101
330 DATA 44,78,47,67,82,41,58,
    32,0,13                :rem 32
340 DATA 83,69,84,32,83,65,77,
    80,76,69               :rem 112
350 DATA 32,82,65,84,69,32,40,
    49,45,57               :rem 94
360 DATA 41,58,32,0,173,127,19
    2,208,78,173           :rem 27
370 DATA 126,192,141,127,?192,
    65,157,208,106,169     :rem 75
380 DATA 1,141,128,192,165,57,
    160,0,209,251          :rem 65
390 DATA 208,7,200,165,58,209,
    251,240,28,165         :rem 125
400 DATA 251,197,253,208,6,165
    ,252,197,254,240       :rem 228
410 DATA 42,24,165,251,105,3,1
    33,251,165,252         :rem 104
420 DATA 105,0,133,252,24,144,
    213,200,177,251        :rem 144
430 DATA 201,255,240,5,24,105,
    1,145,251,169          :rem 54
440 DATA 3,133,251,169,195,133
    ,252,206,127,192       :rem 221
450 DATA 76,49,234,24,165,253,
    105,3,133,253          :rem 72
460 DATA 165,254,105,0,133,254
    ,160,0,165,57          :rem 62
470 DATA 145,253,165,58,200,14
    5,253,200,169,1        :rem 167
480 DATA 145,253,24,144,210,17
    3,128,192,240,213      :rem 6
490 DATA 169,0,141,128,192,169
    ,131,141,2,3           :rem 16
500 DATA 169,193,141,3,3,169,1
    3,141,119,2            :rem 219
510 DATA 169,1,133,198,24,144,
    178,120,169,49         :rem 133
520 DATA 141,20,3,169,234,141,
    21,3,88,169            :rem 221
530 DATA 131,141,2,3,169,164,1
    41,3,3,173             :rem 160
540 DATA 129,192,208,103,160,0
    ,177,251,170,200       :rem 205
550 DATA 177,251,32,205,189,17
    3,130,192,208,66       :rem 230
560 DATA 169,32,32,210,255,56,
    32,240,255,160         :rem 115
570 DATA 6,24,32,240,255,160,2
    ,177,251,170           :rem 13
580 DATA 169,0,32,205,189,169,
    13,32,210,255          :rem 73
590 DATA 165,251,197,253,208,6
    ,165,252,197,254       :rem 244
600 DATA 240,16,24,165,251,105
    ,3,133,251,165         :rem 103
610 DATA 252,105,0,133,252,24,
    144,182,173,129        :rem 157
620 DATA 192,208,43,76,131,164
    ,169,32,32,210         :rem 117
630 DATA 255,160,2,177,251,170
    ,169,42,32,210         :rem 113
640 DATA 255,202,208,250,24,14
    4,194,169,4,170        :rem 170
650 DATA 160,255,32,186,255,32
    ,192,255,162,4         :rem 126
660 DATA 32,201,255,24,144,134
    ,32,204,255,76         :rem 113
670 DATA 131,164,0,0,255,255,0
    ,0,255,255,0           :rem 255
```

Atari Typo Tool

Patrick Dell'Era

Correcting typing mistakes is much easier with this multifunction utility. It lists your program a line at a time, and separately displays the items in critical DATA statements. It also contains a machine language subroutine that quickly deletes any range of lines from a BASIC program. For all 400/800, XL, and XE computers with at least 16K RAM (tape) or 24K RAM (disk).

How many times have you burned the midnight oil searching for a typo in a program you've typed in? Although COMPUTE!'s "Automatic Proofreader" greatly reduces the chances of typos, it won't catch transposition errors, and it can't be used with listings published in other magazines or user group newsletters. Now there's help—"Atari Typo Tool."

This program individually lists on the screen the lines of your freshly typed-in program, ignoring DATA statements and lines that begin with REM. Because DATA statements can be the most critical parts of a BASIC program—they often contain the decimal numbers of machine language subroutines—Typo Tool displays DATA elements one by one in large-size characters.

At any time, when you spot a typo, you can enter edit mode to make corrections, then continue where you left off. After you've weeded out all the typos in a program, you can tell Typo Tool that you're finished. It then erases itself from memory, leaving only your program in RAM.

Preparing Typo Tool

To get started, the first thing to do is type in Typo Tool and make sure *it* doesn't contain any typos. It can't be used to check itself. Use the Automatic Proofreader and be extra careful to avoid transposition errors (such as DATA 196 instead of DATA 169).

Store a copy of Typo Tool on disk or tape with the LIST command, not SAVE or CSAVE. That is, LIST"D:*filename.ext*" for disk or LIST"C:" for cassette. It's important to use the LIST format because this lets you merge Typo Tool in memory with the errant program you'll be checking. It's also vital to save a copy of Typo Tool before running it for the first time, because it erases part of itself when you type RUN.

Now you're ready. Type in or load the program you want to examine for typos. Make sure this program doesn't use line number 0 or line numbers from 32100 upward—Typo Tool uses these line numbers, and it will replace the lines in the other program if there's a conflict. Then load Typo Tool, using ENTER"D:*filename.ext*" for disk or ENTER"C:" for cassette. Finally, type RUN. There's a short pause as Typo Tool loads a machine language subroutine into memory. Then you'll see the main menu on the screen.

The menu offers three choices: Line Lister, Data Reader, and Finished. You need to type only the first letter of your choice. When you type L or D, you'll be asked for a starting line number. If you want to begin checking lines or data from the beginning of your program, just press RETURN. If, however, you want to start at another point, type in the appropriate line number and press RETURN.

In the Line Lister mode, Typo Tool displays a single line of the program you're checking. Pressing the cursor up-arrow (without CTRL) lists the next higher line number. Pressing the cursor down-arrow lists the next lower line number. Pressing RETURN brings you back to the main menu. And pressing E enters the edit mode.

You'll notice that at the bottom of each line listed on the screen is the command CONT. As we'll show in a moment, this lets you leave edit mode and continue with the Line Lister mode.

Making Corrections

When you press E to enter edit mode, the message STOPPED AT LINE 32180 appears. Disregard this. To fix a typo in the listed line, cursor up to the line as usual and make your corrections. Press RETURN on the line to enter it into memory. Then press RETURN on the CONT command to continue with the Line Lister mode. Typo Tool relists the corrected line on the screen so you can verify that it contains no additional errors.

Since typos in lines beginning with REM do not affect a program's operation, Typo Tool ignores these lines. However, it does list lines in which REM follows another statement.

The Line Lister mode also ignores DATA lines. Since DATA numbers are especially susceptible to typos, Typo Tool has a special mode for checking these important statements. Enter the Data Reader mode from the main menu by pressing D. In this mode, Typo Tool displays each individual DATA item in double-size (GRAPHICS 2) characters in the middle of the screen. Just below, it displays the DATA line number and DATA item number. If the DATA item is longer than eight characters, the item is broken into pairs and displayed one pair at a time. The pair number of the item is shown below the DATA item number.

Typo Tool automatically displays each DATA item in sequence for a few seconds so you can sit back with the source listing and check for mistakes. Each time Typo Tool advances to the next DATA item, it sounds a buzzer. If you want the display to pause, press any key except RETURN or E. The RETURN key exits the Data Reader mode and returns you to the main menu. The E key enters the Data Reader's edit mode.

This mode is similar to the Line Lister's edit mode. The DATA line you were checking is listed on the screen followed by a CONT command. Below that is the line number, DATA item number, and, if appropriate, the DATA pair number. You'll also see the message STOPPED AT LINE 32325, which you can disregard. To fix the typo, cursor up to the DATA line and make your corrections as usual, then press RETURN to enter the line into memory. Press RETURN over the CONT command to leave edit mode and continue with the Data Reader mode.

When Typo Tool has listed the last line of your program or read the last item in the DATA statements, it displays the message THAT'S ALL THERE IS! and returns to the main menu. If you have no further checking to do, press F to tell Typo Tool you're finished. Instantly, Typo Tool erases itself from memory, leaving your corrected program behind. You can then save the program and try running it. If everything works OK, your mission is accomplished. Otherwise, you may have to reENTER Typo Tool and search for further mistakes.

Squeezing Memory

If you have 16K RAM and a cassette drive, you'll find you can use Typo Tool with programs that are about twice as long as itself before running out of memory. On a 24K computer with a disk drive, you'll be able to work on programs about five times as long as Typo Tool. If you're having memory problems, there are a few things you can do to squeeze more room out of your available RAM.

Typo Tool includes a machine language subroutine that deletes blocks of BASIC lines (that's how it erases itself when you type F). Typo Tool also uses this routine to delete the block of DATA statements within it, freeing up a little more RAM. If you see an ERROR 2 (insufficient memory) message when you try to use Typo Tool with your program, follow these steps:

1. Type NEW to clear memory, LOAD the program to be checked, and LIST it to disk or cassette.

2. ENTER Typo Tool from disk or cassette and type RUN. After the menu appears, press BREAK. If you type LIST now, you'll see that all the DATA statements are gone, as are the lines that POKEd them into memory.

3. ENTER the program you want to check from disk or cassette. Type RUN. With any luck at all, everything should work.

4. If you still get an ERROR 2 message, you'll have to start over and LIST your program in two or more parts to disk or cassette. To do this, type LIST"D:*filename.ext*",FIRST-,LAST or LIST"C:",FIRST,LAST where FIRST is the first line number of the block to be saved and LAST is the last line number of the block. For each part of your program, type NEW and repeat steps 2 and 3 above. When you've corrected all the parts, you can reunite them by ENTERing each block of lines into RAM. Then save the whole program onto disk or tape.

Bonus Block Deleter

As mentioned above, Typo Tool includes a machine language subroutine that instantly deletes any range of BASIC lines. The routine is stored in memory page 6, beginning at location 1536 decimal. This memory area is not erased when you type NEW, although it's frequently used by other programs for storing machine language routines, which may cause a conflict.

To use the block deleter on any program in memory, type the following line and press RETURN:

```
A=USR(1536,FIRST,LAST)
```

where FIRST is the first line number of the block to be deleted, and LAST is the last line number of the block. Be sure of these numbers before you press RETURN.

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

Atari Typo Tool

```
FP Ø GOTO 32100Ø
LD 32100Ø DIM A$(106),TITLE$(
          23),T$(4),BLANK$(2Ø
          ):N7=Ø:NUL=1536:MRG
          =82:CURS=752
MD 32105 BLANK$="  ":BLANK$(2
          Ø)="  ":BLANK$(2)=BL
          ANK$:GOSUB 32380
DK 32110 GRAPHICS N7:OPEN #3
          ,4,N7,"K:":OPEN #2,
          4,N7,"E:":POKE MRG,
          15:POKE CURS,1
LJ 32115 POSITION 3,10:? "SE
          LECT:{5 SPACES}■ Li
          ne Lister":? :? "■
          Data Reader":? :? "
          ■ Finished{DOWN}"
CF 32120 POKE MRG,2:GET #3,A
          :IF A=76 THEN 32140
KD 32125 IF A=68 THEN 32250
JK 32130 IF A=70 THEN 32540
CP 32135 GOTO 32120
GP 32140 TITLE$="■■■■■■■■■■
          ":GOSUB 32400
HH 32145 GOSUB 32210
PJ 32150 IF LINE<A THEN GOSU
          B 32205:GOSUB 32200
          :GOTO 32150
BE 32155 IF LINE=32100 THEN
          32240
IL 32160 IF LINE=N7 OR TYPE
          THEN 32195
LH 32165 ? "{CLEAR}":POSITIO
          N 2,10:LIST LINE:?
          "CONT":GOSUB 32505
CO 32170 GET #3,S:IF S<>45 A
          ND S<>61 AND S<>155
           AND S<>69 THEN 321
          70
KH 32175 IF S=61 THEN GOSUB
          32215:GOTO 32155
AB 32180 IF S=69 THEN STOP
MB 32185 IF S=69 THEN 32165
OI 32190 IF S=155 THEN 32245
KE 32195 GOSUB 32205:GOSUB 3
          2200:GOTO 32155
EO 32200 LINE=PEEK(PRGM)+PEE
          K(PRGM+1)*256:TYPE=
          PEEK(PRGM+4)<2:RETU
          RN
OO 32205 B=PEEK(PRGM+2):PRGM
```

```
        =PRGM+B:RETURN
EJ 32210 PRGM=PEEK(136)+PEEK
         (137)*256:GOSUB 322
         00:RETURN
MG 32215 PRGM=PRGM-B:GOSUB 3
         2200
FB 32220 A=LINE:GOSUB 32210
PP 32225 IF LINE<A THEN GOSU
         B 32205:GOSUB 32200
         :GOTO 32225
PH 32230 IF TYPE THEN 32215
NP 32235 RETURN
LJ 32240 GRAPHICS N7:POKE CU
         RS,1:POSITION 10,10
         :? "THAT'S ALL THER
         E IS!":FOR I=1 TO 5
         00:NEXT I
P6 32245 CLOSE #2:CLOSE #3:P
         OKE 188,N7:GOTO 321
         10
LH 32250 TITLE$="DATA READER
         ":GOSUB 32400:GOSUB
          32435:GOSUB 32505
NH 32255 RESTORE A:TRAP 3233
         5
OJ 32260 POKE 764,255:READ A
         $:DITEM=PEEK(182):D
         LINE=PEEK(183)+PEEK
         (184)*256:IF DLINE>
         32099 THEN 32240
MJ 32265 TYPE=LEN(A$):IF TYP
         E>8 THEN 32275
OH 32270 GOSUB 32440:POSITIO
         N INT(8-(LEN(A$)/2)
         +0.5),3:FOR I=N7 TO
          20:POKE 53279,N7:N
         EXT I:? #6;A$:GOSUB
          32295:GOTO 32260
LJ 32275 IF TYPE/2<>INT(TYPE
         /2) THEN A$(TYPE+1)
         =BLANK$(1,1)
BL 32280 GOSUB 32440:FOR A=1
          TO LEN(A$) STEP 2:
         POSITION 7,3:FOR I=
         N7 TO 20:POKE 53279
         ,N7:NEXT I:? #6;A$(
         A,A+1)
IN 32285 POSITION 12,9:? #6;
         INT(A/2+0.4)+1:GOSU
         B 32295:NEXT A:POSI
         TION 1,9:? #6;BLANK
         $:POSITION 12,9:? #
         6;BLANK$(1,3)
DG 32290 GOTO 32260
CI 32295 FOR I=N7 TO 200:ON
         PEEK(764)<>255 GOTO
          32305:NEXT I
AE 32300 POSITION N7,3:? #6;
         BLANK$:RETURN
NG 32305 POP :GET #3,B:IF B<
         >155 AND B<>69 THEN
          GET #3,B:IF B<>69
         THEN 32300
DD 32310 IF B=155 THEN POP :
         GOTO 32245
EN 32315 GRAPHICS 0:POSITION
          2,7:LIST DLINE:? "
         CONT":GOSUB 32440
EM 32320 IF TYPE>8 THEN POSI
         TION 13,16:? INT(A/
         2+0.4)+1
EF 32325 STOP
AG 32330 GOSUB 32435:GOSUB 3
         2505:GOSUB 32440:RE
         TURN
EB 32335 IF PEEK(195)=6 THEN
          32240
KH 32340 GRAPHICS N7:? "ERRO
         R-  ";PEEK(195):END

AN 32345 DATA 216,104,104,14
         1,185,6,104,141,184
         ,6,165,136,133,203,
         165,137,133,204,32,
         136,6,165,203,133,2
         05,165,204,133
LG 32350 DATA 206,104,141,18
         5,6,104,24,105,1,14
         1,184,6,144,3,238,1
         85,6,32,136,6,56,16
         5,144,229,203,141,1
         82,6,165,145
PJ 32355 DATA 229,204,141,18
         3,6,56,165,203,229,
         205,141,186,6,165,2
         04,229,206,141,187,
         6,160,0,174,183,6,2
         40,14,177,203
MJ 32360 DATA 145,205,200,20
         8,249,230,204,230,2
         06,202,208,242,204,
         182,6,240,7,177,203
         ,145,205,200,208,24
         4,162,0,160,4
IN 32365 DATA 56,181,138,237
         ,186,6,149,138,181,
         139,237,187,6,149,1
         39,232,232,136,208,
         236,96,160,2,177,20
         3,141,188,6,136
IG 32370 DATA 177,203,136,20
         1,128,240,30,205,18
         5,6,240,4,176,23,14
         4,7,177,203,205,184
         ,6,176,14,24,173,18
         8,6,101,203
CH 32375 DATA 133,203,144,21
         5,230,204,208,211,9
         6,0,0,0,0,0,0,0,0
PK 32380 IF PEEK(NUL)=216 TH
         EN 32390
KJ 32385 RESTORE 32345:FOR I
         =NUL TO 1724:READ A
         :POKE I,A:NEXT I
BD 32390 A=USR(NUL,32345,323
         90)
OG 32395 RETURN
AL 32400 POKE CURS,N7:? TITL
         E$;:FOR I=N7 TO 100
         :NEXT I:?
BA 32405 IF TITLE$(8)="D" TH
         EN RETURN
HC 32410 ? "{UP}Press RETURN
          t{DOWN}**OR**{UP}
         {6 LEFT}o start at
         first line"
NJ 32415 ? :? "Enter specifi
         c line number to st
         art.":? :? "=>> ";
JC 32420 TRAP 32425:INPUT #2
         ,A:TRAP 40000:GOTO
         32430
AD 32425 A=N7
NM 32430 RETURN
DI 32435 GRAPHICS 2:POKE 712
         ,148:POKE 708,154:R
         ETURN
LF 32440 Q=PEEK(87):B=2:B=B+
         (TYPE>8):FOR X=1 TO
          B:RESTORE 32485:RE
         AD TITLE$:POKE 182,
         X:READ T$:TITLE$(6,
         10)=T$
LD 32445 POSITION 1+(Q=0),6+
         X+((Q=0)*7)
MB 32450 IF Q THEN ? #6;TITL
         E$;
CD 32455 IF  NOT Q THEN ? TI
         TLE$;
OO 32460 GOSUB 32475+X+((Q=0
         )*10):NEXT X
FB 32465 RESTORE DLINE:POKE
         182,DITEM:RETURN
EO 32476 ? #6;BLANK$(1,3):PO
         SITION 12,7:? #6;DL
         INE:RETURN
FH 32477 ? #6;BLANK$(1,3):PO
         SITION 12,8:? #6;DI
         TEM:RETURN
OI 32478 RETURN
OE 32485 DATA DATA
         {6 SPACES}#,line,it
         em,pair
MM 32486 ? DLINE:RETURN
NE 32487 ? DITEM:RETURN
OJ 32488 RETURN
OE 32505 Q=PEEK(87):POKE CUR
         S,1:IF  NOT Q THEN
         POSITION 2,20
IE 32510 ? "{5 SPACES}{ESC}
         {TAB}","RETURN FOR
         MENU"
HL 32515 IF  NOT Q THEN ? "
         {5 SPACES}{ESC}
         {TAB}","{ESC}{UP} N
         EXT HIGHEST LINE":?
          "{5 SPACES}{ESC}
         {TAB}","{ESC}{DOWN}
          NEXT LOWEST LINE"
GO 32520 IF Q THEN ? "
         {5 SPACES}{ESC}
         {TAB}{3 SPACES}ANY
         KEY TO PAUSE/RESTAR
         T"
OG 32525 ? "{2 UP}PRESS
         {2 DOWN}{ESC}{TAB}
         {3 SPACES}E TO EDIT
         ";
DI 32530 IF  NOT Q THEN POSI
         TION 5,14
DJ 32535 POKE CURS,N7:RETURN
MN 32540 TITLE$="FINISHED":G
         OSUB 32400:GRAPHICS
          N7:CLR :X=USR(1536
         ,N7,N7):X=USR(1536,
         32100,32540)
```

©

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."

ST Doodler

D.W. Neuendorf

This short, simple drawing program for the Atari ST demonstrates how to write a Logo program that takes advantage of GEM's built-in features and user interface. It works on any Atari ST with Logo.

When Atari first started shipping the 520ST last summer, ST BASIC wasn't quite ready, so the only programming language supplied was Digital Research's Logo. While borrowing a friend's 520ST, I decided to put Logo to work in a drawing program. This version of Logo, however, was translated from Digital's Logo for the IBM PC and doesn't run particularly fast on the Atari ST. After some experimenting, I realized it was too slow to support a full-fledged drawing utility.

Not yet ready to write off the ST/Logo combination, I considered alternatives. Calls to the operating system were out, since there was only a limited memory map and no CALL statement. Furthermore, Logo restricts the areas of memory accessible to the EXAMINE and DEPOSIT commands (similar to BASIC's PEEK and POKE). Even hand-assembled machine language routines are useless without a way to call them.

Then I remembered GEM—Digital Research's Graphics Environment Manager, which sits as a shell on TOS, the ST's operating system. In Logo, GEM has a Settings menu that lets you change line-drawing colors, line-drawing widths, fill colors and patterns, and other parameters. There must have been some reason why Atari included all these settings in a drop-down menu, duplicating many of the Logo commands. One very good reason, I concluded, was to avoid forcing someone like me to code a complex user interface in Logo, which would bog down the program. After all, the Settings menu is quite similar to the menu I planned to include in my drawing program.

Therefore, I took a hard look at what's *really* needed to draw pictures on the ST. Letting GEM handle the fancy features via its Settings menu, all we need is an easy way to fill areas and draw lines, circles, and boxes. Providing these functions are well within the capabilities of Logo. "ST Doodler" is the result.

Drawing With Doodler

To use ST Doodler, run Logo and type in the listing below. Be sure to save at least one copy before you try to use Doodler for the first time. Next, you should decide which screen resolution mode you wish to use. The monochrome screen gives you the highest resolution—and thus the capability to draw finely detailed pictures—but allows only black and white. The low-resolution mode allows 16 different colors, but with the loss of some detail. The medium-resolution mode offers more detail than low resolution, with up to four colors. If you wish to draw in a mode other than the one currently selected, you must quit Logo and return to the GEM desktop, then select Set Preferences from the Options menu. After making your selection, run Logo again.

To begin drawing with Doodler, clear the Graphics window by typing CS and pressing RETURN at a Logo top-level ? prompt, then type SKETCH. You'll probably want to expand the Logo Graphics window to full-screen size to give yourself more room to work. Doodler's pen color can be altered at any time, along with the background color, line width, fill color, and fill pattern settings, by dropping down the Settings menu and selecting the Graphics option. Click the pointer on the setting you want to change, then type in the appropriate number and click on the OK box.

This picture was created by the author using "ST Doodler."

To draw with Doodler, you connect a series of lines end to end. Choose the beginning of a series of line segments by moving the mouse pointer to the first desired endpoint and pressing the left button. (For all button presses in ST Doodler, hold down the button for at least a second or two; Logo cannot detect a faster press. If a Doodler command doesn't respond, you're probably not holding down the button long enough.)

You can specify subsequent endpoints the same way. You can draw a continuous line by holding down the left button while moving the mouse. However, because Logo isn't too adept at reading the buttons, you must move the mouse very slowly to draw smooth lines. To end the series of connected line segments, move the pointer outside the drawing area and press the left button. If you're using a full-size Graphics window, the best place outside the drawing area is the upper-right corner of the screen beyond the menu bar.

To fill an area with color, place the mouse pointer inside the area and press the right button. Be sure

the area you're trying to fill is completely enclosed by lines. If there are any holes, the fill "spills out" and colors the entire background.

To draw circles or boxes, press the right button while the pointer is outside the drawing area. A prompt asks you to press the C or B keys for a circle or box, respectively. Pressing any other key exits the circle/box mode. After pressing C to choose a circle, point to the desired center and press the left button; then move the pointer to the desired radius and press the button again. If you pressed B to choose a box, point first to its lower-left corner and press the left button; then point to its upper-right corner and press the button again.

You can erase portions of your drawing by dropping down the Settings menu, selecting the Graphics option, changing the line color to match the background color, then drawing over the parts you want to erase. You may also want to widen the line setting for this purpose.

How It Works

The top-level procedure, SKETCH, does a little initializing before invoking the main procedure, PT, which executes repeatedly. PT stores the current mouse status in the variable T, then analyzes it for the state of the left and right mouse buttons. Each mouse button has two functions, depending on whether the pointer is inside or outside the drawing area when the button is pressed.

Pressing the left button (indicated in ITEM 3 of MOUSE) specifies the endpoints in a series of connected line segments. DRAW? sets a flag, depending on whether ITEM 5 of MOUSE is TRUE (pointer inside the drawing area) or FALSE (pointer outside the drawing area). This flag, in turn, controls whether DR draws another line segment or sets a new starting point.

Pressing the right button (indicated in ITEM 4 of MOUSE) fills an area with the current fill pattern if the pointer is within the Graphics window boundaries (ITEM 5 of MOUSE is TRUE), or initiates circle or box drawing if the pointer is outside the Graphics window (ITEM 5 of MOUSE is FALSE). The circle and box prompts are drawn in the pen-reversed mode, then selectively erased by redrawing them in the same place. Although it can be hard to read these prompts over existing screen graphics, the alternative—printing to the Dialog window—stops the program.

You can save your artwork using the Save Pic option in the File menu, and reload previous drawings with the Load Pic option in that menu. When reloading pictures, you must set the screen for the same resolution that was in effect when the picture was saved. For example, you cannot load a picture drawn on the low-resolution screen into a medium-resolution Graphics window.

The lesson for programmers here is that programming on the ST will be very different than programming on earlier computers with traditional operating systems. Whether you're using Logo or a very fast compiled language, it would be a mistake to ignore the high-level tools available in GEM and TOS. Not only is it a waste of effort to write everything from scratch, but it's also wise to stick to the user interface which is already thoroughly familiar to every ST owner.

ST Doodler in Logo

```
TO SKETCH
  HIDETURTLE
  PENUP
  MAKE "GFILL "TRUE
  MAKE "TF 0
  PT
END

TO PT
  MAKE "T MOUSE
  IF (ITEM 3 :T) [DRAW?]
  IF (ITEM 4 :T) [BCORF]
  PT
END

TO DRAW?
  IF (ITEM 5 :T) [DR] [MAKE "TF 0]
END

TO DR
  IF (:TF = 0)
    [PENUP SETPOS :T MAKE "TF 1]
    [PENDOWN SETPOS :T PENUP]
END

TO BCORF
  IF (ITEM 5 :T)
    SETPOS PIECE 1 2 :T FILL]
    [BORC]
END

TO BORC
  MAKE "PCOL ITEM 5 TURTLEFACTS
  BCMSG
  MAKE "CH READCHAR
  BCMSG
  SETPC :PCOL
  IF (:CH = "B) [BX]
  IF (:CH = "C) [CIRC]
  MAKE "TF 0
END

TO BCMSG
  SETPOS [-70 80]
  SETHEADING 0
  TMSG [Circle: Press C]
  TMSG [Box: Press B]
  TMSG [Abort: Press any]
  TMSG [# # # # other key]
END

TO TMSG :MESSAGE
  PENREVERSE
  TURTLETEXT :MESSAGE
  PENUP
  BACK 18
END

TO BX
  MAKE "GFILL "FALSE
  BOX MBP
  MAKE "GFILL "TRUE
END

TO CIRC
  MAKE "GFILL "FALSE
  CIRCLE MCP
  MAKE "GFILL "TRUE
END

TO MBP
  GETPOINTS
  MAKE "PAR3 ABS ((FIRST :PAR2) -
    (FIRST :PAR1))
  MAKE "PAR4 ABS ((LAST :PAR2) -
    (LAST :PAR1))
  OUTPUT (SENTENCE :PAR1 :PAR3
    :PAR4)
END

TO MCP
  GETPOINTS
  MAKE "PAR3 (ABS ((FIRST :PAR2) -
    (FIRST :PAR1))) ^ 2
  MAKE "PAR4 (ABS ((LAST :PAR2) -
    (LAST :PAR1))) ^ 2
  OUTPUT SENTENCE :PAR1 SQRT
    (:PAR3 + :PAR4)
END

TO GETPOINTS
  MAKE "PAR1 GETPOS
  DELAY
  MAKE "PAR2 GETPOS
END

TO GETPOS
  MAKE "T MOUSE
  IF (ITEM 3 :T) [OUTPUT PIECE 1 2 :T]
  GETPOS
END

TO DELAY
  REPEAT 10 [MAKE "JUNK SIN 5]
END
```

Instant Apple Help Screens

Kent Brewster

With this short utility you can design and save your own custom help screens to disk, then quickly call them into BASIC programs. For all Apple II-series computers with DOS 3.3 or ProDOS.

As professional software designers have discovered, help screens are very popular features in all kinds of programs. Users don't have to fumble around with "handy" reference cards—they can just hit ESC or some other key and call up a screenful of instructions.

"Help Screen Editor," listed below, is a utility program that lets you create help screens of your own which are then saved to disk as binary files. Once saved, the file can be summoned back to the screen with a simple BLOAD command. If you BLOAD the file to an area of memory known as text page 2 with this statement:

BLOAD ***filename*****,A$800**

your help screen can be viewed and swapped with text page 1 (the normal screen) with this statement:

```
POKE -16299,0
```

The following statement will switch back to the normal screen display (text page 1):

```
POKE -16300,0
```

By placing these lines in a loop, the screens can be swapped some 30 times per second. Pretty impressive for BASIC.

For even greater convenience, you can use the following statement, which displays your help screen until a key is pressed, then switches back to the normal screen:

```
POKE -16299,0:GET G$:POKE -16300,0
```

Other uses for BLOADed screens include interactive software demonstrations and adventure games that show a screen and offer several options, each leading to another screen.

Designing A Help Screen

Any BASIC program that uses text page 2 must reserve space for that memory with POKE 104,12 and POKE 3072,0. This is the purpose of Program 1, which changes the bottom of program memory and calls Program 2, Help Screen Editor. (Don't confuse this process with changing the value of LOMEM, which merely relocates the bottom of variable memory, not program memory.) If you want, you can save Program 1 on a new disk with the filename HELLO so it automatically boots the editor when the computer is turned on.

Type in and save Program 2 (use the filename SE if you want the program to work properly with Program 1). It's quite short for a full-featured editor.

The first time you run Help Screen Editor, obviously there will be no help screen on the disk for it to load. Your first project should be to create a help screen for the screen editor itself, so temporarily modify the program to skip loading a help screen by inserting this line:

```
15 GOTO 40
```

The help screen you make for Help Screen Editor should look something like the example in the accompanying screen photo. Here are the screen editor commands:

CTRL-I	cursor up
CTRL-J	cursor left (or use the left cursor key)
CTRL-K	cursor right (or use the right cursor key)
CTRL-M	cursor down
CTRL-N	normal text
CTRL-R	reversed text
CTRL-F	flashing text
CTRL-S	save screen
CTRL-L	load screen
CTRL-C	clear screen
CTRL-P	print screen
CTRL-T	change title
CTRL-Q	quit editor
ESC = view help screen	

Once you've created a help screen, save it to disk by pressing CTRL-S and specifying a filename. For the screen editor's help screen, use the filename SEHELP. (The program automatically appends the filename extender .SCR when saving or loading screen files.) Then delete the temporary line 15 we added above, and you're ready to go. From now on, you can call up this help screen of screen editor commands merely by pressing ESC. Press any key to switch from the help screen back to the editor.

"Help Screen Editor" lets you add custom help screens to your own programs. This sample screen was created for use with the editor program itself.

Programming Notes

To add help screens to your own programs, follow these steps:

1. Be sure your program reserves space for text page 2 just as Program 1 does, with POKE 104,12 and POKE 3072,0.

2. To load the help screen into memory, your program should execute the command BLOAD *filename*,A$800. Make sure the filename corresponds with the name of the screen file on the disk.

3. To swap text page 2 with text page 1 and make the help screen visible, your program should execute the statement POKE −16299,0.

To return to the original screen, your program should execute the statement POKE −16300,0. If you want to make the help screen visible until any key is pressed, use POKE −16299,0:GET G$:POKE −16300,0.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" published this month in COMPUTE!.

Program 1: Screen Editor Loader

```
74 10 HOME : POKE 104,12: POKE 3
      072,0: PRINT CHR$(4);"RUN
      SE"
```

Program 2: Help Screen Editor

```
9E 10 HOME :R = 1:C = 1
74 20 PRINT CHR$ (4);"BLOAD SEHE
      LP.SCR,A$800"
4A 30 POKE - 16299,0: GET G$: PO
      KE - 16300,0
62 40 ST$ = "NEW SCREEN"
DC 50 DIM LS(23): FOR I = 1 TO 2
      3: READ LS(I): NEXT : DATA
       1024,1152,1280,1408,1536,
      1664,1792,1920,1064,1192,1
      320,1448,1576,1704,1832,19
      60,1104,1232,1360,1488,161
      6,1744,1872
B9 60 DIM CV(3): FOR I = 1 TO 3:
       READ CV(I): NEXT : DATA 3
      ,6,27
57 70 D$ = CHR$ (4)
56 80 GOSUB 590
DE 90 VTAB R: HTAB C: GET G$:G =
       ASC (G$): IF G > 31 THEN
      560
83 100 FOR I = 1 TO 3: IF G = CV
      (I) THEN 120
88 110 NEXT I: GOTO 130
85 120 ON I GOSUB 150,170,180: G
      OTO 90
49 130 V = G - 7: IF V < 1 OR V
      > 14 THEN 90
E6 140 ON V GOSUB 190,210,230,24
      0,260,300,320,330,340,430
      ,470,480,510,550: GOTO 90
32 150 VTAB 24: HTAB 1: PRINT "C
      LEAR THE SCREEN? (Y/N)";:
       GET G$: IF G$ = "Y" THEN
       HOME
A6 160 GOSUB 590: RETURN
DB 170 M = 2: GOSUB 590: FLASH :
       RETURN
76 180 POKE - 16299,0: GET G$: P
      OKE - 16300,0: RETURN
77 190 C = C - 1: IF C = 0 THEN
      C = 40:R = R - 1: IF R =
      0 THEN R = 23:C = 40
12 200 RETURN
B3 210 R = R - 1: IF R = 0 THEN
      R = 23
16 220 RETURN
9D 230 GOSUB 190: RETURN
98 240 C = C + 1: IF C = 41 THEN
       C = 1:R = R + 1: IF R =
      24 THEN R = 1:C = 1
1C 250 RETURN
17 260 NORMAL :M = 0: GOSUB 580:
       VTAB 24: HTAB 1: POKE 34
      ,23: PRINT "TITLE TO LOAD
      ? <RET> = QUIT ";: INPUT
      "";T$
E6 270 IF LEN (T$) < 1 THEN 290
CB 280 ST$ = T$: PRINT D$"BLOAD
      ";ST$;".SCR"
83 290 POKE 34,0: GOSUB 580: GOS
      UB 590: RETURN
6A 300 R = R + 1: IF R = 24 THEN
       R = 1
15 310 RETURN
1B 320 M = 0: GOSUB 590: NORMAL
      : RETURN
19 330 RETURN
D6 340 POKE 34,23: VTAB 24: HTAB
       1: PRINT "PRESS <RET> TO
       PRINT, OTHER TO ABORT.";
A5 350 GET G$:G = ASC (G$): IF G
       = 13 THEN 370
8E 360 PRINT D$"PR#0": POKE 34,0
      : GOSUB 580: GOSUB 590: R
      ETURN
AF 370 GOSUB 580: POKE 34,23
45 380 PRINT D$"PR#1": FOR I = 1
       TO 23: FOR J = 0 TO 39:P
       = PEEK (LS(I) + J)
1B 390 IF P < 192 THEN P = P + 6
      4: GOTO 390
B6 400 P = P - 128: IF P > 94 TH
      EN P = P - 64
98 410 PRINT CHR$ (P);
AC 420 NEXT J: PRINT : NEXT I: G
      OTO 360
D3 430 GOSUB 580: POKE 34,23: VT
      AB 24: HTAB 1: PRINT "QUI
      T THE EDITOR? (Y/N)";
F1 440 GET G$: IF G$ = "Y" THEN
      POKE 34,0: HOME : END
66 450 IF G$ = "N" THEN GOSUB 58
      0: GOSUB 590: RETURN
1F 460 GOTO 440
C9 470 M = 1: GOSUB 590: INVERSE
       : RETURN
23 480 IF LEN (ST$) < 1 OR ST$ =
       "NEW SCREEN" THEN GOSUB
      510
27 490 NORMAL : GOSUB 580: VTAB
      24: HTAB 1: PRINT "SAVE "
      ;ST$;"? (Y/N)";: GET G$:
      IF G$ = "Y" THEN GOSUB 58
      0: PRINT D$"BSAVE ";ST$;"
      .SCR,A$400,L$400"
AF 500 GOSUB 580: GOSUB 590: RET
      URN
EE 510 NORMAL : GOSUB 580: HTAB
      1: VTAB 24: PRINT "TITLE?
       ( <= 10 CHAR ) ";: POKE
      34,23: INPUT T$: IF LEN (
      T$) > 10 THEN PRINT CHR$
      (7): GOTO 510
C3 520 IF LEN (T$) < 1 THEN 540
DE 530 ST$ = T$
B5 540 GOSUB 580: POKE 34,0: GOS
      UB 590:R = 1:C = 1:M = 0:
       RETURN
9B 550 GOSUB 240: RETURN
9D 560 PRINT G$;:C = C + 1: IF C
       > 40 THEN C = 1:R = R +
      1: IF R > 23 THEN R = 1
3F 570 GOTO 90
6E 580 POKE 34,23: VTAB 24: HTAB
       1: PRINT :R = 1:C = 1: P
      OKE 34,0: RETURN
03 590 NORMAL : VTAB 24: HTAB 1:
       ON M + 1 GOSUB 610,620,6
      30
CA 600 PRINT " EDITING ";: INVER
      SE : PRINT ST$;: NORMAL :
       VTAB 1: HTAB 1: PRINT CH
      R$ ( PEEK (1024)): RETURN
66 610 PRINT "NORMAL  ";: RETURN
D3 620 PRINT "REVERSED";: RETURN
C3 630 PRINT "FLASH   ";: RETURN
```

©

IBM PrtSc Protector

Marc Sugiyama

If you've ever hit Shift-PrtSc by mistake and accidentally dumped a screen to the printer, you'll appreciate this short keyboard patch program. It works on any IBM PC with PC-DOS 2.0 or higher.

IBM has taken a good deal of flak over the years about the layout of the PC keyboard. One major complaint is the position of the PrtSc (Print Screen) key: It's next to the righthand Shift key. (This has been corrected on the PCjr and PC-AT.) If your finger goes astray and accidentally hits both Shift and PrtSc, the PC suddenly dumps the screen to the printer. It's particularly annoying when you're printing a long document or when you don't have a printer attached. If there's no printer, the PC locks up until it figures out that there's nothing to print to.

On the other hand, it's nice to have the screen dump capability handy when you need it. You wouldn't want to completely disable the function, but it would be nice if it were a little harder to call by accident.

"PrtSc Protector" offers a good compromise. It's a short machine language program that patches into the PrtSc function and distinguishes between the two Shift keys. If you press the *right* Shift key with PrtSc, nothing happens. If you press the *left* Shift key with PrtSc, you get the screen dump you really wanted.

The program below is a BASIC loader that creates the machine language file NOPRTSC.COM for PrtSc Protector directly on disk. If the BASIC loader detects any errors in the DATA (highly unlikely if you enter the program using COMPUTE!'s Automatic Proofreader"), it reports the mistake and erases the incorrect file. Because the file has the extension .COM, you can activate the program simply by typing the filename at a DOS prompt:

```
A> NOPRTSC
```

The resident portion of NOPRTSC.COM takes only about 320 bytes of memory. If you want to dump graphics screens to the printer, install NOPRTSC.COM after installing GRAPHICS. Don't try to install GRAPHICS more than once, or the computer will crash. Likewise, don't try to install NOPRTSC.COM more than once (why would you want to?). When NOPRTSC.COM is installed successfully, it returns a zero in the ERRORLEVEL variable; otherwise, it returns a one.

IBM PrtSc Protector

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

```
EF 100 CLS:LOCATE 10,10:PRINT"Wr
       iting file ..."
JA 110 OPEN "noprtsc.com" FOR OU
       TPUT AS #1
BP 120 FOR I=1 TO 279:READ BYTE:
       CKSUM=CKSUM+BYTE
GH 130 PRINT#1,CHR$(BYTE);
GD 140 NEXT I:CLOSE 1
OH 150 IF CKSUM <> 25301 THEN PR
       INT"** Error in DATA stat
       ements **":KILL "noprtsc.
       com":STOP
MK 160 PRINT:PRINT"File for nopr
       tsc.com has been created.
       ":END
NK  200 DATA 233,171,0,80,97,117
       ,108,80,83,81,82,86
KJ  210 DATA 87,85,180,2,205,22,
       168,2,116,6,156,154
HK  220 DATA 0,0,0,0,93,95,94,90
       ,89,91,88,207
LL  230 DATA 80,114,111,116,101,
       99,116,101,100,32,80,114
JB  240 DATA 116,83,99,32,105,11
       0,115,116,97,108,108,101
MM  250 DATA 100,46,32,32,83,104
       ,105,102,116,45,80,114
CL  260 DATA 116,83,99,32,117,11
       5,105,110,103,32,114,105
EI  270 DATA 103,104,116,45,115,
       104,105,102,116,32,100,10
       5
EG  280 DATA 115,97,98,108,101,1
       00,46,13,10,36,80,114
BF  290 DATA 111,116,101,99,116,
       101,100,32,80,114,116,83
LF  300 DATA 99,32,97,108,114,10
       1,97,100,121,32,105,110
AB  310 DATA 115,116,97,108,108,
       101,100,46,13,10,36,82
BN  320 DATA 101,113,117,105,114
       ,101,115,32,68,79,83,32
JE  330 DATA 50,46,48,32,111,114
       ,32,97,98,111,118,101
GE  340 DATA 46,13,10,36,0,0,180
       ,48,205,33,60,0
BG  350 DATA 117,9,186,143,1,180
       ,9,205,33,205,32,187
DJ  360 DATA 36,1,177,4,211,235,
       67,137,30,172,1,184
HB  370 DATA 5,53,205,33,137,30,
       24,1,140,6,26,1
EH  380 DATA 190,3,1,141,127,252
       ,185,4,0,252,243,166
LJ  390 DATA 131,249,0,116,33,18
       0,9,186,36,1,205,33
FK  400 DATA 184,5,37,186,7,1,20
       5,33,161,44,0,142
DD  410 DATA 192,180,73,205,33,1
       84,0,49,139,22,172,1
FO  420 DATA 205,33,186,106,1,18
       0,9,205,33,184,1,76
AB  430 DATA 205,33,0
```

Apple Error-Trapping

Ann Baldridge

You can add the professional touch to your BASIC programs by checking for common user errors—such as mistyped filenames, attempts to save data files on write-protected disks, invalid input, and the like. This article shows how any Applesoft program can be improved with proper use of the ONERR statement. The techniques apply to all Apple II-series computers with either DOS 3.3 or ProDOS.

Computers can be impolite. If you make the tiniest little nit-picking mistake, they balk with an error message and interrupt what you were trying to do. Or worse, they sometimes decide to freeze up and challenge you to a staredown.

This is bad enough when it happens to you. But if you write software for other people to use, part of your job is to protect them against the computer's insistence on perfection. One tool you can employ is an Applesoft BASIC command that is underutilized and even mysterious to many Apple programmers: ONERR (*on error*).

Picture a typical user who has just spent a half-hour entering information into your program. The program asks, SAVE TO DISK?. He taps Y in response, watches the drive's busy light come on, and then reaches over and pops opens the drive to make sure he's saving to the right disk—thereby aborting the save. A dumb mistake? Not for a beginner.

Although you may think he *deserves* to hear the drive's angry clacking sound and the error beep, and see the I/O ERROR warning and the now-mindless blinking cursor, you *can* keep all this from happening. If you inserted just one instruction before the save routine—ONERR GOTO 500—your program could be recovering from this mistake in whatever way you planned in line 500 and beyond.

Just because computers can sometimes be impolite doesn't mean your programs must be, too.

ONERR Tools And Rules

To save you the trouble of plowing through piles of programming manuals and reference guides, not to mention the hours you might spend experimenting, I've compiled a list of rules to follow when trapping errors in Applesoft programs. These are the results of my own trial-and-error efforts.

1. The ONERR command must be paired with GOTO, not GOSUB. Here's the proper syntax:

```
10 ONERR GOTO 1000
```

assuming, of course, that your error-handling routine begins at line 1000. If you try pairing ONERR with GOSUB, you'll get a syntax error.

2. The ONERR command must *precede* the statement where you anticipate an error may occur. For instance, if you want to protect against the disk drive input/output error described above, the ONERR command must be executed before the save routine. ONERR can be included in a multiple-statement line, but must be the *last* statement in the line. (I discovered that rule when some other statements following the ONERR command evaporated. Took me a while, too, because none of the books I read mentioned it.) Most programmers isolate ONERR GOTO on its own line. Now you know why.

3. When an error happens, the computer places a code number identifying the error into memory location 222. So ERRNUM=PEEK (222) yields a number that helps you plan what to do next. For instance, the numbers between 1 and 15 indicate a disk operation error. Some of these error codes are listed in the table below.

4. The command RESUME returns the program to the *beginning* of the statement or instruction where the error occurred—but not necessarily to the beginning of the line. It acts kind of like RETURN, except that it doesn't come back to the statement or line after the statement which called it. Let's say your program contains a line like this:

```
10 PRINT A:PRINT B:PRINT C
```

and an error occurs while PRINT B is executed. RESUME returns to the

PRINT B statement—not PRINT A or PRINT C. You can determine the line number where the error happened with this statement:

```
LINERR=PEEK(218)+PEEK(219)*256
```

5. You can turn off the ONERR command with POKE 216,0. If you do this too soon after an error occurs, however, the error code won't be stored in memory location 222.

6. There are some problems with ONERR in Applesoft. You get into a bunch of trouble if the error happens within a subroutine or a FOR-NEXT loop or, heaven forbid, both. For instance, suppose your program encounters a few errors while executing a load or save routine within a nested FOR-NEXT loop that is inside a nested subroutine. Boom. Reset City. There is a very short machine language fix that avoids these and other troubles. The routine is included (in the form of DATA statements) in lines 3000–3020 of Program 1. You must call this routine *before* you POKE 216,0. (I spent two days chasing a phantom syntax error before discovering this rule.) To be safe, make POKE 216,0 the last command before returning from the error-handling routine.

7. You can set a variable to either zero or one depending on whether the ONERR routine has been called. Known as *setting a flag*, this technique uses an IF-THEN statement to direct the route of the program.

Not A Cure-All

The purpose of ONERR, by the way, is to trap system errors which may occur as a program runs. You should not rely on it to trap your BASIC programming mistakes—errors like OUT OF DATA, TYPE MISMATCH, SYNTAX ERROR, and so on. Those kinds of errors should be caught when you test and debug the program. Too many lazy programmers use ONERR to cover their inability or unwillingness to find and correct their *own* errors.

The types of errors you should trap with ONERR are those which can be anticipated but not predicted: a data disk that is left out of the drive; a drive door that's left open; a mistyped filename; or a data disk filled to capacity. In these cases, ONERR routines protect the user from himself.

Putting Theory To Practice

Program 1 shows how to apply these rules. Note that this program is for illustration only, not to be typed in and run. Notice these variables:

ER is the error flag. When set to zero, it means no error has occurred. The program resets the flag to one when a problem exists.

EC is the error code number PEEKed from location 222.

Here's how the program works:

Line 10 sets up D$ as the DOS selector code, which saves keystrokes later on.

Line 20 sets up the machine language fixer for ONERR. This subroutine is called only once.

Lines 100 and 200 set the error flag (ER) to zero and prepare the ONERR routine to be called if necessary. Tip: Put the ONERR GOTO command(s) into a REM statement (e.g., 100 ER=0:REM ONERR GOTO 2000) until you've fixed all of your *own* errors. Otherwise, simple typos and logic mistakes will trigger ONERR and call your error-handling routine. When you're satisfied the program works the way you intended, remove the REM to activate ONERR.

Lines 110 and 210 ask the user to enter a filename.

Lines 120 and 220 are the important ones. You *must* put your error flag and IF-THEN statements into the line where you expect trouble to occur. If no error routine is called, the line goes on to execute normally. If an error does occur, then line 2000 executes (ONERR GOTO 2000).

Line 2000 begins the error-handling routine. It sets the error flag to one (ER=1); examines the error code (EC=PEEK(222)); calls the machine language fixer routine at memory location 768 in case the error was within a subroutine or FOR-NEXT loop (CALL 768); and tells the program to continue execution at the beginning of the statement where the error happened (RESUME).

What happens next is the key. When the program goes back to the appropriate statement (in this case either line 120 or 220), the error flag (ER) no longer equals zero. So, when the program reaches a statement that tests IF ER=0, execution drops to the next line (either 130 or 230). Lines 250 and 260 show how to use the error flag method in a multistatement line. Remember, RESUME returns to the beginning of the *statement* where the error was detected. So the flag must be set flag in a second location, too.

When execution drops through to line 130 or 230, GOSUB statements call subroutines which print specific information depending on the problem encountered. Then the ONERR function is turned off with POKE 216,0 before returning to let the user try again. You may wish to carry out this step just before the RESUME statement in the error-handling routine. And you'll certainly want to include it at the end of each disk access routine to stay aware of any other problems which may crop up.

Eliminating Hostility

Program 2 shows the changes you could make to lines 200–240 of Program 1 if you wanted to save text rather than program files. The error flag appears in line 220, which writes to the disk, since a DISK FULL or similar error is more likely then. Note that you don't have to close the file when the program drops through to the next line. When the Apple detects an error, it clears all of its file buffers, which makes closing unnecessary.

You may want to add an error flag to line 15, too. When the file is opened, an error could result if there's no disk in the drive or if the drive door is open. You could also precede each WRITE command with a trap, since a DISK FULL error bounces the program back to BASIC.

The error-message routines at lines 2100 and 2200 help the user find out what happened so he can correct the mistake before continuing. Although brief, they are written in a friendly tone. Frankly, beeping the computer and flashing I/O ERROR is not only hostile, but gives beginners no clue about how

to proceed.

This approach to error-trapping is quite flexible and adds a professional touch to any program. Don't overdo it, though. ONERR is best used to help people with problems you can't handle via skilled and careful programming.

Important DOS Error Codes

Code	Error Description
4	Write-protected disk
6	File not found
8	I/O error
9	Disk full
10	File locked

Note: A complete list of error codes is commonly found in many Apple manuals and books.

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

Program 1: ONERR Demo

```
09 10 D$ = CHR$ (4): REM SET APP
     LE DOS SELECTOR
09 20 GOSUB 3000
9A 100 ER = 0: ONERR GOTO 2000
62 110 INPUT "WHAT FILE WOULD YO
     U LIKE TO SEE?";FILE$
F7 120 IF ER = 0 THEN PRINT D$;"
     LOAD "FILE$: GOTO 140
A9 130 GOSUB 2100: POKE 216,0: G
     OTO 100: REM GO BACK TO T
     RY AGAIN
9C 140 PRINT "HERE'S YOUR FILE."
D5 199 REM SAVE A FILE
9B 200 ER = 0: ONERR GOTO 2000
87 210 INPUT "WHAT FILENAME WOUL
     D YOU LIKE TO USE?";FILE$
EA 220 IF ER = 0 THEN PRINT D$;"
     SAVE "FILE$: GOTO 240
B4 230 GOSUB 2200: POKE 216,0: G
     OTO 200: REM GO BACK TO T
     RY AGAIN
87 240 PRINT "YOUR FILE HAS BEEN
      SAVED":GOTO 4000
7B 1999 REM ONERR ROUTINES
48 2000 ER = 1:EC = PEEK (222):
     CALL 768: RESUME
79 2099 REM LOADING ERROR MESSAG
     ES
E3 2100 IF EC = 6 THEN PRINT "UN
     ABLE TO LOCATE FILE BY T
     HAT NAME.      PLEASE CHE
     CK TO SEE IF YOU SPELLED
      IT    CORRECTLY."
0C 2110 IF EC = 8 THEN PRINT "EI
     THER YOUR DISK DRIVE DOO
     R IS OPEN OR  YOU DON'T
     HAVE A DISK IN THIS DRIV
     E."
75 2120 GOSUB 2300: RETURN
C2 2199 REM SAVING ERROR MESSAGE
     S
47 2200 IF EC = 4 THEN PRINT "TH
     IS DISK IS WRITE-PROTECT
     ED.            PLEASE REM
     OVE THE WRITE-PROTECT TA
     B OR  PUT IN A NON-PROTE
     CTED DISK."
42 2210 IF EC = 8 THEN PRINT "EI
     THER YOUR DISK DRIVE DOO
     R IS OPEN OR  YOUR DISK
     IS BAD."
40 2230 IF EC = 9 THEN PRINT "TH
     IS DISK IS FULL. PLEASE
     INSERT A       DIFFERENT
     DISK."
7F 2240 GOSUB 2300: RETURN
ED 2300 PRINT : PRINT "PRESS ANY
     KEY TO TRY AGAIN.": GET
     K$: PRINT K$: RETURN
CB 3000 FOR I = 0 TO 9: READ MT:
     POKE 768 + I,MT: NEXT I
5F 3010 DATA 104,168,104,166,223
     ,154,72,152,72,96
DB 3020 RETURN
C5 4000 END
```

Program 2: Text Save Routine

```
A8 215 PRINT D$;"OPEN "FILE$
60 220 IF ER = 0 THEN PRINT D$;"
     WRITE "FILE$: GOTO 240
F3 230 GOSUB 2200: POKE 216,0: G
     OTO 200
C6 240 PRINT "OKAY, I'M SAVING Y
     OUR FILE NOW."
22 250 IF ER = 0 THEN PRINT A$:
     PRINT B$: PRINT C$: PRINT
      A: PRINT B: PRINT C: IF
     ER = 0 THEN 4000
F9 260 GOSUB 2200: POKE 216,0: G
     OTO 200
```

©

News & Products

Tax, Telecommunications Programs

Arrays, Inc., has released the 1985 version of *Tax Advantage*, an income tax preparation program for the Commodore 64 and 128; Apple II, II+, IIe, and IIc; Atari 800, 800XL, and 130XE; and IBM PC, XT, and AT computers. The package aids in preparing forms 1040, 6251, 2106, 2441, 4562, and schedules A, B, C, D, E, G, SE, and W. All information can be printed directly onto IRS forms and schedules. In addition, *Tax Advantage* performs income averaging, line itemization, and minimun tax calculations. The program is planned for ease of use, even for users new to computers and tax preparation, and works with Arrays' *The Home Accountant*.

Suggested retail price is $69.95. For an additional $10 warranty fee, users can purchase next year's update at a substantial price reduction.

Placing and answering calls on the Commodore 64 can be done automatically with *PhoneCall*, a new telecommunications program also from Arrays. It can be used to access online databases such as The Source or Compuserve, as well as other micro and mainframe computers. The program can send and receive both text and program files, and can convert CompuServe files to Commodore ASCII files.

PhoneCall's macro capability can store numbers, log on codes and billing information. It also includes a small bulletin board, and has a 26K buffer. The program comes with a tutorial and quick reference manual. Suggested retail price is $59.95.

Arrays, Inc./Continental Software, 6711 Valjean Avenue, Van Nuys, CA 91406.

Circle Reader Service Number 190.

Filing Program For Atari ST

AtariSoft and Stoneware, creator of the original *DB Master* database management system for several different microcomputers, have introduced a filing system for the Atari 520ST that is easy enough for first-time computer users. Called *DB Master One*, the system enables users to create forms with different colors and typestyles, or engage ready-to-use templates. Reports can also be easily generated. The program can hold up to 320K files, with 100 fields per form, 3,000 characters per record, four report forms, and ten report designs. Suggested retail price, $49.

Atari Corp., 1196 Borregas Ave., P.O. Box 3427, Sunnyvale, CA 94088-3427

Circle Reader Service Number 191.

Star Gazing On The Mac

Halley's Comet has been included in the new 512K Macintosh version of *Tellstar*, Spectrum HoloByte's computer astronomy software package. The program can track the comet from any location and on any date and time from 1980 to 1991. Also included in the program are displays and astronomical data on solar bodies, constellations, and Messier objects. *Tellstar Level I* includes all basic functions and one star table and retails for $49.95; *Tellstar Level II* comprises three detailed star tables and retails for $79.95. Versions for IBM and Apple II computers are also available.

Orbiter, another new release from Spectrum HoloByte, is a space shuttle simulation and game with 3-D graphics and voice synthesis that puts the player at the helm of NASA operations. Players earn points based on missions completed and performance throughout the entire space flight. *Orbiter* runs on the 512K Macintosh and sells for $49.95.

Spectrum HoloByte Inc., 1050 Walnut, Suite 325, Boulder, CO 80302.

Circle Reader Service Number 192.

Commodore, Apple II Word Processor

Better Working, the home productivity software brand from Spinnaker Software, has announced its *Word Processor with Spellchecker* for the Commodore 64/128, the Apple II series, and the Atari ST computers. *Word Processor* is the third product in the Better Working line, joining *Spreadsheet* and *File and Report*.

The new package combines a full-featured word processor with the 50,000-word *American Heritage Dictionary Spellchecker*. The three Better Working programs are integrated. Suggested retail for *Word Processor with Spellchecker* is $59.95 for the Apple II and ST versions and $49.95 for the Commodore version.

Better Working, One Kendall Square, Cambridge, MA 02139.

Circle Reader Service Number 193.

Prime Printers

A new low cost dot-matrix printer has been released from the Japanese company Citizen America Corp. Called the 120D, it features 120 characters per second (cps) printing, graphics capability, switch selectable IBM and Epson compatibility; a 25 cps correspondence quality mode; and a standard 4K buffer. List price is $249.

Another new printer from Citizen America, the Premiere 35, is a letter-quality daisywheel printer with a 35 cps printing speed and a low operating noise level of 55 decibels. It also has an 8K buffer, an LCD display of print functions and error messages, and selectable proportional spacing for justified text. List price is $599.

Citizen America Corp., 2425 Colorado Ave., Santa Monica, CA 90404.

Circle Reader Service Number 194.

New From Learning Well

Know Logo, an educational program for grades one and up, has been released by Learning Well. In a series of 50 games and discovery-based activities, *Know Logo* provides practice in turtle moves and turns, estimating angles, screen distances and positioning, and commands for circles and arcs. It requires a basic knowledge of Logo commands.

Also new from Learning Well is *Typing Well*, a typing tutor for children and adults. Players gobble up letters, create word pictures, and play table tennis as they sharpen their touch typing skills. The word-per-minute speed is automatically adjusted for each player, so that the program challenges without going beyond individual capabilities.

Each program runs on the Apple II series and lists for $49.95

Learning Well, 200 S. Service Rd., Roslyn Heights, NY 11577.

Circle Reader Service Number 195.

Commodore 64 Talking Text

The Votalker C-64 speech synthesizer from Votrax offers three types of text vocalization: conversation mode, which reads text as it's spoken; verbatim mode, which reads text and pronounces symbols; and character mode, which spells each word and pronounces numbers and symbols. It also has a screen echo that allows all words, numbers, punctuation marks and other symbols to be automatically spoken as they are printed to the terminal screen.

The four-by-five inch unit plugs into the Commodore 64 expansion port and contains its own amplifier, speaker, and external speaker jack. Suggested retail price is $99; and for a limited time, those who purchase the Votalker C-64 will receive a free copy of *Trivia Talker II*, Votrax's talking trivia game.

Votrax, Inc., 1394 Rankin Rd., Troy, MI 48083.

Circle Reader Service Number 196.

The Votalker C-64 voice synthesizer speaks text automatically and sells for $99.95.

Modula-2 Programming Language

TDI Software has announced the release of *Modula-2/ST* programming language for the Atari ST, in addition to versions for the Amiga and Macintosh computers. TDI *Modula-2/ST* comes with a full screen editor linked to the compiler, and flags all errors during compilation. It also has a full GEM interface, which enables the user to access GEM routines including GEM DOS, windows, mice, menus, and graphics.

Suggested price is $69.95.

TDI Software Inc., 10410 Markison Rd., Dallas, TX 75238.

Circle Reader Service Number 197.

Transparent Technology For Commodore

Transparent utilities—programs that run concurrently with other programs but only appear when called upon—are now possible for the Commodore 64 through Cardco, which recently introduced *StealthTec*, a transparent program interrupt technology on a cartridge.

Cardco's first product to use this technology is *Freeze Frame*, a transparent screen dump utility. A couple of keystrokes will send whatever is displayed on the computer screen to the printer. *Freeze Frame* is compatible with all programs and languages, and supports any printer or interface which emulates the Commodore 1525, as well as Epson- and Okidata-compatible printers. It retails for $49.95.

A second program in this line (unnamed at press time) is similar to Borland's *Sidekick* for the IBM-PC. The product offers access to things like a calculator, appointment calendar, telephone directory/database, and a memo writer. Any of the functions can be called up while another program is running. Suggested retail price is $69.95.

Cardco plans to make the *StealthTec* technology available for licensing by other software vendors.

Cardco, Inc., 300 S. Topeka, Wichita, KS 67202

Circle Reader Service Number 198.

Hippopotamus Introduces 14 ST Products

Hippopotamus Software has announced an initial line of 14 programs for the Atari ST. The products take advantage of the ST's GEM environment, incorporating pull-down menus, windows, and online help screens.

The line includes *HippoWord*, a mouse-based word processor ($89.95); *Hippo Concept*, an idea organizer compatible with *HippoWord* ($89.95); *HippoSimple*, a database manager ($49.95); *Hippo Disk Utilities*, compatible with floppy and hard disks ($49.95); *HippoBackgammon*, using full-color animated graphics ($39.95); *Hippo Computer Almanac*, a combination game/reference tool that contains more than 35,000 facts ($34.95); and *HippoPixel*, which allows users to create their own sprites and fonts ($39.95).

Hippopotamus Software, Inc., 985 University Ave., Suite 12, Los Gatos, CA 95030

Circle Reader Service Number 199.

Classics On Computer For Apple II

An educational game based on the book *Treasure Island* has been introduced by Classics On Computer. Appropriate for either home or classroom use, the program contains a high-resolution graphics game board, a short review game, and a special manual for teachers.

Treasure Island is designed to improve reading skills and build vocabulary. It is recommended for grades 5-9. Available for the Apple II, II+, or Apple IIe with minimum 48K memory, the program retails for $39.95.

Classics On Computer, 5150 Wilshire Blvd., Suite 502, Los Angeles, CA 90036.

Circle Reader Service Number 200.

Low-Cost Word Processor For IBM-PC

Dac Software, has announced *Dac Easy Word*, a versatile word processor for the IBM-PC.

The program features the ability to work on four different documents simultaneously (using DAC Windows); automatic hyphenation; file merges; automatic search; page numbering; and word count. It requires 256K memory on the IBM-PC, and retails for $49.95.

Dac Software, Inc., 4801 Spring Valley Rd., Building 110B, Dallas, TX 75244

Circle Reader Service Number 201.

SpeedScript Enhancer For 64

Upstart Publishing has released *Speedpak*, an enhancement to COMPUTE!'s popular *SpeedScript* word processor (Commodore 64 versions 3.0–3.2). The program adds six new commands, three printer codes, and eight user-definable 31-character macro phrase keys.

Additional features include alternate screens, which enable switching between and editing two documents instantly; a help screen and onscreen font installer, 32-character encryption, code conversion to Commodore ASCII or screen codes; default selection to disk/tape storage, set printer device and secondary address; and a Dvorak keyboard option.

Speedpak comes with printed instructions and includes three disk-based tutorials and three sample files. Price, $15.

Upstart Publishing, Dept. SP-NP2, P.O. Box 22022, Greensboro, NC 27420

Circle Reader Service Number 202.

Word Munchers On The Apple

A world of Munchers and Troggles awaits the player of *Word Munchers*, a new educational game for grades one through five, from Minnesota Educational Computing Corporation (MECC). Players move their Word Muncher around a game screen and direct it to eat words that have a particular vowel sound, while avoiding the enemy Troggles. *Word Munchers* becomes progressively more difficult at higher levels. Teachers can determine which vowel sounds are used and can control the level of word difficulty. About 1700 words of varying difficulty

can be used.

Word Munchers runs on all Apple II computers with at least 64K memory. Use of a joystick is optional. A support manual is included. Suggested retail price, $49.

Minnesota Educational Computing Corporation, 3490 Lexington Ave. North, St. Paul, Minnesota 55126-8097

Circle Reader Service Number 203.

***HomePak, BatteryPak* For Mac**

Batteries Included's *HomePak* and *BatteryPak* programs have been developed for the Macintosh. *HomePak* is a three-in-one telecommunications/ word processing/database manager program with macro command capability. *BatteryPak* is a set of nine accessories: Calendar, with Daytimer, keeps track of appointments and deadlines; the 250-page Phonepad stores and retrieves phone numbers; Automatic Modem/ Phone Dialer automatically dials any number listed in Phonepad or Calendar; Scientific Calculator includes statistical, logarithmic and trigonometric functions; RPM Calculator is for everyday calculating; a 7-function disk utility includes Trash, Copy, and Rename commands; High-Speed Launcher transfers to and from any program; Print Text creates draft copies while continuing to use the Mac; Windows Listing brings any window to the front of the screen.

Suggested retail price of *HomePak* is $69.95; for *BatteryPak*, $49.95.

Batteries Included, 17875 Sky Park North, Suite P, Irvine, CA 92714

Circle Reader Service Number 204.

ST Telecommunications

Online databases and bulletin boards can be accessed from the Atari 520ST using Atari's *Fastcom* telecommunications software. The program features integrated ASCII, VT100, and Viewdata modes; GEM drop down menus; transmission of text and binary files; macro commands; multi-tasking; full Prestel functions; printing of both graphics and text; and autodial and auto answer modem support.

The program comes with user guide and reference manual, and lists for $69.

Atari Corp., 1196 Borregas Ave., P.O. Box 3427, Sunnyvale, CA 94088-3427

Circle Reader Service Number 205.

Electronic Word Book

Richard Scarry's Best Electronic Word Book Ever!, a new program from CBS Software, is a picture and word game from children's author and illustrator Richard Scarry. Word recognition, vocabulary building, and objects recognition are developed as players travel with Lowly Worm to six different colorful environments: a farm, a town, a park, a railroad yard, a construction site and a harbor. In each environment, players learn to identify objects and associate them with their printed names. Animated graphics and familiar childhood tunes are used in each of the game's four skill levels. For the Apple II family and the Commodore 64 and 128. Suggested retail price, $19.95.

CBS Software, One Fawcett Place, Greenwich, CT 06836

Circle Reader Service Number 206.

ST Software From Spinnaker

Spinnaker Software has released Atari 520ST versions of several of their more popular programs. *Homework Helper Math* and *Homework Helper Writing*, two educational programs, sell for $49.95 each; *Amazon, Dragonworld, Fahrenheit 451, Nine Princes in Amber,* and *Perry Mason: The Case of the Mandarin Murder*, all graphics-and-text adventure games also sell for $49.95 each; *Treasure Island* and *Wizard of Oz*, from the Windham Classics series, sell for $39.95 each; and *Kung Fu: The Way of the Exploding Fist* sells for $39.95.

Spinnaker Software, One Kendall Square, Cambridge, MA 02139

Circle Reader Service Number 207. ©

HOTWARE: Software Best Sellers

This Month	Last Month	Title	Publisher	Remarks	Systems: Apple	Atari	Commodore	IBM	Macintosh
Entertainment									
1.	1.	*F-15 Strike Eagle*	MicroProse	Air combat simulation	●	●	●	●	
2.	5.	*Karateka*	Brøderbund	Action karate game	●	●	●		
3.	3.	*Jet*	Sublogic	Flight simulation	●	●	●		
4.	2.	*Flight Simulator II*	SubLogic	Aircraft simulation	●	●	●		
5.		*Ultima IIII*	Origin Systems, Inc.	Fantasy game	●	●	●	●	
Education									
1.	1.	*Typing Tutor III*	Simon & Schuster	Typing instruction program	●		●	●	●
2.	2.	*Math Blaster!*	Davidson	Introductory math program, ages 6-12	●	●	●	●	
3.	3.	*New Improved MasterType*	Scarborough	Typing instruction program	●	●	●	●	●
4.	4.	*Music Construction Set*	Electronic Arts	Music composition program	●	●	●		
5.	5.	*Sky Travel*	Commodore	Astronomy learning program			●		
Home Management									
1.	1.	*Print Shop*	Brøderbund	Do-it-yourself print shop	●	●	●		
2.	2.	*The Newsroom*	Springboard	Do-it-yourself newspaper	●		●	●	
3.	4.	*Print Shop Graphics Library II*	Brøderbund	Upgraded graphics library	●	●	●		
4.	5.	*Print Shop Graphics Library*	Brøderbund	100 additional graphics	●	●	●		
5.		*Three-In-One Bundle*	Timeworks	Word processor, spreadsheet, database manager			●		

SpeedCalc
For Apple II Computers

Kevin Martin

In response to popular request, COMPUTE! *presents this professional-quality spreadsheet program for all Apple II-series computers with either DOS 3.3 or ProDOS. Written completely in high-speed machine language, Apple* SpeedCalc *has all the important features you'd expect from a commercial spreadsheet program. In addition, its data files can be merged into text files created with the Apple* SpeedScript *word processor published last year in* COMPUTE!. *Apple* SpeedCalc *requires a disk drive, and a printer is optional but recommended.*

Have you ever planned a budget for your home or office? If so, you probably used some sort of worksheet divided into rows and columns. Perhaps you wrote the months of the year along the top of the sheet and listed categories for earnings and expenses along one side. After entering data for each category and month of the year, you could calculate total income figures by adding or subtracting numbers in each of the sheet's "cells."

That's a classic example of a worksheet. It lets you enter and organize data, then perform calculations that produce new information. A *spreadsheet* program is an electronic version of the familiar paper worksheet. Since it does all the calculations for you at lightning speed, an electronic spreadsheet is far more convenient than its paper counterpart. And spreadsheet programs also offer editing features that let you enter and manipulate large amounts of data with a minimum of effort.

Apple *SpeedCalc* is an all machine language spreadsheet program for Apple II computers with either DOS 3.3 or ProDOS. Though relatively compact in size, *SpeedCalc* is fast, easy to use, and has many of the features found in commercial spreadsheet programs. Even better, the "*SpeedScript* File Convertor" program lets you merge your *SpeedCalc* files into word processing documents created with *SpeedScript*, COMPUTE!'s popular word processor (see COMPUTE!, July 1985, or *SpeedScript: The Word Processor for Apple Personal Computers*, published by COMPUTE! Books).

Working together, *SpeedCalc* and *SpeedScript* make a powerful team. You can merge a chart of sales figures into a company report, create a table of scientific data for a term paper, and manipulate numeric information in many other ways. In a sense, a spreadsheet program brings to arithmetic all of the flexibility and power that a word processor brings to writing.

Preparing The Program

Although Apple *SpeedCalc* is small in comparison to similar commercial programs, it is one of the longest programs COMPUTE! has ever published. Fortunately, the "Apple MLX" machine language entry utility makes it easier to type a program of this size. Be sure to carefully read the Apple MLX article elsewhere in this issue before you begin.

We're publishing two separate versions of Apple *SpeedCalc*: Program 1 is for Apple computers with DOS 3.3, and Program 2 is for Apples with ProDOS. Be sure to type the correct version for your system, since the DOS 3.3 version doesn't work with ProDOS and vice versa.

Since the DOS 3.3 version of *SpeedCalc* resides in the same area of memory normally used by BASIC programs, you must relocate the BASIC program storage area *before* loading MLX to enter the data for *SpeedCalc*. If you're using DOS 3.3, enter the line below in direct mode (without a line number) and press RETURN:

```
POKE 104,38:POKE 9728,0:NEW
```

Then load and run MLX.

If you're using ProDOS, no special actions are required before loading and running MLX.

Here are the addresses you need to enter *SpeedCalc* with Apple MLX:

DOS 3.3:
Starting address: 07FA
Ending address: 24F9

ProDOS:
Starting address: 2000
Ending address: 3D67

After you finish typing, be sure to save at least one copy before attempting to run *SpeedCalc* for the first time. To start the DOS 3.3 version, first enter BLOAD SPEEDCALC (replace SPEEDCALC with the appropriate filename if you

From the publishers of *COMPUTE!*

February 1986 *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 Apple II, II+, IIe, and IIc computers. The February 1986 *COMPUTE! Disk* contains the entertaining and useful Apple II programs from the December 1985 and January and February 1986 issues of *COMPUTE!*. This easy-to-use disk also features *SpeedCalc*, the spectacular new spreadsheet program written entirely in machine language for the Apple II-series, and the latest version of *SpeedScript*, the bestselling word processing program.

The February 1986 *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 Apple II machine from the previous three issues of *COMPUTE!*.

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.

To subscribe to the *COMPUTE! Disk*, call toll free 1-800-247-5470 or write *COMPUTE! Disk*, P.O. Box 10036, Des Moines, Iowa 50950.

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

used some other name when saving the program). After the program loads, simply type RUN as you would for a BASIC program. To start the ProDOS version of *SpeedCalc*, first boot ProDOS, then enter −SPEEDCALC (replace SPEEDCALC with the appropriate filename if you used some other name when saving the program). This removes the BASIC interpreter and lets *SpeedCalc* take over the system.

If you're using an Apple IIe or IIc, be sure the Caps Lock key is down: *SpeedCalc* doesn't accept lowercase text input.

The Apple *SpeedCalc* Screen

SpeedCalc uses the top line of the screen as the *command line*. This is where *SpeedCalc* displays messages and asks you questions.

Screen lines 2–4 are the *input buffer* area. This is the work area where you enter and edit data. As you'll see in a moment, the input buffer also displays the data contained in the current cell. The work area cursor is an inverse less-than symbol (<). When the cursor is solid (nonblinking), *SpeedCalc* is waiting for a command or for data to be entered. After a character of data has been entered, the cursor begins blinking. While the cursor is blinking, most *SpeedCalc* commands (except for the cursor movement keys) are deactivated until you press RETURN to enter the data into the worksheet.

The lower 20 screen lines are your window into the spreadsheet. Though the spreadsheet contains many rows and columns, only a few can fit on the screen at one time. By scrolling the screen back and forth with the cursor, you can move the display window to any part of the spreadsheet.

The *SpeedCalc* worksheet consists of 50 vertical columns labeled with letters (AA, AB ... BX) and 200 horizontal rows numbered from 1–200. The rectangle where a row and column intersect is called a *cell*. Cells are where you store data. With 50 columns and 200 rows, the *SpeedCalc* spreadsheet has a maximum of 10,000 (50*200) cells. Due to memory limitations, however, only about a third of these can actually contain data. But you may spread out the data over all 10,000 cells if necessary, depending on the format you need.

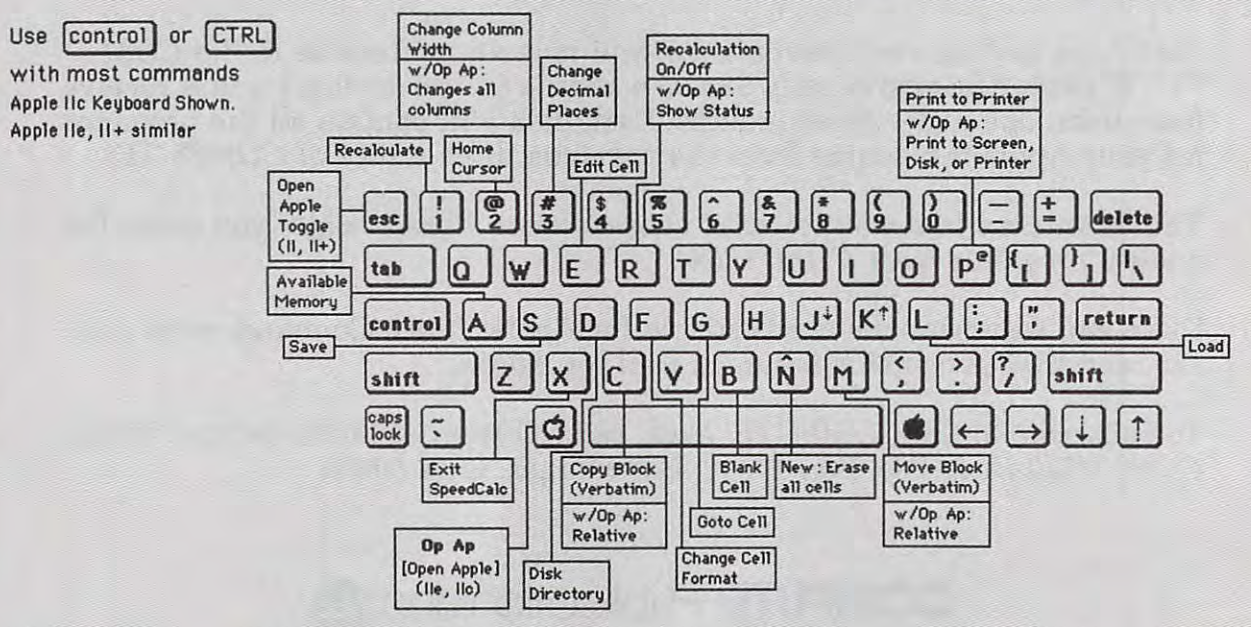
Moving The Cursor

Each cell is identified with the letters of its column and the number of its row. For example, the cell at the extreme upper-left corner of the sheet is called AA1, since it's in column AA and row 1. The cell below that is AA2. Moving one cell to the right from AA2 puts you in cell AB2, and so on.

Your current position in the spreadsheet is shown by the highlighted cursor. The simplest way to move around the sheet is with the cursor keys (on the Apple II or II+, use CTRL-K to move up and CTRL-J to move down). Another way to move the cursor is with CTRL-@ (CTRL-SHIFT-P for the Apple II or II+, or CTRL-2 for the Apple IIe and IIc.) Press CTRL-@ once to "home" the cursor on the current screen: The cursor moves to the upper-left cell. Press CTRL-@ twice in succession to move the cursor to cell AA1, the home position for the entire sheet.

SpeedCalc also has a *goto* command for moving the cursor over long distances. When you press CTRL-G, the command line displays GOTO: followed by an underline cursor. The underline cursor generally indicates that *SpeedCalc* is waiting for data—in this case it expects the name of the cell where you wish to go. If you enter **BA188** at this point, *SpeedCalc* moves the cursor to the cell at column BA in row 188, adjusting the screen window as needed. Take a few moments to practice moving around the spreadsheet with all three methods; you'll be using them a lot. In a later section, we'll discuss how to change the size and format of a cell.

SpeedCalc Keyboard Reference



Keyboard Commands

SpeedCalc offers many different commands, a few of which are entered by pressing one key. However, most commands are entered by pressing CTRL along with another key. CTRL-G, as you've seen, is the goto command. CTRL-A displays the amount of free memory available, and so on. The most drastic command is CTRL-X, which exits *SpeedCalc* and reboots the system. Since this effectively erases all data in memory, *SpeedCalc* prompts you with ARE YOU SURE Y/N? before it shuts down. To cancel the command, simply type N (or any key other than Y).

A few commands require you to press three keys at once. This sounds more awkward than it is in practice, since two of the three keys are Open Apple and CTRL. For instance, the *relative copy* command is performed by pressing Open Apple-CTRL-C (hold down Open Apple and CTRL, then press C).

The older Apple II and II+ models don't have an Open Apple key, so ESC is programmed to act as an Open Apple toggle. Pressing ESC once makes all following keypresses behave as if they were preceded by Open Apple. Pressing ESC again turns off this effect. In this article, wherever the instructions call for the Open Apple key, Apple II and II+ owners should instead precede the keypress with ESC, then use ESC again afterwards to disable the Open Apple toggle. For example, the command to check the recalculation status is Open Apple-CTRL-R; Apple II and II+ owners should instead press (and release) ESC, then press CTRL-R. There's no visible indication that the Open Apple toggle is in effect, so you must use ESC carefully or your keypresses will have unexpected results. For safety, always remember to press ESC again to toggle this function off after using a command that requires Open Apple. The table lists all the *SpeedCalc* commands, and the figure shows the keyboard layout with a description of what each key does. We'll be discussing each command in more detail below.

Three Data Types

Before entering any data, you must know what kind of data *SpeedCalc* accepts. There are three different types: numbers, text, and formulas. Let's look at each type in turn.

1. Numeric data consists of numbers—the basic stuff that spreadsheets work with. *SpeedCalc* has a few simple rules for numeric data: A number must be a decimal value (base 10, not hexadecimal) composed of one or more digits from 0–9, with an optional plus or minus sign. A decimal point is also optional. If you include any other characters in numeric input, *SpeedCalc* treats the entire input as text data (as explained below). Thus, the numbers 123, .001, and −65535 are valid numeric data. The number 65,535 is invalid because it includes a comma.

The allowable range for numbers in Apple *SpeedCalc* is the same as for Applesoft, roughly −1.7E38 to +1.7E38. If a calculation produces a number outside the allowable range, you'll see the message *ERROR* in the cell containing the formula. This doesn't happen very often, since *SpeedCalc* won't let you enter a number more than 36 digits long, and there's rarely a need to use such large numbers unless you're tracking the national debt.

Although an input value can be up to 36 digits long, numbers in *SpeedCalc* calculations are accurate only to nine digits. This must be taken into account when doing any calculation involving large values. For example, you can enter the value 1122334455.66 into a cell, and the cell holds the value with no rounding. However, if you use the value from that cell in a formula, the value is rounded to nine digits—1122334460.00—and the result of the calculation is accurate only for the first nine digits.

You can enter values in scientific notation by following a number with the letter E and the appropriate power of 10. For example, you can enter 1,234,000 as 1.234E06. However, *SpeedCalc* never uses scientific notation itself, no matter how big the number you enter. Scientific notation should generally be avoided, since values outside the Apple's maximum range will crash the program. (Press CTRL-RESET to recover.)

For example, let's enter the number 123 in cell AA1. No special commands are required to enter data: Just move the cursor to AA1 and begin typing. The blinking inverse < symbol shows the end of the data. While you're entering the number, it appears only in the input buffer near the top of the screen (the blinking underline shows your cursor position). As soon as you press RETURN, the number appears in AA1 and the letter N appears at the upper right of the screen. The N signifies *numeric*, meaning that *SpeedCalc* has accepted the entry as valid numeric data. Move the cursor to a vacant cell, then move it back to AA1. The input buffer displays whatever data is found in the cell under the cursor. When the current cell is empty, the buffer is empty as well.

If you want to change anything during data entry, press the ESC key. ESC always deletes the character before the cursor (or has no effect if the cell is empty). Later on, we'll explain how to edit existing data. Use ESC carefully; remember that when you're not editing (when the cursor is not blinking), ESC acts as an Open Apple toggle. On the Apple IIe and IIc, you can (and should) use DELETE instead of ESC.

As you've seen, pressing RETURN enters a data item into the current cell. You can also end the input by pressing a cursor key. The data is entered as if you had pressed RETURN, and the cursor moves in the indicated direction. This feature is handy for entering a lot of data: Simply type the entry, move the cursor to the next cell, enter more data, and so on.

2. Text data is not "data" in the strict sense, since *SpeedCalc* doesn't use it in calculations as it does numbers and formulas. Text data is there only to help people understand what the other data means. Text may consist of comments, titles, column headings, subheadings, or whatever you need to interpret the numbers and formulas. As an example, move the cursor to cell AA2 (just under AA1) and type the following line.

THIS IS A PIECE OF TEXT DATA.

You can use the ESC key (or DELETE on the Apple IIe and IIc) to erase mistakes while you're typing.

When you press RETURN, *SpeedCalc* displays T (for text) in the upper-right corner. In this example, the cell isn't long enough to accept all the text, so only the leftmost portion appears in AA2. But even though you can't see it, all of the text is there. Move the cursor to another cell, then move it back to AA2. As soon as you return to AA2, *SpeedCalc* displays all the text in the input buffer area.

3. Formula data is a mathematical expression or formula. It may be as simple as 2 + 2 or as complex as your imagination (and mathematical prowess) allows. The first character in a formula must always be an equal sign (=). If you omit this symbol, *SpeedCalc* either signals an error or treats the data as text.

The true power of a spreadsheet is that a formula in one cell can refer to another cell. This is easier to demonstrate than to explain. Move the cursor to cell AA3 and type the following line:

```
=AA1*25.01+@SQR(4)
```

As soon as you press RETURN, *SpeedCalc* displays F (for formula) in the upper-right corner and puts the *result* of the formula (not the formula itself) in AA3. If AA1 contains 123, the value 3078.23 appears in AA3. In plain English, this formula means "multiply the contents of cell AA1 by 25.01 and add the square root of 4." Before we examine the formula more closely, here's a quick demonstration of what makes a spreadsheet such a powerful tool. Move the cursor back to AA1 and press CTRL-R. The command line displays the message RECALCULATION IS ON, meaning *SpeedCalc* now automatically recalculates the entire sheet whenever you make a change. Now change the number in AA1 to 456 (simply move to the cell and start typing). The new result (11406.56) automatically appears in cell AA3. We'll explain more about automatic recalculation later.

Note that the referenced cell must contain data that *SpeedCalc* can evaluate: a number or another formula. If the formula refers to an empty cell, or one that contains text, *SpeedCalc* signals the error by printing *ERROR* in the cell containing the incorrect formula.

Mathematical Operators

These symbols can be used as *operators* in a formula:

Operator	Function
+	addition
−	subtraction
*	multiplication
/	division
^	exponentiation
=	equality

One factor that affects formulas is *precedence*, or the order in which mathematical operations are performed. In *SpeedCalc*, formula operators have the same precedence as in ordinary math.

The first operators to be evaluated—those with the highest precedence—are those enclosed in parentheses. Where one set of parentheses encloses another, the expression in the innermost set is evaluated first. The next operators to be evaluated are exponents. Multiplication and division have equal precedence; both operations are lower than exponentiation. Addition and subtraction have the lowest precedence of all. To take one example, *SpeedCalc* evaluates the formula =5*(8+3*−2)^2−10/+2 as the value 15, just as in ordinary math. Note how the result is affected by the plus and minus signs before the two 2's.

Functions

Formulas may also include any of the functions listed here:

@ABS( )	absolute value
@ATN( )	arctangent
@AVE( )	average of a block of cells
@COS( )	cosine
@EXP( )	natural exponent
@INT( )	integer
@LOG( )	natural logarithm
@SGN( )	sign
@SIN( )	sine
@SQR( )	square root
@SUM( )	sum of a block of cells
@TAN( )	tangent
PI	value of pi (3.14159265)

All the functions except PI begin with the @ symbol and are followed by parentheses. The parentheses of a function may contain a number or formula. For example, the formula =@SQR(4) generates the square root of 4. The formula =@SQR(AA1) returns the square root of whatever value cell AA1 contains. Note that the argument (value within parentheses) of the functions @TAN( ), @SIN( ) and @COS( ) must be expressed in radians; the result of the function @ARC( ) is expressed in radians. The function @INT( ) generates an integer (whole number) by truncating (discarding the fractional part of) a numeric value; note that this is different from rounding.

The function @AVE( ) calculates the mean average of the values in a block (group) of cells. The function @SUM( ) calculates the sum of a block. Both functions require you to define the block so *SpeedCalc* knows which cells to include in the calculation. This is done by putting two cell names separated by a colon in the parentheses. The first cell name defines the upper-left corner of the block, and the second defines the bottom-right corner. For instance, @AVE(AA1:AD20) calculates the average of all the cells from AA1 to AD20. The function @SUM (AA1:AD20) calculates the sum of AA1 through AD20, and so on. An error results if any cell in the block is blank or contains text data.

Editing The Sheet

Editing is a very important spreadsheet function. The simplest way to change what a cell contains is to move to it and start typing. The old data in that cell is replaced by whatever you enter. For instance, to replace the contents of cell AA1 with the number 456, move to that cell, type 456, and press RETURN or exit with a cursor key. Press CTRL-B (think of *blank*) to erase what's in the current cell. To erase everything in the sheet, press CTRL-N (think of *new*). Before carrying out this drastic operation, *SpeedCalc* asks you to confirm it by pressing Y or N.

In some cases, only a minor change is needed. *Edit mode* lets you change the data in a cell without retyping the entire entry. To activate edit mode, move to the desired cell and press CTRL-E. In this mode, up and down cursor movement is disabled, and the left/right cursor keys move within the input buffer. Erase unwanted characters with the ESC key (or the DELETE key on the Apple IIe and IIc). Typing in edit mode inserts new characters in the line: Everything to the right of the new character moves right one space (unless the buffer is already full). Since the cursor keys

have a different function in edit mode, you cannot use them to end the input. Press RETURN to enter the new data and escape from edit mode.

SpeedCalc displays *ERROR* in a cell when you enter an erroneous formula. Usually this means you've made a typing error in that cell, or the formula refers to text or an empty cell. A line of asterisks (*********) signals that a number is too large to be printed in the cell. Though these messages appear in the cell area, no data is lost. You may move to the affected cell, view its contents in the input buffer, and make whatever correction is needed.

Recalculation

This feature is the very core of a spreadsheet. As you know, entering or editing a piece of data makes *SpeedCalc* perform a calculation and put the result in the cell under the cursor. In most cases, the new data relates to data in other cells, so you'll ultimately want to recalculate the entire spreadsheet as well. This can be done manually or automatically.

To recalculate the spreadsheet manually, enter an exclamation point (SHIFT-1). *SpeedCalc* begins at AA1 and recalculates every cell that contains data, placing fresh results wherever needed. If you switch to automatic recalculation mode, *SpeedCalc* automatically recalculates the entire spreadsheet each time you enter new data or edit what exists. When you press CTRL-R, *SpeedCalc* changes the recalculation status and displays it at the top of the screen. If automatic recalculation was turned off before, it is now on (and vice versa). If you aren't sure which mode you're in, press Open Apple-CTRL-R; *SpeedCalc* displays the mode without changing it.

Automatic recalculation can be fun to watch in a large spreadsheet: Every time you make a change, new results appear everywhere on the screen. However, the more data your spreadsheet contains, the longer it takes to update the entire sheet. For this reason, you may want to turn off automatic recalculation most of the time, recalculating manually whenever you need to view results.

One problem with recalculation arises from the order in which cells are calculated. Because only one cell can be calculated at a time, you must sometimes recalculate the entire spreadsheet two or three times to get correct results in every cell (this is common to all spreadsheet programs). For instance, say you have a formula in AA1 which refers to a formula in AB15. When *SpeedCalc* calculates AA1, it must use the existing data from AB15—which is probably out of date, since the formula in AB15 hasn't been recalculated yet. To avoid this problem, you should always recalculate a sheet manually two or three times before printing or saving it to disk.

SpeedCalc offers a number of other features. Before experimenting with them, you should spend some time typing in a hypothetical spreadsheet—perhaps a fictitious yearly budget—to become thoroughly familiar with the basic commands covered so far. Most importantly, create formulas using all the operators in different combinations. Try doing things that you know will cause errors. Then correct the errors in edit mode, and so on. It takes a thorough grasp of the fundamentals to get the most out of *SpeedCalc*'s advanced features.

Change Format

The default (normal) format for numeric data is flush right with rounding to two decimal places. In other words, the number is displayed in the rightmost part of the cell, with two numbers after the decimal point. Text and formulas are also displayed flush right. *SpeedCalc* offers several commands for changing cell formats. (Apple II and II+ owners who are using the ESC toggle in place of the Open Apple key should be careful that ESC is not in effect when it's not desired; accidental global changes may be difficult to reverse.)

Change Format (CTRL-F). This command changes the location of data in the cell. When you press CTRL-F, the *SpeedCalc* command line displays the question FORMAT: LEFT, CENTER, OR RIGHT JUSTIFY?. Press L, C, or R to move the data to the left, center, or right of the cell.

Change Decimal Places (#). *SpeedCalc* also lets you change the number of decimal places for any cell. The default number of decimal places is 2, but you may change it to anything from 0–15. Press # (SHIFT-3) to change this value: *SpeedCalc* prompts you to enter a number from 0–15. If you choose zero decimal places, any number in that cell is rounded off to the nearest integer (whole number). If you choose 15, a number in that cell is not rounded off at all—*SpeedCalc* displays it exactly as you entered it or as it was calculated from a formula.

Width (CTRL-W). The width command changes the width of an entire column of cells. Move the cursor to any cell in the desired column, then press CTRL-W. When *SpeedCalc* displays the prompt WIDTH:, respond with a number from 4–36. The entire screen is redrawn to accommodate the new format, and may look very different depending on what value you chose. For instance, if you increase a column's width, the rightmost column of the former display may disappear: *SpeedCalc* only displays as many complete columns as it can fit on the screen. If you decrease the width of a column, you may see asterisks where numbers used to be (indicating the cell is now too small to display the entire number). To get rid of the asterisks, expand the column as necessary.

Global Format (Open Apple-CTRL-F). This is the same as the ordinary format command, but operates globally, changing every cell in the sheet instead of just one.

Global Width (Open Apple-CTRL-W). This is a global version of the width command. Every column in the sheet changes to the designated width.

Macro Editing

After typing in a large spreadsheet, you may decide to make a major change. You may want to add new data somewhere in the middle, delete a section, or move a group of cells from one location to another. *SpeedCalc*'s macro (large-scale) editing commands simplify such operations, affecting an entire block of cells at once. A *block* is simply a group of cells connected in rectangular fashion. You can define it as a single cell, a row or column, or any

rectangular area within the spreadsheet.

There are two ways macro commands work: *verbatim* or *relative*. To take a simple example, say that cell AA2 contains the formula =AA1*5 and you want to move its contents to cell AB2. When this is done in verbatim mode, AB2 contains an exact copy of what was in AA2 (=AA1*5). Note that the cell name used in the formula does not change: The formula still refers to AA1. If you perform the same operation in relative mode, the cell name in the formula is adjusted to fit the new location. In this case, AB2 would contain the formula =AB1*5. (Apple II and II+ owners who are using the ESC toggle in place of the Open Apple key should be careful that the toggle is not in effect when not desired; accidental relative changes can lead to problems that are difficult to detect and correct.)

Copy (CTRL-C). The copy command copies a block of cells into a different location without disturbing the original cells. Place the cursor on the upper-left corner of the block you want to copy, then press CTRL-C. *SpeedCalc* prompts you to move the cursor to the lower-right corner of the block you want to copy. Once the cursor is in place, press RETURN. Now *SpeedCalc* prompts you to move the cursor to the place where you want to put the block: This is the upper-left corner of the new position. Once the cursor is there, press RETURN again. The new data replaces whatever was contained in the designated cells. Note that if you define an impossible block (for instance, moving the cursor to the upper-left of the original position, rather than below and to the right), *SpeedCalc* does not copy any data. This provides a way to cancel the command if you press CTRL-C accidentally.

Move (CTRL-M). This command works like a copy, but it fills the original cells with blanks. Though *SpeedCalc* has no express insert command, you can use this command to make space for new data in the middle of a spreadsheet. Simply move everything below the insertion point down as far as you need.

Because RETURN generates the same character code as CTRL-M, you may find when you first begin using *SpeedCalc* that you accidentally invoke the move function by pressing RETURN when you shouldn't have. To cancel this, simply press RETURN twice more without moving the cursor.

Relative Copy (Open Apple-CTRL-C). This form of the copy command adjusts the cell names used in formulas within the copied block (see explanation above).

Relative Move (Open Apple-CTRL-M). This is the relative form of the move command. Cell names in formulas are adjusted to reflect the move.

Memory Management

The DOS 3.3 version of *SpeedCalc* makes about 12K (over 12,000 characters) of memory available for data; the ProDOS version provides approximately 17K. As noted earlier, *SpeedCalc* lets you spread your data out over a much larger number of cells than you can actually fill with data. The extra space is provided to give you full control over the final format of the spreadsheet and to leave some elbow room for move and copy operations.

Because memory is limited, you should keep careful track of how much is free while using the program. Press CTRL-A to display the amount of free memory. We suggest limiting your spreadsheets to 1,600 cells (equivalent to 40 rows by 40 columns) when using the DOS 3.3 version, or 2,500 cells (a 50 × 50 worksheet) when using the ProDOS version. If you've filled nearly all of free memory, you may have to break the spreadsheet into two smaller sheets.

Although *SpeedCalc* checks the amount of available memory and displays an error message if you run out, you should be careful not to exhaust free memory. Any move or copy operation in process will be aborted if sufficient memory is not available.

Disk Operations

SpeedCalc has three disk commands which allow you to save a spreadsheet to disk, load it, and display the disk directory. The directory command is the simplest to use: Simply press CTRL-D. The spreadsheet disappears and a directory of the disk in drive 1 is displayed. Press RETURN to return to the spreadsheet.

To save a spreadsheet to disk, press CTRL-S. *SpeedCalc* prints

SpeedCalc Commands

Command	Action
CTRL-A	available memory check
CTRL-B	blank (erase) current cell
CTRL-C	copy block verbatim
CTRL-D	disk directory
CTRL-E	edit current cell
CTRL-F	change cell format
CTRL-G	goto selected cell
CTRL-L	load *SpeedCalc file*
CTRL-M	move block verbatim
CTRL-N	new (erase entire sheet)
CTRL-P	print file on printer
CTRL-R	turn recalculation on/off
CTRL-S	save *SpeedCalc* file
CTRL-W	change column width
CTRL-X	exit *SpeedCalc*
CTRL-@	home cursor
Open Apple-CTRL-C	copy block relative
Open Apple-CTRL-M	move block relative
Open Apple-CTRL-P	print to screen, disk, or printer
Open Apple-CTRL-R	check recalculation status
Open Apple-CTRL-W	change width of all columns
! (SHIFT-1)	recalculate sheet
# (SHIFT-3)	change decimal places

Note: The Apple II and II+ have no Open Apple key, so ESC must be used as an Open Apple toggle. Pressing ESC once makes all following keypresses behave as if Open Apple were pressed. Press ESC again to turn off the Open Apple toggle.

SAVE: on the command line, followed by an underline cursor. Enter a valid Apple filename and press RETURN. (If you change your mind and decide not to save anything, press RETURN without typing a filename.) If no disk error occurs while the spreadsheet is being saved, *SpeedCalc* displays NO ERRORS in the command line and returns you to command mode. If there was an error, you'll hear a beep and see the message I/O ERROR in the command line.

To load a saved file from disk, press CTRL-L. Again, you can cancel the operation by pressing RETURN without entering a filename. *SpeedCalc* prompts you to enter the filename and displays the error status when the operation is complete.

When saving or loading *SpeedCalc* files with ProDOS, you must specify the prefix along with the name. If you don't want to type the prefix every time you enter a filename, simply call up a directory for the disk you want to use to save or load. This automatically sets the prefix to match the current disk, relieving you of the need to enter it with every name.

Printing

SpeedCalc lets you print data to three different devices: to the screen for previewing output, to a printer for permanent documentation, or to a disk file for integrating the data with a *SpeedScript* document.

To print a hardcopy of the spreadsheet to a printer in slot number 1, press CTRL-P. *Before using this command, you must position the cursor below and to the right of the block of cells you wish to print.* The upper-left corner of the printout starts at cell AA1.

To send output to a printer with a slot number other than 1, or to the screen or a disk, first position the cursor in the lower-right corner of the block you want to print. Then press Open Apple-CTRL-P (toggle ESC on the Apple II and II+). *SpeedCalc* asks if you want to print to the screen, to disk, or to the printer. Press S to preview output on the screen, D to print to disk, or P to select printer output. Pressing any other key cancels the command.

If you select the P option after pressing Open Apple-CTRL-P, *SpeedCalc* asks you specify a slot number by pressing one of the number keys from 1–7. This permits you to use a printer in any of those slots. If you change your mind at any point during this process, press RETURN without entering anything; *SpeedCalc* returns you to command mode.

You can also print *SpeedCalc* data to a disk file for use in a *SpeedScript* document. Select the D option after pressing Open Apple-CTRL-P, then enter a filename. The data is saved as a disk file of that name. Note that *printing* to disk creates a different type of file than *saving* to disk, and *SpeedCalc* cannot reload files in the print format. You should *save* files you wish to reload into *SpeedCalc*, and *print* files you wish to convert for *SpeedScript*. Unlike the *SpeedCalc* save and load commands, no error messages are provided if the spreadsheet cannot be printed to disk. Thus, you must ensure that the drive contains a write-enabled disk with sufficient space to hold the printed spreadsheet before you attempt to print to disk.

SpeedScript File Converter

SpeedCalc sends data to the printer in simple, plain vanilla form. That may be fine for personal use, but if you're creating a document for others to view, you may want special features such as boldface, underlining, etc. Since Apple *SpeedScript*—COMPUTE!'s popular word processor—already offers a way to access these features (and many more), no attempt has been made to include them in *SpeedCalc*. All that's needed is a simple program to convert *SpeedCalc* files into a form that *SpeedScript* can load. Then you can edit the file with *SpeedScript* as you would any other document—inserting printer control codes, reformatting the text, merging it with other text, and so on. The "*SpeedScript* File Converter" program published in the same issue as *SpeedScript* makes it easy to perform the conversion. Here are the steps to follow to convert a *SpeedCalc* file for *SpeedScript:*

1. After creating a spreadsheet with *SpeedCalc*, print it to disk as described above.

2. Exit *SpeedCalc*, then load and run *SpeedScript* File Converter. The program prompts you to enter the name of the *SpeedCalc* file you printed to disk. Then it asks you to enter the name of the *SpeedScript* file you want to create (of course, this name should be different from the first). The File Converter then constructs a *SpeedScript*-loadable disk file from the *SpeedCalc* file.

3. After the File Converter is finished, load and BRUN *SpeedScript*, then load the new *SpeedScript* file as you would any *SpeedScript* document. The data appears on the screen, ready to be edited in any way you wish. ©

Program 1: Apple *SpeedCalc* For DOS 3.3

Please refer to the "MLX" article in this issue before entering the following listing.

```
START ADDRESS: 07FA
END ADDRESS:   24F9

07FA: 20 65 D6 4C D2 D7 00 0A 12
0802: 08 0A 00 A5 AB 33 30 00 7D
080A: 14 08 14 00 BC 32 30 38 6E
0812: 33 00 1E 08 1E 00 BC 32 3C
081A: 30 38 30 00 00 00 4C 4D 3C
0822: 22 20 58 FC AD 61 C0 8D 28
082A: 59 25 A9 00 8D F2 03 A9 4D
0832: 09 8D F3 03 49 A5 8D F4 C9
083A: 03 A9 FD 85 39 85 37 A9 46
0842: 1B 85 38 A9 F0 85 36 A9 96
084A: 25 18 69 01 8D F0 24 1B C0
0852: 69 4F 85 6C A9 00 8D EF BA
085A: 24 8D F1 24 85 6B 8D 69 BE
0862: 22 85 FF 8D 58 25 A9 A5 0E
086A: 8D F2 24 A9 09 20 61 09 B1
0872: 20 D9 0A A9 23 A0 46 20 2D
087A: 3E 09 20 88 0D 20 25 09 B4
0882: 48 20 7C 09 68 AE AC 08 3E
088A: DD AC 08 F0 0A CA D0 F8 DA
0892: C9 20 90 E6 4C 37 0C CA 32
089A: 8A 0A AA A9 08 48 A9 7B 92
08A2: 48 BD D3 08 48 BD D2 08 28
08AA: 48 60 17 0E 00 17 06 07 2A
08B2: 10 03 13 0C 18 0A 0B 15 C2
08BA: 08 02 05 21 01 12 04 0D 67
08C2: 1B 23 0D 31 32 33 34 35 D9
08CA: 36 37 38 39 30 2B 2D 2E 15
08D2: C4 0A DB 11 13 10 A8 0C 8A
08DA: 4E 11 32 14 ED 15 A2 19 FF
08E2: 40 1A D5 1E DD 10 F6 10 63
08EA: 0D 11 37 11 94 1C DB 1C A6
08F2: 08 1C 03 1E A7 1C CC 1B B2
08FA: C8 15 08 09 ED 0C 20 58 7E
0902: FC 20 22 0B 4C 75 08 AD 85
090A: 58 25 49 FF 8D 58 25 60 33
0912: 2C 00 C0 10 0B AD 00 C0 23
091A: 8D 10 C0 29 7F C9 FF 60 25
0922: A9 00 60 A5 FF F0 07 48 89
092A: A9 00 85 FF 68 60 20 12 D8
0932: 09 F0 FB 60 20 F2 EB A5 D4
093A: A0 A4 A1 60 85 FC 84 FB 25
0942: 20 6F 09 20 80 FE A9 00 B6
094A: 85 28 85 24 85 25 A9 04 34
0952: 85 29 A0 00 B1 FB F0 06 EA
095A: 20 ED FD C8 D0 F6 60 A2 0A
0962: 32 9D F6 24 CA D0 FA A9 4F
096A: 28 8D 29 25 60 A0 00 A9 9A
0972: 20 99 00 04 C8 C0 28 D0 A5
097A: F6 60 AD 01 04 C9 10 D0 1E
0982: 0A AD 0A 04 C9 02 F0 03 C1
```

```
098A: 4C 94 09 A9 23 A0 3C 20 D7
0992: 3E 09 38 20 C7 1F 90 03 ED
099A: 4C 32 0F 4C 40 0F 09 80 D6
09A2: 8D 80 02 A9 3C 8D 81 02 93
09AA: A2 76 A9 A0 9D 81 02 CA AC
09B2: D0 F8 A0 01 D0 02 A0 00 5F
09BA: B9 80 02 8D 3C 25 A9 DF 8C
09C2: 99 80 02 20 AB 0A 20 12 DB
09CA: 09 D0 16 EE 3B 25 10 08 DD
09D2: A9 DF 99 80 02 4C C5 09 C2
09DA: AD 3C 25 99 80 02 4C C5 7B
09E2: 09 09 80 8D 3B 25 AD 3C AA
09EA: 25 99 80 02 AD 3B 25 AE 79
09F2: 95 0A DD 95 0A F0 2C CA 9E
09FA: D0 F8 C9 A0 90 BA 8C 3C BB
0A02: 25 CE 3C 25 A2 77 BD 80 25
0A0A: 02 C9 3C F0 AB CA BD 80 AC
0A12: 02 9D 81 02 CA EC 3C 25 86
0A1A: D0 F4 AD 3B 25 99 80 02 CF
0A22: C8 D0 95 CA 8A 0A AA BD BD
0A2A: 9E 0A 48 BD 9D 0A 48 60 FA
0A32: A0 00 B9 80 02 C9 3C F0 76
0A3A: 08 29 7F 99 00 03 C8 D0 94
0A42: F1 A9 00 99 00 03 8C 2C A4
0A4A: 25 60 AD 6A 22 F0 20 C0 3B
0A52: 00 F0 01 88 4C BA 09 AD 58
0A5A: 6A 22 F0 13 B9 80 02 C9 19
0A62: 3C F0 F1 C8 4C BA 09 AD A8
0A6A: 6A 22 F0 03 4C BA 09 AD 97
0A72: 3B 25 29 7F 85 FF 4C 32 81
0A7A: 0A C0 00 F0 D7 88 98 AA 8F
0A82: BD 81 02 9D 80 02 E8 C9 97
0A8A: 3C D0 F5 A9 A0 9D 80 02 C8
0A92: 4C BA 09 07 8D 9B 8A 8B 88
0A9A: 88 95 FF 31 0A 7A 0A 68 22
0AA2: 0A 68 0A 4B 0A 58 0A 7A 0C
0AAA: 0A A2 00 BD 80 02 9D 80 10
0AB2: 04 BD A8 02 9D 00 05 BD 22
0ABA: D0 02 9D 80 05 E8 E0 28 29
0AC2: D0 E9 60 A9 23 A0 60 20 DC
0ACA: 3E 09 20 25 09 C9 59 D0 89
0AD2: 03 20 D9 0A 4C 7C 09 20 D2
0ADA: FA 0A A9 09 20 61 09 20 6D
0AE2: 22 0B 20 88 0D A9 2C 8D 4C
0AEA: B4 22 A9 00 8D B3 22 A5 3C
0AF2: 6B 85 6F A5 6C 85 70 60 21
0AFA: AD EF 24 85 FB AD F0 24 5B
0B02: 85 FC A0 00 98 91 FB C8 F9
0B0A: D0 FB E6 FC A6 FC EC F2 2A
0B12: 24 D0 F2 A9 01 8D F4 24 B3
0B1A: 8D F5 24 85 1D 85 1E 60 EC
0B22: 20 28 0B 4C B0 0B A0 05 70
0B2A: 8C 3B 25 B9 83 22 85 28 6D
0B32: B9 6B 22 85 29 A0 00 AE 17
0B3A: F5 24 A9 00 8D 29 25 8D 72
0B42: 2A 25 F8 AD 29 25 18 69 28
0B4A: 01 8D 29 25 AD 2A 25 69 85
0B52: 00 8D 2A 25 CA D0 EC D8 AF
0B5A: A2 00 20 8D 0B F8 AD 29 5F
0B62: 25 18 69 01 8D 29 25 AD 57
0B6A: 2A 25 69 00 8D 2A 25 D8 44
0B72: EE 3B 25 AC 3B 25 B9 83 A3
0B7A: 22 85 28 B9 6B 22 85 29 BB
0B82: A0 00 E8 E0 12 D0 D3 20 AF
0B8A: 8D 0B 60 AD 2A 25 18 69 90
0B92: 30 91 28 C8 AD 29 25 29 3C
0B9A: F0 4A 4A 4A 4A 18 69 30 5F
0BA2: 91 28 C8 AD 29 25 29 0F BE
0BAA: 18 69 30 91 28 60 A0 04 4E
0BB2: B9 83 22 85 28 B9 6B 22 44
0BBA: 85 29 A0 00 A9 20 91 28 0B
0BC2: C8 91 28 C8 91 28 C8 AE A0
0BCA: F4 24 A9 00 8D F3 24 BD DB
0BD2: F6 24 8E 29 25 4A 69 00 F6
0BDA: AA CA A9 20 91 28 C8 CA B9
0BE2: D0 FA AD 29 25 0A AA BD CC
0BEA: B5 22 29 3F 91 28 C8 BD F9
0BF2: B6 22 29 3F 91 28 C8 AE 73
0BFA: 29 25 BD F6 24 4A AA CA 80
0C02: CA A9 20 91 28 C8 CA 10 11
0C0A: FA AE 29 25 BD F6 24 18 EC
0C12: 6D F3 24 8D F3 24 E8 BD FA
0C1A: F6 24 18 6D F3 24 C9 25 79
0C22: 90 AD CA 8E 32 25 A9 20 C9
0C2A: C0 28 D0 01 60 91 28 C8 39
0C32: C0 28 D0 F9 60 20 A0 09 3C
0C3A: AD 00 03 F0 3F C9 3D F0 25
0C42: 26 AE C4 08 DD C4 08 F0 35
0C4A: 07 CA D0 F8 A9 01 D0 19 4E
0C52: AD 2C 25 C9 25 B0 25 A0 64
0C5A: 00 A9 03 20 B1 0C 20 B7 73
0C62: 00 D0 E9 A9 00 F0 02 A9 F7
0C6A: 02 8D 2B 25 AD B4 22 8D B0
0C72: 2D 25 18 20 C7 1F 20 27 91
0C7A: 20 20 03 1C 4C 7C 08 85 B6
0C82: B9 84 B8 20 B7 00 4C 4A 52
0C8A: EC A2 32 A9 00 8D 38 25 6E
0C92: BD F6 24 18 6D 38 25 8D 71
0C9A: 38 25 C9 25 B0 03 CA D0 9B
0CA2: EF E8 E8 8E 3C 25 60 A9 D3
0CAA: 00 2C 59 25 30 03 AD 61 95
0CB2: C0 0D 58 25 8D 57 25 A0 80
0CBA: A5 A9 23 20 3E 09 20 25 F1
0CC2: 09 C9 4C F0 0F C9 43 F0 81
0CCA: 0F C9 52 F0 03 4C 85 0D 97
0CD2: A2 0C D0 06 A2 08 D0 02 92
0CDA: A2 04 AD B4 22 29 F0 8D 6B
0CE2: 3B 25 8A 0D 3B 25 8D 3B C8
0CEA: 25 4C 2F 0D A9 00 2C 59 5E
0CF2: 25 30 03 AD 61 C0 0D 58 65
0CFA: 25 8D 57 25 A0 CE A9 23 FC
0D02: 20 3E 09 20 76 10 F0 7B 30
0D0A: A0 00 A9 02 20 81 0C 20 09
0D12: 36 09 C9 00 D0 6D C0 10 90
0D1A: B0 69 AD B4 22 29 0C 8D 43
0D22: 3B 25 98 0A 0A 0A 0A 0D 70
0D2A: 3B 25 8D 3B 25 AD 57 25 44
0D32: 10 41 AD 3B 25 8D B4 22 F8
0D3A: AD EF 24 85 1B AD F0 24 99
0D42: 85 1C A0 01 B1 1B F0 11 37
0D4A: 85 1A 88 B1 1B 85 19 B1 AC
0D52: 19 29 03 0D B4 22 91 19 DE
0D5A: C8 A5 1B 18 69 02 85 1B A0
0D62: A5 1C 69 00 85 1C A5 1C 87
0D6A: C5 6C D0 D8 38 20 C7 1F 1B
0D72: 4C 85 0D 38 20 C7 1F 90 28
0D7A: 0A A0 00 AD 3B 25 0D 2B 50
0D82: 25 91 19 4C 7C 09 A5 1D EB
0D8A: 8D 30 25 A5 1E 8D 31 25 25
0D92: A9 03 8D F3 24 AE F4 24 1D
0D9A: 86 1D AC F5 24 84 1E 98 3C
0DA2: 18 69 13 8D 2E 25 BD F6 D6
0DAA: 24 8D 38 25 A9 FF EC 30 EA
0DB2: 25 D0 07 CC 31 25 D0 02 03
0DBA: A9 3F 8D 33 25 98 18 69 83
0DC2: 05 38 ED F5 24 A8 B9 6B 2D
0DCA: 22 85 29 B9 83 22 85 28 EF
0DD2: 38 20 C7 1F B0 05 A9 A0 89
0DDA: 4C 67 0E AD 2B 25 F0 70 D1
0DE2: C9 02 F0 6C AD 38 25 38 18
0DEA: ED 2C 25 AA E8 30 32 E8 AB
0DF2: AD 2D 25 29 0C C9 08 F0 EE
0DFA: 28 B0 05 8A 4A F0 22 AA A3
0E02: 8E 34 25 A9 A0 2D 33 25 F6
0E0A: AC F3 24 91 28 C8 CA D0 E1
0E12: FA 8C 35 25 AD 38 25 38 98
0E1A: ED 34 25 AA A0 02 4C 2E 5D
0E22: 0E AE 38 25 AD F3 24 8D 5D
0E2A: 35 25 A0 02 B1 19 8C 34 9D
0E32: 25 AC 35 25 09 80 2D 33 DC
0E3A: 25 91 28 AC 34 25 EE 35 66
0E42: 25 CA F0 09 C8 CC 2C 25 49
0E4A: D0 E2 20 A9 0E 4C 76 0E C2
0E52: 20 4E 0F AE 2C 25 CA CA 35
0E5A: CA EC 38 25 B0 03 4C E6 81
0E62: 0D A9 2A 09 80 2D 33 25 89
0E6A: AC F3 24 AE 38 25 91 28 EA
0E72: C8 CA D0 FA A4 1E A6 1D 77
0E7A: C8 CC 2E 25 F0 05 84 1E 09
0E82: 4C A8 0D AC F5 24 84 1E C2
0E8A: AD 38 25 18 6D F3 24 8D C2
0E92: F3 24 E8 86 1D E0 33 F0 FA
0E9A: 27 BD F6 24 18 6D F3 24 5D
0EA2: C9 28 B0 1C 4C A8 0D E0 85
0EAA: 00 F0 14 AD F3 24 18 6D 2E
0EB2: 38 25 A8 88 A9 A0 2D 33 2F
0EBA: 25 91 28 88 CA D0 FA 60 4B
0EC2: A9 28 38 ED F3 24 8D 38 27
0ECA: 25 A0 05 84 1E B9 6B 22 5B
0ED2: 85 29 B9 83 22 85 28 AC 8F
0EDA: F3 24 AE 38 25 A9 A0 91 F5
0EE2: 28 C8 CA D0 FA E6 1E A4 FF
0EEA: 1E C0 18 D0 E0 AD 30 25 99
0EF2: 85 1D AD 31 25 85 1E A0 FD
0EFA: 00 A9 A0 99 B0 02 C8 C0 8D
0F02: 78 D0 F8 38 20 C7 1F 90 22
0F0A: 35 A0 02 A2 00 AD 2B 25 87
0F12: C9 02 D0 09 AC 2C 25 B1 52
0F1A: 19 8D 2C 25 C8 B1 19 09 48
0F22: 80 9D 80 02 E8 C8 CC 2C 48
0F2A: 25 D0 F2 A9 3C 9D 80 02 63
0F32: AE 2B 25 BD B0 22 29 3F 92
0F3A: 8D 27 04 4C AB 0A A9 20 27
0F42: 8D 27 04 A9 3C 8D 80 02 27
0F4A: 20 AB 0A 60 A9 20 8D 00 93
0F52: 02 A0 02 B1 19 C9 2A F0 2A
0F5A: F2 AD 2D 25 4A 4A 4A 4A AF
0F62: 8D 36 25 A2 FF C9 0F F0 D9
0F6A: E2 B1 19 C9 2E D0 09 AE 9B
0F72: 36 25 F0 10 E8 8E 00 02 97
0F7A: 99 FF 01 C8 CC 2C 25 F0 64
0F82: 03 CA D0 E5 AD 36 25 F0 CE
0F8A: 1E E0 00 F0 1A AD 00 02 88
0F92: C9 20 D0 0A A9 2E 99 FF 91
0F9A: 01 C8 AE 36 25 E8 A9 30 F4
0FA2: 99 FF 01 C8 CA D0 F9 A9 71
0FAA: 20 8D 00 02 CC 2C 25 F0 AE
0FB2: 0C B0 3F B1 19 C9 2E F0 43
0FBA: 08 C9 35 B0 0C C8 4C F4 12
0FC2: 0F C8 B1 19 C9 35 90 2A D0
0FCA: 88 98 C8 AA CA CA BD 00 14
0FD2: 02 C9 2E F0 0B 90 0C C9 B5
0FDA: 39 D0 14 A9 30 9D 00 02 E0
0FE2: CA 10 EB CA 9D 00 02 58 6E
0FEA: A9 31 9D 00 02 D0 03 FE 36
0FF2: 00 02 88 8C 2C 25 AD 00 BC
0FFA: 02 C9 20 D0 09 A9 01 85 14
1002: 1A A9 FF 85 19 60 A9 01 90
100A: 85 1A A9 FE 85 19 EE 2C 33
1012: 25 60 A9 00 2C 59 25 30 53
101A: 03 AD 61 C0 0D 58 25 8D 01
1022: 57 25 A9 23 A0 79 20 3E 08
102A: 09 20 76 10 A0 00 A9 02 01
1032: 20 81 0C 20 36 09 C9 00 AF
103A: D0 33 C0 04 90 2F C0 25 CF
1042: B0 2B A5 1D 8D F4 24 AD 42
104A: 57 25 10 07 98 20 61 09 E2
1052: 4C 5B 10 98 A6 1D 9D F6 D6
105A: 24 20 8B 0C A5 1D CD 3C 40
1062: 25 90 07 AC 3C 25 88 8C F8
106A: F4 24 20 B0 0B 4C 7C 09 A8
1072: A9 01 D0 02 A9 00 8D 37 81
107A: 25 A0 00 A9 1F 20 ED FD 43
1082: A9 88 20 ED FD 20 25 09 40
108A: C9 0D F0 3F C9 08 F0 26 5B
1092: C9 7F F0 22 C9 20 90 ED 95
109A: AE 37 25 D0 08 C9 30 90 E9
10A2: E4 C9 3A B0 E0 A6 24 E0 C4
10AA: 26 F0 DA 99 00 02 09 80 A9
10B2: 20 ED FD C8 D0 C5 C0 00 C9
10BA: F0 CB A9 A0 20 ED FD A9 E3
10C2: 88 20 ED FD 20 ED FD 88 0A
10CA: 4C 7D 10 A9 A0 20 ED FD 6C
10D2: A9 00 99 00 02 8C 36 25 CE
10DA: AD 00 02 60 A5 1E C9 C8 1A
10E2: F0 12 E6 1E AD F5 24 18 64
10EA: 69 12 C5 1E B0 06 EE F5 50
10F2: 24 20 28 0B 60 A5 1E C9 82
10FA: 01 F0 10 C6 1E AC F5 24 F9
1102: 88 C4 1E 90 06 CE F5 24 E1
110A: 20 28 0B 60 A5 1D C9 32 15
1112: F0 23 E6 1D AC 32 25 C4 61
111A: 1D B0 1A EE F4 24 AE F4 B3
1122: 24 A9 00 18 7D F6 24 E8 3B
112A: C9 25 90 F7 CA CA E4 1D 74
1132: 90 E9 20 B0 0B 60 A5 1D 68
113A: C9 01 F0 10 C6 1D AC F4 99
1142: 24 88 C4 1D 90 06 CE F4 32
114A: 24 20 B0 0B 60 A9 23 A0 DD
1152: 80 20 3E 09 20 72 10 A9 A9
115A: 01 85 B9 A9 FF 85 B8 20 D7
1162: B1 00 90 4E 38 E9 41 30 70
116A: 49 F0 06 C9 02 B0 43 A9 CD
1172: 1A 8D 3B 25 20 B1 00 90 17
117A: 39 38 E9 40 30 34 F0 32 EE
1182: C9 1B B0 2E 18 6D 3B 25 5B
118A: C9 33 B0 26 8D 3B 25 20 9A
1192: B1 00 B0 1E 20 4A EC 20 A9
119A: 36 09 C9 00 D0 14 C0 00 AB
11A2: F0 10 C0 C9 B0 0C C0 B7 E4
11AA: 90 0B A9 B6 8D F5 24 4C 51
```

```
11B2: BA 11 4C 7C 09 8C F5 24 52
11BA: 84 1E 20 8B 0C AD 3B 25 16
11C2: CD 3C 25 90 0A AC 3C 25 29
11CA: 88 8C F4 24 4C D4 11 8D 9A
11D2: F4 24 85 1D 20 22 0B 4C E6
11DA: 7C 09 AD F4 24 C5 1D D0 C5
11E2: 17 AD F5 24 C5 1E D0 10 55
11EA: A9 01 8D F4 24 85 1D 8D 22
11F2: E5 24 85 1E 20 22 0B 60 AB
11FA: AD F4 24 85 1D AD F5 24 BD
1202: 85 1E 60 20 B1 00 8D 4F 76
120A: 25 20 B1 00 8D 50 25 20 17
1212: B1 00 8D 51 25 20 B1 00 E2
121A: C9 28 F0 03 4C 4D 22 AE 06
1222: 6A 12 AD 4F 25 DD 6A 12 32
122A: F0 06 CA D0 F5 4C 4D 22 4C
1232: AD 50 25 DD 76 12 F0 02 A3
123A: D0 F0 AD 51 25 DD 82 12 85
1242: D0 E8 8E 29 25 E0 0B B0 E0
124A: 0C 8A 48 A9 00 48 4C 22 96
1252: 21 68 8D 29 25 20 B1 00 72
125A: AE 29 25 CA 8A 0A AA BD 01
1262: 90 12 48 BD 8F 12 48 60 ED
126A: 0C 41 41 43 45 49 4C 53 7C
1272: 53 53 54 53 41 42 54 4F DF
127A: 58 4E 4F 47 49 51 41 55 24
1282: 56 53 4E 53 50 54 47 4E 56
128A: 4E 52 4E 4D 45 AE EB 9D 63
1292: F0 E9 EF 08 EF 22 EC 40 4A
129A: E9 8F EB F0 EF 8C EE 39 EC
12A2: F0 A9 13 11 14 20 64 13 1A
12AA: 8E 52 25 8C 54 25 20 B7 47
12B2: 00 C9 3A D0 3F 20 B1 00 7B
12BA: 20 64 13 8E 53 25 8C 55 F0
12C2: 25 20 B7 00 C9 29 D0 2C 39
12CA: 20 B1 00 AE 52 25 CA EC FF
12D2: 53 25 90 03 4C 4D 22 AC B4
12DA: 54 25 88 CC 55 25 90 03 B3
12E2: 4C 4D 22 E8 C8 A5 1D 8D F7
12EA: 39 25 A5 1E 8D 3A 25 86 B1
12F2: 1D 84 1E 60 4C 4D 22 18 84
12FA: 20 C7 1F 90 42 A0 00 B1 54
1302: 19 29 03 C9 01 F0 38 C8 01
130A: B1 19 8D 3C 25 A2 00 C8 41
1312: B1 19 9D 00 02 E8 C8 CC 1D
131A: 3C 25 D0 F4 A5 B8 48 A5 57
1322: B9 48 A9 00 9D 00 02 A9 07
132A: 02 A0 00 20 81 0C 68 85 0E
1332: B9 68 85 B8 A5 1D CD 53 1C
133A: 25 F0 15 E6 1D 18 60 AD F7
1342: 39 25 85 1D AD 3A 25 85 F6
134A: 1E 18 20 C7 1F 4C 4D 22 EC
1352: AD 52 25 85 1D A5 1E CD 6A
135A: 55 25 F0 04 E6 1E 18 60 13
1362: 38 60 A2 00 20 B7 00 C9 BA
136A: 41 F0 06 C9 42 D0 D0 A2 64
1372: 1A 8E 3B 25 20 B1 00 C9 94
137A: 41 90 C4 C9 5B B0 C0 38 F1
1382: E9 40 18 6D 3B 25 C9 33 BC
138A: B0 B5 8D 3B 25 20 B1 00 E8
1392: B0 AD 20 4A EC 20 36 09 82
139A: C9 00 D0 A3 C0 00 F0 9F B1
13A2: C0 C9 B0 9B AE 3B 25 60 78
13AA: A9 01 8D 29 25 A9 00 8D B7
13B2: 2A 25 20 A7 12 20 F9 12 CC
13BA: B0 47 20 72 EB A5 A2 48 B9
13C2: A5 A1 48 A5 A0 48 A5 9F 98
13CA: 48 A5 9E 48 A5 9D 48 EE F9
13D2: 29 25 D0 03 EE 2A 25 20 AB
13DA: F9 12 08 68 8D 3B 25 68 16
13E2: 85 A5 68 85 A6 68 85 A7 24
13EA: 68 85 A8 68 85 A9 68 85 6B
13F2: AA 45 A2 85 AB A5 9D 20 BB
13FA: C1 E7 AD 3B 25 48 28 90 90
1402: B9 AD 39 25 85 1D AD 3A 22
140A: 25 85 1E 18 20 C7 1F 60 2A
1412: 20 AA 13 A2 06 B5 9C 95 57
141A: A4 CA D0 F9 AD 2A 25 AC 0E
1422: 29 25 20 F2 E2 A5 AA 45 A3
142A: A2 85 AB A5 9D 20 F3 21 4B
1432: 60 20 58 FC A9 01 8D 56 30
143A: 25 A9 00 2C 59 25 30 03 E4
1442: AD 61 C0 0D 58 25 30 03 3D
144A: 4C A9 14 A9 24 A0 91 20 07
1452: 3E 09 20 25 09 C9 53 F0 39
145A: 0B C9 44 F0 0E C9 50 F0 3B
1462: 28 4C B2 15 A9 03 8D 56 24
146A: 25 D0 3C A9 00 8D 56 25 83
1472: A0 B5 A9 24 20 3E 09 20 FB
147A: 72 10 A9 00 AA 20 0A 1B 1A
1482: F0 25 C9 06 F0 21 4C A5 50
148A: 15 A9 24 A0 8A 20 3E 09 90
1492: 20 25 09 38 E9 30 C9 00 5C
149A: B0 03 4C B2 15 C9 08 90 01
14A2: 03 4C B2 15 8D 56 25 A9 C0
14AA: 24 A0 7E 20 3E 09 20 84 B9
14B2: FE AD 56 25 F0 14 C9 03 51
14BA: D0 0D AD 05 C3 18 6D 07 F4
14C2: C3 C9 50 D0 05 A9 03 20 4B
14CA: 95 FE A5 1D 8D 53 25 8D 95
14D2: 30 25 A5 1E 8D 55 25 8D 8C
14DA: 31 25 A9 01 85 1D 85 1E F3
14E2: A9 8D 20 BB 15 A6 1D BD 3E
14EA: F6 24 8D 38 25 AA A9 00 F3
14F2: 9D 00 03 CA A9 20 9D 00 FF
14FA: 03 CA 10 FA 38 20 C7 1F F9
1502: 90 58 AD 2B 25 C9 01 D0 16
150A: 23 AD 38 25 38 ED 2C 25 81
1512: AA E8 30 14 E8 AD 2D 25 90
151A: 29 0C C9 08 F0 0A B0 27 CD
1522: 8A 4A F0 04 AA 4C 49 15 B0
152A: A2 00 F0 1B 20 4E 0F AE 7C
1532: 2C 25 CA CA CA EC 38 25 61
153A: 90 CF AE 38 25 A9 2A 9D BB
1542: FF 02 CA D0 FA F0 13 A0 B5
154A: 02 B1 19 9D 00 03 E8 C8 85
1552: EC 38 25 F0 05 CC 2C 25 8D
155A: D0 EF A2 00 BD 00 03 F0 22
1562: 08 09 B0 20 BB 15 E8 D0 B9
156A: F3 A5 1D CD 53 25 F0 05 8E
1572: E6 1D 4C E7 14 A5 1E CD A0
157A: 55 25 F0 0E E6 1E A9 01 9B
1582: 85 1D A9 8D 20 BB 15 4C 2B
158A: E7 14 A9 8D 20 BB 15 AD 83
1592: 56 25 C9 03 D0 03 20 25 92
159A: 09 A9 00 20 95 FE AD 56 10
15A2: 25 D0 03 20 52 1B AD 30 80
15AA: 25 85 1D AD 31 25 85 1E 8E
15B2: 20 58 FC 20 22 0B 4C 7C F6
15BA: 09 48 AD 56 25 F0 04 68 F3
15C2: 4C ED FD 68 4C 95 1B A9 6D
15CA: 00 2C 59 25 30 03 AD 61 C7
15D2: C0 0D 58 25 8D 3F 25 A9 5B
15DA: 00 8D 40 25 A5 1D 8D 41 C0
15E2: 25 A5 1E 8D 42 25 4C 0D F1
15EA: 16 4C 7C 09 A9 00 2C 59 52
15F2: 25 30 03 AD 61 C0 0D 58 77
15FA: 25 8D 3F 25 A9 01 8D 40 02
1602: 25 A5 1D 8D 41 25 A5 1E AE
160A: 8D 42 25 20 48 16 AD 30 5A
1612: 25 8D 45 25 AD 31 25 8D 39
161A: 46 25 20 52 16 AE 41 25 EE
1622: CA EC 45 25 B0 13 AE 42 5B
162A: 25 CA EC 46 25 B0 0A A9 47
1632: 23 A0 FA 20 3E 09 20 44 14
163A: 18 AD 43 25 85 1D AD 44 D8
1642: 25 85 1E 4C 7C 09 A9 24 6A
164A: A0 59 20 3E 09 4C 59 16 47
1652: A9 24 A0 31 20 3E 09 20 AF
165A: 88 0D 20 25 09 AE 8D 16 98
1662: DD 8D 16 F0 06 CA D0 F8 A8
166A: 4C 59 16 CA 8A 0A AA A9 FD
1672: 16 48 A9 58 48 BD 95 16 F0
167A: 48 BD 94 16 48 60 68 68 2B
1682: A5 1D 8D 43 25 A5 1E 8D 38
168A: 44 25 60 06 00 0B 0A 08 D6
1692: 15 0D DB 11 F6 10 DD 10 DC
169A: 37 11 0D 11 7F 16 AD 49 52
16A2: 25 C9 33 B0 5B AD 4A 25 90
16AA: C9 C9 B0 54 AD 47 25 85 E3
16B2: 1D AD 48 25 85 1E 38 20 69
16BA: C7 1F 90 45 A0 02 AD 2B 8C
16C2: 25 C9 02 D0 09 AC 2C 25 B9
16CA: B1 19 8D 2C 25 C8 A2 00 1C
16D2: B1 19 9D 00 03 E8 C8 CC EB
16DA: 2C 25 D0 F4 A9 00 9D 00 58
16E2: 03 8E 2C 25 20 25 17 AD 7D
16EA: 40 25 D0 03 20 13 17 AD F3
16F2: 49 25 85 1D AD 4A 25 85 F5
16FA: 1E 18 20 C7 1F 20 27 20 A4
1702: 60 AD 49 25 85 1D AD 4A 8D
170A: 25 85 1E 18 20 C7 1F 90 60
1712: EF 20 33 1E 18 20 C7 1F 78
171A: A9 00 A8 91 1B C8 91 1B 85
1722: 4C 02 17 AD 3F 25 30 01 A4
172A: 60 AD 2B 25 C9 02 F0 01 E4
1732: 60 AD 49 25 38 ED 47 25 A4
173A: 8D 4D 25 AD 4A 25 38 ED 47
1742: 48 25 8D 4E 25 A2 00 8E B6
174A: 2A 25 BD 00 03 9D 80 02 20
1752: E8 EC 2C 25 D0 F4 A9 00 B5
175A: 9D 80 02 A9 80 85 B8 A9 B7
1762: 02 85 B9 A9 00 85 FB A9 7C
176A: 03 85 FC 20 B7 00 20 37 52
1772: 18 20 B1 00 C9 00 D0 03 DD
177A: 4C 2C 18 C9 40 D0 03 4C 11
1782: 17 18 90 EA C9 43 B0 E6 A6
178A: A2 00 C9 42 D0 02 A2 1A 55
1792: 8E 29 25 20 B1 00 C9 41 5B
179A: 90 66 C9 5B B0 62 38 E9 03
17A2: 40 18 6D 29 25 C9 33 B0 9E
17AA: 57 18 6D 4D 25 A2 41 C9 0D
17B2: 1B 90 05 A2 42 38 E9 1A 3E
17BA: 18 69 40 8D 29 25 8A 20 43
17C2: 37 18 AD 29 25 20 37 18 0B
17CA: 20 B1 00 B0 33 20 4A EC 1C
17D2: 20 36 09 C9 00 D0 29 C0 B2
17DA: 00 F0 25 C0 C9 B0 21 98 E1
17E2: 18 6D 4E 25 A8 A9 00 20 A0
17EA: F2 E2 20 34 ED A2 00 BD 4A
17F2: 00 01 F0 06 20 37 18 E8 D6
17FA: D0 F5 20 B7 00 4C 76 17 C3
1802: A2 00 BD 80 02 F0 06 9D C0
180A: 00 03 E8 D0 F5 A9 00 9D 19
1812: 00 03 4C 36 18 20 37 18 B7
181A: 20 B1 00 20 37 18 20 B1 D4
1822: 00 20 37 18 20 B1 00 4C D6
182A: 70 17 AC 2A 25 8C 2C 25 69
1832: A9 00 91 FB 60 AC 2A 25 58
183A: C0 78 F0 05 91 FB EE 2A DB
1842: 25 60 AD 45 25 38 ED 41 4E
184A: 25 18 6D 30 25 8D 4B 25 DE
1852: AD 46 25 38 ED 42 25 18 ED
185A: 6D 31 25 8D 4C 25 AD 42 9F
1862: 25 CD 31 25 B0 03 4C 07 42
186A: 19 AD 41 25 CD 30 25 90 17
1872: 4A AD 41 25 8D 47 25 AD 2F
187A: 42 25 8D 48 25 AD 30 25 B0
1882: 8D 49 25 AD 31 25 8D 4A CE
188A: 25 20 A0 16 AD 47 25 CD 6D
1892: 45 25 F0 08 EE 47 25 EE 1B
189A: 49 25 D0 ED AD 48 25 CD 58
18A2: 46 25 F0 14 EE 48 25 EE 70
18AA: 4A 25 AD 41 25 8D 47 25 26
18B2: AD 30 25 8D 49 25 D0 D1 95
18BA: 4C A0 19 AD 45 25 8D 47 58
18C2: 25 AD 4B 25 8D 49 25 AD 36
18CA: 42 25 8D 48 25 AD 31 25 03
18D2: 8D 4A 25 20 A0 16 AD 47 03
18DA: 25 CD 41 25 F0 08 CE 47 18
18E2: 25 CE 49 25 D0 ED AD 48 B6
18EA: 25 CD 46 25 F0 CA EE 48 15
18F2: 25 EE 4A 25 AD 45 25 8D 67
18FA: 47 25 AD 4B 25 8D 49 25 99
1902: D0 D1 4C A0 19 AD 41 25 CB
190A: CD 30 25 90 4A AD 41 25 8D
1912: 8D 47 25 AD 46 25 8D 48 86
191A: 25 AD 30 25 8D 49 25 AD 2C
1922: 4C 25 8D 4A 25 20 A0 16 1B
192A: AD 47 25 CD 45 25 F0 08 2F
1932: EE 47 25 EE 49 25 D0 ED AF
193A: AD 48 25 CD 42 25 F0 14 73
1942: CE 48 25 CE 4A 25 AD 41 03
194A: 25 8D 47 25 AD 30 25 8D B3
1952: 49 25 D0 D1 4C A0 19 AD 6E
195A: 45 25 8D 47 25 AD 4B 25 3A
1962: 8D 49 25 AD 46 25 8D 48 57
196A: 25 AD 4C 25 8D 4A 25 20 76
1972: A0 16 AD 47 25 CD 41 25 AC
197A: F0 08 CE 47 25 CE 49 25 91
1982: D0 ED AD 48 25 CD 42 25 DC
198A: F0 14 CE 48 25 CE 4A 25 B6
1992: AD 45 25 8D 47 25 AD 4B DF
199A: 25 8D 49 25 D0 D1 4C 09 AD
19A2: 1C A9 23 A0 99 20 3E 09 8E
19AA: 20 72 10 D0 03 4C 7C 09 E3
19B2: A2 00 A9 08 20 0A 1B F0 3C
19BA: 07 C9 06 F0 03 4C B7 1B 86
19C2: A9 FF 20 95 1B A9 FF 20 C6
19CA: 95 1B A5 6F 20 95 1B A5 6D
19D2: 70 20 95 1B A0 32 B9 F6 E1
19DA: 24 20 95 1B 88 D0 F7 AD B0
```

```
19E2: EF 24 85 1B AD F0 24 85 77
19EA: 1C A0 01 B1 1B F0 16 A5 FC
19F2: 1B 20 95 1B A5 1C 20 95 92
19FA: 1B 88 B1 1B 20 95 1B C8 1B
1A02: B1 1B 20 95 1B A5 1B 18 F0
1A0A: 69 02 85 1B A5 1C 69 00 46
1A12: 85 1C A5 1C C5 6C D0 D1 D9
1A1A: A9 FF 20 95 1B A5 6B 85 4C
1A22: 1B A5 6C 85 1C A0 00 B1 48
1A2A: 1B 20 95 1B C8 D0 F8 E6 BA
1A32: 1C A5 1C C5 70 90 F0 F0 56
1A3A: EE 20 52 1B 4C CD 1A A9 61
1A42: 23 A0 9F 20 3E 09 20 72 EE
1A4A: 10 D0 03 4C 7C 09 A2 01 2E
1A52: A9 08 20 0A 1B F0 03 4C F0
1A5A: B7 1B 20 79 1B C9 FF D0 9D
1A62: 60 20 79 1B C9 FF D0 59 F8
1A6A: 20 FA 0A 20 79 1B 85 6F 63
1A72: 20 79 1B 85 70 A0 32 20 5B
1A7A: 79 1B 99 F6 24 88 D0 F7 B1
1A82: 20 79 1B C9 FF F0 18 85 9E
1A8A: 1B 20 79 1B 85 1C 20 79 8B
1A92: 1B A0 00 91 1B 20 79 1B FC
1A9A: A0 01 91 1B 4C 82 1A A5 89
1AA2: 6B 85 1B A5 6C 85 1C A0 FD
1AAA: 00 20 79 1B 91 1B C8 D0 23
1AB2: F8 E6 1C A5 1C C5 70 90 64
1ABA: F0 F0 EE 20 52 1B 4C CD E8
1AC2: 1A 20 52 1B A9 24 A0 DA 02
1ACA: 4C 3E 09 AD C5 B5 D0 08 5F
1AD2: A9 24 A0 BF 20 3E 09 60 61
1ADA: 20 6F 09 A9 00 85 24 85 9A
1AE2: 28 A9 04 85 29 20 80 FE 38
1AEA: AE C5 B5 BD 3F AA AA 8E 03
1AF2: 29 25 BD 71 A9 48 09 80 D4
1AFA: 20 ED FD AE 29 25 E8 68 7D
1B02: 10 ED A9 87 20 F0 FD 60 8A
1B0A: 8D C2 B5 A9 01 8D BB B5 74
1B12: 8D C0 B5 A9 00 8D BD B5 F7
1B1A: 8D BE B5 8D BF B5 A9 06 84
1B22: 8D C1 B5 A0 3C A9 A0 99 B3
1B2A: 74 AA 88 D0 FA A0 00 B9 77
1B32: 00 02 F0 08 09 80 99 75 7A
1B3A: AA C8 D0 F3 A9 AA 8D C4 29
1B42: B5 A9 75 8D C3 B5 20 60 DA
1B4A: 1B 20 D6 03 AD C5 B5 60 71
1B52: A9 02 8D BB B5 20 60 1B 55
1B5A: A2 01 20 D6 03 60 A9 00 80
1B62: 8D C7 B5 8D C9 B5 8D CB EC
1B6A: B5 A0 A6 8C CC B5 C8 8C 9C
1B72: CA B5 C8 8C C8 B5 60 8D C8
1B7A: C3 B5 98 48 8A 48 A9 03 63
1B82: 8D BB B5 A9 01 8D BC B5 2D
1B8A: 20 60 1B A2 01 20 D6 03 AF
1B92: 4C AE 1B 8D C3 B5 98 48 45
1B9A: 8A 48 A9 04 8D BB B5 A9 0E
1BA2: 01 8D BC B5 20 60 1B A2 0B
1BAA: 01 20 D6 03 AD C5 B5 F0 55
1BB2: 12 68 68 68 68 AD C5 B5 DA
1BBA: 48 20 52 1B 68 8D C5 B5 D3
1BC2: 4C CD 1A 68 AA 68 A8 AD 52
1BCA: C3 B5 60 20 58 FC 20 84 D9
1BD2: FE A9 06 8D BB B5 8D C1 1E
1BDA: B5 A9 01 8D C0 B5 20 60 CC
1BE2: 1B A2 01 20 D6 03 A9 23 AA
1BEA: 85 FC A9 EC 85 FB 20 54 D7
1BF2: 09 20 12 09 C9 0D D0 F9 A6
1BFA: 20 58 FC 20 22 0B 4C 7C 4B
1C02: 09 AD B3 22 D0 01 60 A9 B7
1C0A: 24 A0 C9 20 3E 09 A5 1D 36
1C12: 8D 30 25 A5 1E 8D 31 25 CA
1C1A: A9 01 85 1D 85 1E AD EF D9
1C22: 24 85 1B AD F0 24 85 1C 4B
1C2A: A0 01 B1 1B F0 35 85 1A 5C
1C32: 88 B1 1B 85 19 B1 19 29 C1
1C3A: 03 C9 02 D0 26 38 20 C7 CD
1C42: 1F A2 00 AC 2C 25 B1 19 EF
1C4A: 8D 2C 25 C8 B1 19 9D 00 B2
1C52: 03 E8 C8 CC 2C 25 D0 F4 B8
1C5A: A9 00 9D 00 03 8E 2C 25 EA
1C62: 20 27 20 A5 1B 18 69 02 E0
1C6A: 85 1B 90 02 E6 1C E6 1E F1
1C72: A5 1E C9 C9 D0 B2 A9 01 80
1C7A: 85 1E E6 1D A5 1D C9 33 14
1C82: D0 A6 AD 30 25 85 1D AD AC
1C8A: 31 25 85 1E 38 20 C7 1F 28
1C92: 4C 7C 09 20 33 1E 18 20 95
1C9A: C7 1F A9 00 A8 91 1B C8 3E
1CA2: 91 1B 20 03 1C 60 A9 23 77
1CAA: A0 86 20 3E 09 A9 00 2C D7
1CB2: 59 25 30 03 AD 61 C0 0D 98
1CBA: 58 25 30 08 AD B3 22 49 B8
1CC2: FF 8D B3 22 AD B3 22 C9 41
1CCA: 00 F0 06 A9 CE 20 ED FD 6B
1CD2: 60 A9 C6 20 ED FD 20 ED 16
1CDA: FD 60 EE 6A 22 20 B8 09 BA
1CE2: CE 6A 22 AD 00 03 F0 4E 78
1CEA: C9 3D F0 27 AE C4 08 DD 5E
1CF2: C4 08 F0 08 CA D0 F8 A9 63
1CFA: 01 4C 17 1D AD 2C 25 C9 AD
1D02: 25 B0 33 A0 00 A9 03 20 38
1D0A: 81 0C 20 B7 00 D0 E8 A9 46
1D12: 00 F0 02 A9 02 8D 2B 25 25
1D1A: 18 20 C7 1F B0 09 AD B4 0D
1D22: 22 8D 2D 25 4C 32 1D A0 CE
1D2A: 00 B1 19 29 FC 8D 2D 25 24
1D32: 20 27 20 20 03 1C 60 AE 44
1D3A: 36 25 CA CA CA CA BD 00 DB
1D42: 02 C9 45 D0 78 E8 BD 00 88
1D4A: 02 8D 3B 25 E8 BD 00 02 E2
1D52: 38 E9 30 8D 2A 25 E8 BD 77
1D5A: 00 02 38 E9 30 AE 2A 25 70
1D62: F0 06 18 69 0A CA D0 FA 48
1D6A: 8D 29 25 AD 3B 25 C9 2D 64
1D72: F0 4C A2 00 A0 00 BD 00 0D
1D7A: 02 C9 45 F0 08 E8 C9 2E 85
1D82: F0 F4 C8 D0 F1 88 8C 3B 9E
1D8A: 25 AD 29 25 38 ED 3B 25 4F
1D92: 8D 29 25 A2 01 A0 01 BD F6
1D9A: 00 02 E8 C9 2E F0 F8 C9 FF
1DA2: 45 F0 06 99 00 02 C8 D0 80
1DAA: EE A9 30 AE 29 25 99 00 C8
1DB2: 02 C8 CA D0 F9 A9 00 99 96
1DBA: 00 02 8C 36 25 60 CE 29 DB
1DC2: 25 A2 00 A0 00 BD 00 02 3B
1DCA: E8 C9 2E F0 F8 C9 45 F0 2B
1DD2: 06 99 80 02 C8 D0 EE A9 B7
1DDA: 00 99 80 02 A9 2E 8D 00 CC
1DE2: 02 AE 29 25 A9 30 9D 00 8A
1DEA: 02 CA D0 FA A2 00 AC 29 3A
1DF2: 25 C8 BD 80 02 99 00 02 2A
1DFA: F0 04 E8 C8 D0 F4 8C 36 02
1E02: 25 60 20 6F 09 A9 04 85 60
1E0A: 29 A9 00 85 28 85 24 AD EA
1E12: F1 24 38 E5 6F A8 AD F2 22
1E1A: 24 E5 70 20 21 1E 60 20 54
1E22: F2 E2 20 34 ED A9 01 85 75
1E2A: FC A9 00 85 FB 20 54 09 B9
1E32: 60 A0 01 B1 1B F0 E7 A9 18
1E3A: 00 91 1B 88 91 1B B1 19 3C
1E42: 29 03 C9 02 D0 09 C8 B1 1B
1E4A: 19 A8 B1 19 4C 54 1E C8 BD
1E52: B1 19 85 FB 18 65 19 8D 34
1E5A: 76 1E A5 19 8D 79 1E A5 D3
1E62: 1A 8D 7A 1E 69 00 8D 77 1E
1E6A: 1E A5 70 38 ED 77 1E AA E4
1E72: E8 A0 00 B9 FF FF 99 FF 1A
1E7A: FF C8 D0 F7 EE 77 1E EE 03
1E82: 7A 1E CA D0 EE A5 6F 38 0F
1E8A: E5 FB 85 6F A5 70 E9 00 23
1E92: 85 70 AD EF 24 85 FD AD 43
1E9A: F0 24 85 FE A0 01 B1 FD 63
1EA2: F0 22 38 88 B1 FD E5 19 D9
1EAA: 8D 29 25 C8 B1 FD E5 1A 94
1EB2: 0D 29 25 90 0F 88 B1 FD 69
1EBA: 38 E5 FB 91 FD C8 B1 FD 99
1EC2: E9 00 91 FD C8 F0 03 C8 DE
1ECA: D0 D4 E6 FE C8 A5 FE C5 12
1ED2: 6C D0 CB 60 A9 23 A0 22 36
1EDA: 20 3E 09 20 25 09 C9 59 14
1EE2: D0 03 4C 00 C6 4C 7C 09 3B
1EEA: AD 39 25 85 1D AD 3A 25 82
1EF2: 85 1E 18 20 C7 1F AD 3B CF
1EFA: 25 8D 2B 25 AD 3D 25 8D 1F
1F02: 2D 25 AD 3C 25 8D 2C 25 76
1F0A: 4C 4D 22 48 A5 1D 8D 39 80
1F12: 25 A5 1E 8D 3A 25 AD 2B D5
1F1A: 25 8D 3B 25 AD 2D 25 8D 02
1F22: 3D 25 AD 2C 25 8D 3C 25 BD
1F2A: 68 E9 41 30 BB F0 06 C9 B9
1F32: 02 B0 B5 A9 1A 85 1D 20 30
1F3A: B1 00 E9 40 30 AA F0 A8 49
1F42: C9 1B B0 A4 18 65 1D C9 E6
1F4A: 33 B0 9D 85 1D 20 B1 00 27
1F52: B0 96 20 4A EC 20 36 09 94
1F5A: C9 00 D0 8C C0 00 F0 88 D0
1F62: C0 C9 B0 84 84 1E 38 20 FE
1F6A: C7 1F 90 07 AD 2B 25 C9 05
1F72: 01 D0 03 4C EA 1E A0 02 9D
1F7A: A2 00 B1 19 C9 2A F0 F3 9E
1F82: B1 19 9D 00 02 C8 E8 CC 65
1F8A: 2C 25 D0 F4 A9 00 9D 00 1A
1F92: 02 A5 B8 48 A5 B9 48 A0 1C
1F9A: 00 A9 02 20 B1 0C 68 85 18
1FA2: B9 68 85 B8 AD 39 25 85 36
1FAA: 1D AD 3A 25 85 1E 18 20 71
1FB2: C7 1F AD 3B 25 8D 2B 25 E0
1FBA: AD 3D 25 8D 2D 25 AD 3C 32
1FC2: 25 8D 2C 25 60 08 A6 1D 5C
1FCA: CA 86 1B A9 C8 85 1C 18 BA
1FD2: A9 00 A2 08 6A 66 1B 90 6E
1FDA: 03 18 65 1C CA 10 F5 85 17
1FE2: 1C A6 1E CA 8A 18 65 1B E3
1FEA: 85 1B A5 1C 69 00 85 1C 9B
1FF2: 06 1B 26 1C A5 1C 6D F0 EA
1FFA: 24 85 1C A0 01 B1 1B D0 10
2002: 03 28 18 60 AA 88 B1 1B CC
200A: 85 19 86 1A 28 90 14 B1 23
2012: 19 29 03 8D 2B 25 B1 19 CC
201A: 29 FC 8D 2D 25 C8 B1 19 7B
2022: 8D 2C 25 38 60 20 33 1E 64
202A: AD 2B 25 C9 02 F0 32 EE 74
2032: 2C 25 EE 2C 25 A0 00 A5 C3
203A: 6F 91 1B C8 A5 70 91 1B B3
2042: 88 AD 2B 25 0D 2D 25 91 E2
204A: 6F C8 AD 2C 25 91 6F C8 04
2052: A2 00 BD 00 03 91 6F C8 A1
205A: E8 CC 2C 25 D0 F4 4C B6 C3
2062: 20 20 F2 20 EE 36 25 EE A4
206A: 36 25 38 AD 36 25 6D 2C 3E
2072: 25 8D 2C 25 AC 36 25 AD B6
207A: 2C 25 91 6F A2 00 C8 BD A7
2082: 00 03 91 6F C8 E8 CC 2C 5C
208A: 25 D0 F4 A0 00 A5 6F 91 41
2092: 1B C8 A5 70 91 1B 88 AD 06
209A: 2B 25 0D 2D 25 91 6F C8 45
20A2: AD 36 25 91 6F C8 A2 02 EA
20AA: BD FE 01 91 6F C8 E8 EC 20
20B2: 36 25 D0 F4 A5 6F 18 6D 49
20BA: 2C 25 90 06 A5 70 C9 A4 F3
20C2: F0 0F A5 6F 18 6D 2C 25 DE
20CA: 85 6F A5 70 69 00 85 70 2C
20D2: 60 A9 00 A8 91 1B C8 91 54
20DA: 1B A9 24 A0 13 20 3E 09 40
20E2: A5 1D 8D F4 24 A5 1E 8D BF
20EA: F5 24 A2 FD 9A 4C 7C 08 6A
20F2: BA 8E 3E 25 A2 00 A0 00 A4
20FA: BD 00 03 C9 28 D0 01 C8 66
2102: C9 29 D0 01 88 9D 00 03 5B
210A: E8 EC 2C 25 D0 EA C0 00 B7
2112: F0 03 4C 4D 22 A9 00 48 EB
211A: A9 00 85 B8 A9 03 85 B9 8B
2122: 20 B1 00 90 51 C9 2D F0 E6
212A: 4D C9 2B F0 49 C9 2E F0 B8
2132: 45 C9 50 F0 25 C9 28 F0 34
213A: 15 C9 41 F0 0B C9 42 F0 A5
2142: 07 C9 40 F0 0F 4C 4D 22 F7
214A: 20 0D 1F 4C 7B 21 A9 01 3D
2152: 48 4C 22 21 20 05 12 4C A7
215A: 7B 21 20 B1 00 C9 49 F0 6C
2162: 03 4C 4D 22 A9 73 A0 21 82
216A: 20 F9 EA 20 B1 00 4C 7B 3C
2172: 21 82 49 0F DA A1 20 4A E7
217A: EC 20 B7 00 F0 78 A2 02 E2
2182: C9 2B F0 35 E8 C9 2D F0 9F
218A: 30 E8 C9 2A F0 2B E8 C9 CA
2192: 2F F0 26 E8 C9 5E F0 21 C6
219A: C9 29 F0 03 4C 4D 22 68 9E
21A2: F0 14 C9 01 F0 07 48 20 FF
21AA: 19 22 4C A1 21 E6 B8 D0 8C
21B2: 02 E6 B9 4C 7B 21 4C 53 F7
21BA: 12 86 06 68 48 A8 B9 9B E2
21C2: 22 DD 9B 22 90 10 20 19 41
21CA: 22 A6 06 68 48 A8 B9 9B 03
21D2: 22 DD 9B 22 B0 F0 20 72 2F
21DA: EB A5 A2 48 A5 A1 48 A5 3F
21E2: A0 48 A5 9F 48 A5 9E 48 94
21EA: A5 9D 48 A5 06 48 4C 22 D6
21F2: 21 F0 58 4C 69 EA 68 48 E1
21FA: F0 06 20 19 22 4C F8 21 22
2202: 68 20 34 ED A0 00 B9 00 60
```